

Palestine Polytechnic University

College of Administrative sciences and informatics

Department of Information Technology

Polytechnic Academic Portal

PAP

Prepared By

**Ghazy T. Khamayseh.
Majid I. Khamayseh.
Muhammad J. Zumoot**

Supervised By

Dr. Muhammad Dasht.

**This project was prepared to complete the graduation requirements in Information
Technology major.**

June-2005

Dedication

To our parents for their ongoing support and encouragement all the time.

To our Supervisor Dr. Mohammad Dasht,

To all martyrs who scarify themselves struggling toward freedom,

and to all our teachers, lecturers, and friends.

Project teamwork:

Ghazy T. khamayseh.

Majid I. khamayseh.

Mohammad J. Zomot

Acknowledgment

Many thanks for our supervisor Dr. Mohammad Dasht and to every one who helped us in our project.

Thanks for the Head of the Information Technology Department in our university Mr. Ismail M. Romi.

Great thanks to the registration staff in our university for their help and patience.

Thanks to all who help to reveal this project to light.

Abstract

The ultimate aim of our project is to create a virtual community of PPU students and switch traditional student academic transactions into a computerized system by investing the great advances in web based systems and modulate them into more efficient and effective academic life for PPU students. By cutting paperwork and convert it into web based work to save time and effort of student and make their academic life more easy.

Our system tries to meet the majority of student requirements from academic and financial aspects to make student information accessible from anywhere and provide requirements to improve the student role in his university environment and keep him in touch with all events and announcements that occur in the university such as voting to some questionnaire, and provide other requirements to support student social life by answering frequently asked questions by students.

In addition to meeting student transactions; our portal support administrator and instructor functions to make all processes consistent and integrated. administrator can provide students with their information such as financial balances, boards, marks...etc and instructor can benefit from Portal functions too.

We conclude that the Portal should focus on design a web site to facilitate and enable students to do their university related basic tasks electronically by providing suitable, interesting and strong interface to attract students and minimize their potential errors.

Project problem

The project problem can be summarized in the following points:

❖ From student perspective:

1. the students suffer from many difficulties in obtaining their university related information, such as marks and financial balance whenever he want.
2. the weakness of communication between university locations and students, this problem due to the distant locations of university buildings.
3. political circumstances and occupation obstacles that prevent the student from arriving university and keep touch with their instructors and university.
4. high pressure on registration department caused by increased number of students seek their information.

❖ From instructor perspective :

1. the instructor suffer from difficulty to keep in touch with his student during semester, such as issue specific boards to his students and courses.
2. there is no mechanism that enable instructor to post his students marks to the registration department directly from anywhere .
3. there is no mechanism that enable the instructor to keep in touch with university issues and events.

- So we suggest to build web based system to overcome these problems .

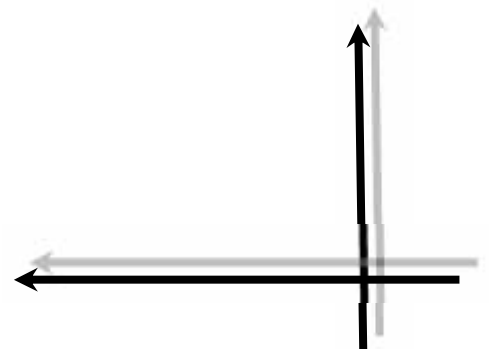
Chapter One

1

System Specification

Chapter One

- ▶ *Introduction*
- ▶ *System specifications*
- ▶ *System requirements*
- ▶ *System analysis*





1.1 Introduction:

This chapter explores and explains the important specifications provided by the system which includes system objectives, functional and non-functional description, project constraints, feasibility study, alternatives, economic feasibility, cost benefit analysis, risk evaluation and schedule for all tasks.

1.2 System Objectives:

In our system we work toward creating an integrated student web portal that should help the university to meet a variety of goals shared by students, faculty, staff and administrators, so objectives will include:

1.2.1 University Objectives:

1. Building virtual community of students by creating relationships by students and their university and information.
2. Improving the quality of the students life, thereby, they will practicing how to cope with the new technology rather than using traditional approach in serving students.
3. Applying technological advances in serving students.
4. Improving effectiveness in the university academic operations.
5. Help Palestine polytechnic university to cope with technological improvements in other universities.
6. The university will satisfy their student's needs so they will feel a kind of interest toward them.



7. Encourage new students to register in the university.
8. Integrated calendar system that would give staff and students an easy and consistent way to communicate and exchange information about deadlines and events

1.2.2 Student Objectives:

1. Provide easy access to on-line system and information sources via the portal.
2. Provide one central source of student's information.
3. Provide student personality by allowing the student to log in using his own username and password.
4. Save time and efforts because students can browse information from anywhere.
5. Students will be in touch with new events, announcements and information.

1.2.3 Administrator Objectives:

1. Build a database using DB server that containing the data and their relationships.
2. supporting security on the database by authorizing only the administrator to manipulate students data by add, delete and update data on a secure web site.



3. Enable the administrator to make some changes on the system if required.

1.2.4 Interactivity Objectives:

1. Portability gives the students to access their information and boards from anywhere they can access the World Wide Web.
2. Gives students more flexibility in when, where and how to show the portal.
3. A consistent look and feel toward portal can enhance the university image in the society.
4. Enhances effectiveness in student's services by enabling them to get their everyday information.
5. Easy for students to use by providing usable and customizable interface.

1.2.5 Development Objectives:

1. Using Visual Studio.NET as an evolutionary, more successor to active server Pages.
2. Using ADO.NET technology that provides a set of classes for working with data and it is evolutionary and more flexible version of active data objects and it provides:
 - a. Designed for disconnected environments.
 - b. Programming model with XML support.
 - c. - Provides a set of classes, interfaces, structures that manage data access from the .NET framework.
 - d. Enables us to connect to SQL server database.
3. Using stored procedures that facilitate the manipulation process of data by crating SQL queries and for the following benefits:
 - Modular Programming.



- Distribution of work.
 - Increased database security.
 - Faster Execution.
 - Reduces network traffic.
 - Provides flexibility.
4. Using Multimedia programs for the following:
- i. Create interactive and tolerant interface that provide friendly relationship between the student and portal.
 - ii. Create a user guidance interface to provide clarity and assistance by using appropriate colors.
 - iii. Allow for integration between existing systems and future systems and communication between them.

1.3 Functional description

student function description:

1. Provide a private account for each student by his username and password.
2. My profile that provide personal information about each student such as student name, address and simple amount of academic information.
3. Showing the current schedule for the logging student that explains enrolled courses in addition of related information.
4. Showing the financial information for each student including financial aids, balance credit and debit, semester balance.
5. Academic status that include academic report that allow the student to see his academic information.
6. Semester status that gives each student the academic status for the current semester only.



7. Showing the passed courses for each student from the beginning until the current semester, however, the student can filter these courses according the following criteria:
 - ▶ Year.
 - ▶ Semester.
 - ▶ Course Type (selective course or elective course).

8. In our portal we have offer a variety of bulletin boards which fall in on of the following divisions:
 - Student boards: here the student can show the announcement that relate to him only no other students can show it.
 - Course boards: these boards devoted for each course so, each student currently taking the course can show it.
 - Department boards: offered for all students within the department or faculty so, each student belongs to the department can show it.
 - College boards: specialized for making board grouping by each college in the university. Consequently, only student who belongs to that university can show it.

9. New boards: this function shows the latest boards that issued within a given time period that inserted by the student, the default value for this period is three days.
10. Changing the password: from this area the student can enter a new password according his desire if he feels that his or her password has gotten by the others.
11. Voting: area from which student can vote on a given question.
12. My style: used to enable student to customize his account appearance.

Administrator function description:

1. Provide a private account for each administrator by his username and password.



2. Insert boards: this process enables administrator to insert new boards for students, courses, departments or for colleges.
3. Delete boards: enables administrator to delete expired boards, or delete board by selecting its name or type.
4. Update boards: enables administrator to make some justifications or changes to either deadline date or body.
5. Insert marks: allows administrator to insert marks for each course by choosing its number.
6. Insert new college: from here the administrator can add new college to the portal and can add information about it.
7. Insert new major: this process devoted to enable administrator to add new major in a specific department.
8. Questionnaire analysis: this process enable administrator to gain control on provided questions supported by him to students in addition to the previous question analysis.
9. Control panel: this function allow administrator to activate or deactivate links in student or instructor main menu or in overall portal links.
10. User's accounts: this process aimed at simplifying administrator access to user accounts in all types, students, instructors.
11. Change Password Form: allow administrator to change his password by providing new password and confirm it.

Instructor function description:

1. Provide a private account for each instructor by his username and password.
2. Insert boards: this process enables instructor to insert new boards for students and courses.
3. Change Password Form: allow instructor to change his password by providing new password and confirm it.



4. My regular students: allows instructor to list all student for a specific course who teach it by selecting course number.
5. Voting: area from which instructor can vote on a given question.

Public function description:

1. Boards marquee: is a horizontal marquee that shows all recent college and department boards since seven days.
2. Honor boards: is a vertical marquee that displays all honored students during the current semester.
3. Forget my password: here, each student and instructor can obtain his password by portal by inserting ID and answer some questions.
4. Search engine: this search limited to portal database that aim to find courses, colleges, majors, employees or boards stored in database by inserting the search item or part of it.
5. PAP links:
 - Bulletin boards.
 - Get my account online.
 - Academic calendar.
 - Graduate projects.
 - Graduated students.
 - PPU WAN diagram.
6. General activities: list all events that take place in Polytechnic University such as exhibits.
7. FAQs : proposed questions from students point of view, the portal try to answer them.
8. PPU WAN: this diagram explain how PPU network zones connected to each other.
9. Site MAP: this link redirect the user to a form that displays a map that navigate web forms interconnections and relationships.
10. Help: enables PAP visitor to familiar with portal functions and use.



1.4 Non-functional description:

Non-functional description defines properties of the system and constraints on the services offered by the system. These properties include the following:

1.4.1 Product Requirements:

1. Maintainability and scalability which enforce the system to coexist with any new changes.
2. Reliability of the system so, processes designed in such a way that prevent errors occurrence before they causing large problems.
3. Robustness that enables the system to continue its functionality in spite of unexpected problems.
4. Safety: the portal must have the ability to prevent illegal or incorrect operations from users or students by using certain tools such as validation controls.
5. Understandability that make it easy for students to use and deal with.
 - User friendly by developing good interface.
 - Data accessibility must be easy.
6. Must operate on windows platform environment.
7. Browser independent
8. Secure and private.
9. The portal developed for the purpose of supporting integration between existing and future systems, and it must have then ability to work with old systems for some period of time.

1.4.2 Process requirements:

The portal project and its documents must be delivered on 11 June 2005.

1.5 Project constraints:

1. Each student must have username and password.
2. Administrator must have username and password.



3. Password must be encrypted.
4. Password must be more than six characters (alpha numeric mixture is preferred).
5. The system must be flexible and adaptable with any preferred future expansion.
6. Only the administrator has the ability to add, delete, and update data related to him.
7. End user must have suitable browser and applications to see and browse the web page in the best manner.

1.6 Feasibility study:

We have found that we can go through developing the portal system, this section will discuss in detail the alternatives, economics feasibility, cost benefit analysis and risk evaluation.

1.6.1 Alternatives:

1. We have to choose one of two available approaches in developing our portal system; these two approaches about the type of network technology that will be used.

- a. Intranet methodology:

An Intranet is a network based on the internet TCP/IP open standard. An intranet belongs to an organization; Authorized users outside your company such as your remote workers, Intranets are used to share information. Secure intranets are much less expensive to build and manage than private.

Advantages:

- ✓ High speed.
- ✓ Shares software between peripherals.



- ✓ Provides backup in a real time systems.
- ✓ Low cost
- ✓ Decentralization.
- ✓ Support high level of security and reliability.

Disadvantages:

- ✓ Internal access only within the organization.
- ✓ Coverage area is limited.

b. Using the Internet and World Wide Web:

Internet is a massive network of networks that connects millions of computers together globally forming network in which any computer can communicate with any other.

Advantages:

- ✓ There is no limit on size.
- ✓ Support student-centered environment.
- ✓ Adapts to students learning styles.
- ✓ Covers the entire world without any restrictions.
- ✓ An easy way for two extremes to interact with each others.

Disadvantages:

- ✓ Duplication of information.
- ✓ Rational:
 - No clear web standards.
 - Anyone can publish anything.
- ✓ Accuracy:



- Information does not reliable all the time.
 - There is no editor or someone who checks information.
- ✓ The problem of authority and security.

By comparing and analyzing these two approaches we choose to work with the web-based methodology to enable students to see their information from outside environment of the organization.

2. To deal with data we have tow alternatives which are:

- a. SQL server 2000: is a full featured relational database management system for data storage and manipulation with support for robust server side development and data replication.

Advantages:

- ✓ Each data item is stored in a central location where all users can work with it.
- ✓ Business and security rules can be defined one time on the server and enforced equally among all users.
- ✓ A relational database server optimizes network traffic by returning only the data an application needs
- ✓ Hardware costs can be minimized.
- ✓ Maintenance tasks such as backing up and restoring data are simplified because they can focus on the central server.
- ✓ The capability to search and sort data and find information quickly.

Disadvantages:

- ✓ A License needs to be purchased for SQL Server.



- ✓ If data is not stored on an existing server; an additional server will need to be purchased to run SQL Server.

- b. Microsoft Access: Access is a database management program that allows you to store, retrieve, analyze, and print.

Advantages:

- ✓ The capability to search and sort the data and find information quickly.
- ✓ Easy for beginners to work with it.
- ✓ Need low level of computer resources.

Disadvantages:

- ✓ Low security level when compared with SQL server 2000.
- ✓ Updating data is complex process.

After previous analysis we choose to work with SQL server 2000.

- 3. for implementing and developing the system we have tow available alternatives which are:

- a. Visual Studio.NET: simplifies the development of powerful, reliable enterprise web solutions and increases developer efficiency by providing a familiar, shared development environment.

Advantages:

- ✓ One IDE for multiple languages and multiple project types.
- ✓ Multiple languages within a project.
- ✓ Multiple project types within a solution.
- ✓ Integrated browser.
- ✓ Debugging support.



- ✓ Customizable interface.
- ✓ High security supported.
- ✓ Deals with XML format.

Disadvantages:

- ✓ Requires high processor and memory speeds.
- ✓ Difficult for beginner programmers to work with it.

b. Microsoft front page: used to create traditional static web pages.

Advantages:

- ✓ Ease of use.
- ✓ Requires low level of computer capabilities.

Disadvantages

- ✓ Low security level when compared with ASP.NET as a huge technology.
- ✓ Creates a static web based.

After analysis the first choice will be selected.

1.6.2 Cost-benefit analysis:

We have found that the system includes the following aspects for cost-benefit analysis:

1. economic aspect:

Explores project hardware, software and human costs that will be discussed later.

2. Technology aspect:



All required programs and software that explained in the feasibility study are available:

- ▶ Visual Studio.NET
- ▶ SQL server 2000.
- ▶ Microsoft office.
- ▶ Multimedia programs.

3. Legislation aspect: this project will be developed for Palestine Polytechnic University (PPU), and the university agrees on this project.

1.6.3 Evaluation of risks :

This section will explain some potential risks that may face the functionality of our PAP system attached with our proposed solutions.

Risks:

1. Students may resist our Portal as a new way in providing services to them and changing the traditional way. The proposed solution: make some training programs to make students familiar with the portal.
2. Many students may not have internet service. The proposed solution: the university can provide them with internet service.
3. PAP may attacked by viruses. The proposed solution: the use of antivirus programs.
4. The occasional damage of the system hardware. The proposed solution: the use of backup copies of data.
5. Heavy load on the main server may occur. The proposed solution: we can use multiple servers as a distributed processing.



1.7 Economic Study:

1) for development:

Hardware:

Item	Number of units	Cost per unit	Cost of units
Desktop computer P4, 256MB RAM, HD 20 GB, CPU with 2.4 GHz speed.	3	\$500	\$1500
Flash Memory 256MB	3	\$30	\$90
Scanner	1	\$140	\$140
Laser HP printer	1	\$120	\$120
Total			\$1850

Table (1.1) Development Hardware cost.

Software:

Item	Number of units	Cost per unit	Cost of units
Microsoft Windows XP	1	\$150	\$150
Visual Studio.Net	1	\$900	\$900
Microsoft Office XP	1	\$270	\$270
PhotoShop	1	\$70	\$70
Photo Impact	1	\$60	\$60
SQL Server 2000	1	\$120	\$120
Total			\$1570

Table (1.2) development software cost

***Human:***

Employee	Cost/month
Web designer	\$800
Database developer	\$800
System programmer	\$800
Total	\$2400

Table (1.3) Costs of human resources.

2) For Implementation:

Hardware:

Item	Items No	Cost/Item	Cost
Server	1	\$1750	\$1750
Monitor 17inch	1	\$110	\$110
Peripherals	1	\$90	\$90
Total			\$1950

Table (1.4) Hardware Cost for Implementation.

Software:

Item	Items No	Cost/Item	Total
Windows Server 2003	1	\$450	\$450
MS SQL Server 2003	1	\$230	\$230
Norton Antivirus 2003	1	\$180	\$180
Total			\$860

Table (1.5) Software cost Implementation.

***Human:***

Employee	No	Monthly cost
Administrator	1	\$900
Assistance	1	\$700
Total		\$1600

Table (1.6) Human recourses cost for implementation.

The overall costs for the Portal system summarized in the following table:

1)	For development:	
a)	Hardware:	\$1850
b)	Software:	\$1570
c)	Human:	\$2400
Total Cost		\$5820

Figure (1.7) The overall development costs summery.

2)	For Implementation:	
a)	Hardware:	\$1950/Month
b)	Software:	\$860/Month
c)	Human:	\$1600/Month
Total Cost		\$4410/Month



Figure (1.1) Gantt chart for time schedule.



2.1 Introduction:

This chapter will explore in detail the system requirements specifications, functional description, validation criteria that contain data flow diagram (DFD), data structure presentation, data dictionary and database requirement.

2.2 Requirements specifications:

A. Student requirements specifications:

- The student requests the PAP site from web server using internet explorer.
- The students insert his username and password at log in area in the log in Page.
- If the student inserts a valid username and password he will see the student main menu that contains the following items:

1. The student My Profile:

By clicking on My Profile in students main menu the student will show a new page that contains his information as follow:

- Name information: first, second, third and final name.
- Address information: address and telephone number
- Academic information: academic number, scientific degree, major and outline.
- Add my picture: if the student wants to insert his picture to database for using it in the portal.
- Add my document: if the student was graduates he will be able to upload his project document over portal to benefit the other student (under graduate) from this document.

2. the student current schedule:

By clicking on the current schedule, the student can see the following :

- Course number.



- Current semester.
- Course name.
- Instructor.
- Section.
- Time.
- Room number.

3. the student financial status:

When the student chooses to click on the financial status he will see his related financial information that contain the following:

- Financial aids: aid, scholarship and other.
- Financial balance: credit and debit.
- Semester balance.
- Total balance.
- In addition to that semester classification and academic year information shown on the top of the page.

4. the student academic status:

This section shows in detail all academic information for each student from the beginning until the current semester. When the student chooses academic status from the main menu he will show the following report:

- Student full name.
- Birth date.
- Student number.
- Degree: defines the scientific degree of the student.
- Major: defines the student major.
- Tawjihi information: branch and average.
- Community work hours: show passed and remained work hours.
- Department: defines the student's faculty.
- College: specifies the student's college.
- Averages: displays accumulative and major averages.



- Academic alerts: specifies how many academic warnings the student has gained.
- Dismiss: this item tells in yes or no if the student dismissed or not.
- Delay: specifies if the student delay for the current semester or not.
- Level: defines the student level in semesters.
- Regular: specifies if the student is regular or not.
- Graduated: specifies if the student is expected to graduate or not.

5. Semester status:

If the student clicks on this link then he can see his marks only for the current semester only in a table that contains the following:

- Course number.
- Course name.
- Course mark.

In addition to that, he can see his semester average and if he honored or not.

6. Passed courses:

By clicking this link, the student can show all of his passed courses since starting his study until the current semester and he can display these courses in three different ways:

- By filtering courses according study year. So only the courses studied at that year displayed by choosing the year from dropdown list that shown automatically after clicking the year checkbox.
- By filtering courses according semester or according a combination of year and semester. So we can select the year and semester from dropdown lists that shown automatically. After that, the passed courses can be displayed by applying the filtering condition.
- By filtering according the type of the course (college requirement, department requirement, elective, selective) and then display these



courses in combination of type and year or type and semester or type and (year and semester).

The student can see some statistics such as:

- ✓ Accumulative average.
- ✓ Passed courses.
- ✓ Credit hours remained.

After performing any of previous filtering criteria, all resulted courses will be displayed in the list below. This list contains

- Course number.
- Course name.
- Mark.
- Course type.
- Credit hours.
- Semester.
- Academic year.

7. Student boards.

This type of boards devoted to offer special area to display each student boards only. So no other students can see these boards or memos. The board contains:

- Board number.
- Student number.

8. courses boards:

In this section the student can show all of his courses boards and he can filter these boards by their date and he can specify what the course board he want. Consequently, he can show it or show all the boards.



9. department boards:

Here we can tell the student about his department boards, the portal enables the student to pick his department and then show the board. Here the student he can put the number of days ago to see only the boards issued within this time period.

10. college boards:

here the student show his college related boards by selecting the college name from a dropdown list, and can show only the boards within a specified period of time.

11. change password:

From this area the student has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

12. Questionnaire:

From the student main menu page, student can show the question to be answered and he can show the questionnaire results and he can answer in yes, no or may be.

13. New boards

In this section, the student can see all recent boards types related to him. By default these boards since three days ago and the student can insert the number of days ago to check the issued boards within this time period.

14. Social guide center.

This section devoted to social support system for students; here the administrator (social guide) can provide the students with solutions and suggestions and focus the light on sensitive problems that the students face in their academic life.



15. Documents.

From here the student can print his academic status with finished hours and marks, and he can print paper that insure that this students is regular students in the university.

16. My style.

This function allow the student to customize his account interface, by picking his preferred style from choices and make apply, then after refresh the page the new setting will apply.

17. Graduate Projects.

From here the student can show information about all graduate projects that done by university students in different majors, and he can download the attachment if it found.

18. PPU WAN Diagram.

This page explains the PPU WAN network diagram and the interconnections between them.

B. administrator requirements specifications:

1. insert boards:

Provide administrator the ability to insert boards. In this section the administrator can insert new boards by providing the primary board data that include the following:

- Date.
- Due-to date.
- Board title.
- Board number.
- The author.
- The board text.



- The board attachment.

This section has many tools to facilitate the insertion process; if the board body saved at any location the administrator can brows it as any file instead of typing it, then he can specify the board type.

2. Delete boards:

Provides administrator the ability to delete boards. This form specialized to enable the administrator to delete expired boards that exceeded the specified due-to date.

By selecting the expired board's radio button, board name and type and then click on get boards' button to attach these boards in a dropdown list called target boards, after that he can put the intended boards to be deleted in a new list called boards to delete and insure the deleting process by clicking the delete button.

3. Update boards:

Provides the administrator the ability to update boards; This form used to make some editing to the boards by selecting the board name from the available boards and perform the update process to body and deadline date only. Finally, submitting the process by clicking change button.

4. Insert Marks:

Provides administrator the ability to insert marks. Here the administrator can show courses that offered at this semester, and then he select any course to get the sections of that course, and display them in drop down list, and then he select any section to view the students enrolled at that course, and then select any student to insert new mark or update the existing mark related to him.

5. Insert new college.

This form used in the case of university expansion and a new college is opened. Here administrator can add a brief description for the new college



and the college number determined automatically and he can insert the college name and click insert button to save the information in the database.

6. Insert new major.

This form enable the administrator to insert new major at any college or department, a brief description can be added and the primary major information can be added too such as major number, major name, credit hours and outline year, the process completed by clicking on the insert button to save data in the database.

7. Provides the administrator the ability to add general question for voting.

The student can answer in yes, no or maybe.

8. From the administrator main menu, administrator can provide students with the questionnaire results and voting analysis that often come from database and calculated automatically, the student can vote one time only.

9. Control panel.

From this link the administrator gain control over all the portal processes, links and forms, he can activate or deactivate any link or control in the students accounts.

10. Users accounts.

In this form administrator should select the PAP user type and then he will show all the authenticated users with their ID's and their usernames.

11. Change password.

From this area the administrator has the ability to insert his username and change his password and replace it by a new one and confirm it. Then click update button to complete the changing process.



C. Instructor requirements specification:

1. Insert Boards

Provide instructor the ability to insert his course's boards and his student's boards only. In this section the instructor can insert new boards by providing the primary board data that include the following:

- Date.
- Due-to date.
- Board title.
- Board number.
- The author.
- The board text.
- The board attachment.

This section has many tools to facilitate the insertion process; if the board body saved at any location the instructor can brows it as any file instead of typing it, then he can specify the board type.

2. Insert Marks

Here the instructor can show only courses that he give, and then he select any course to get the sections of that course that he gives and display them in drop down list, and then he select any section to view the students enrolled at that course, and then select any student to insert new mark or update final course mark related to him.

3. My regular students.

In this form the instructor can view report that contain names, courses, and sections that related to instructor course, and he can print it.



4. Chang my password.

From this area the instructor has the ability to change his password and replace it by a new one and confirm it. Then click update button to complete the changing process.

D. Public requirements specifications:

This requirement does not depend on the student or administrator accounts on other words it is visible and accessed to any portal visitor.

This requirement includes:

1. Log in area devoted for administrators and students that enter the portal by using valid username and password; in addition to that these can remember their password and username by clicking forget my password link.
2. Help: this link used to conduct the portal visitors how to benefit form services and facilities provided by the portal.
3. PAP links: this section includes:
 - Bulletin boards: this link leads the user to view all public university boards, and he can show details about each board by clicking on read link.
 - Graduate projects: displays all graduate projects in a specific major by choosing major name, then all related projects appear in a table.
 - Get my account on line: here an authenticated user can retrieve his account by asking him some questions. The system asks the user to enter his username. If he authenticated, the system asked him some additional questions for supporting security, if his answers correct then the system gives him his account information.



- Academic calendar: in this area the user can see the user, this academic year calendar that contains all holydays and occasions during academic year.

4. Search engine:

This search engine limited to courses, colleges, majors, employees and boards, at the first, the student should select what he going to search from dropdown list then type a word or statement who looking for. Finally click on portal search button, this search limited on portal database.

5. PAP FAQs:

This link used to serve students with dynamic help that provided by portal specialists that including common Frequently Asked Questions and try to answer them.

6. General activities:

This area contains current general activities take place or expected to occur.

7. Honor board:

This corner displays scrolled marquee with links names of honored students. To show details of each student you have to click on name link.



2.3 Functional description:

This section explores all the portal functions in detail.

1. The user requests the PAP home page from web server.

Function: user requests the page from web server.

Description: this function provides ability to browse the portal.

Input: the portal URL

Source: user

Output: Portal Home page.

Destination: web server.

Require: insert correct site address.

Pre-condition: availability of internet service.

Post-condition: displaying the portal Home Page.

Figure (2.1) Request home page.



2.Log in as student.

Function: student log in.

Description: enabling the student to access his account, see his information and use the student menu using valid username and password.

Input: student username and password

Source: student and log in form.

Output: The student main menu.

Destination: SQL database server.

Require: valid log in and correct username and password.

Pre-condition: Home Page and disconnected student.

Post-condition: connected student and student menu displayed.

Figure (2.2) Student login.



3. Log in as administrator.

Function: log in as administrator.

Description: identifying the administrator by using valid username and password.

Input: administrator username and password

Source: administrator and log in form.

Output: administrator main menu.

Destination: SQL database server.

Require: valid log in and valid username and password.

Pre-condition: Home Page and disconnected student.

Post-condition: connected student and student menu displayed.

Figure (2.3) Administrator login.



4.The student my profile.

Function: displaying the student information

Description: by clicking the my profile link from the main menu a new page will be displayed and contained the student information

Input: clicking my profile link from student main menu.

Source: the student and student menu web form.

Output: the student profile.

Destination: SQL database server.

Require: valid log in and single click on my profile item.

Pre-condition: Home Page and no student displayed in main menu.

Post-condition: the student can see his profile.

Figure (2.4) My profile.



4.1 upload picture.

Function: upload picture.

Description: this function used to give the student the ability access the database for the purpose of saving his picture in database to be used by the portal administrator and show it with the student related activities such as honor board.

Input: click on brows button to specify image location.

Source: client primary or secondary memory.

Output: saved picture in the portal database.

Destination: SQL database server.

Require: single click on brows button.

Pre-condition: specifying the picture location.

Post-condition: saved picture in the portal database.

Figure (2.5) Upload student picture.



5.the student current schedule

Function: current schedule.

Description: this function enables the student to show his registered courses and their time schedule, in addition to room number, section and course instructor.

Input: clicking current schedule for student main menu.

Source: the student and student main menu web form.

Output: current schedule web form.

Destination: SQL database server.

Require: valid log in and single click on the current schedule item.

Pre-condition: Home Page and no schedule information displayed.

Post-condition: the student can see his current schedule.

Figure (2.6) Current schedule for student..



6.financial status

Function: display student's financial status.

Description: this function displaying the student financial status by clicking on financial status he will show financial information related to the student.

Input: clicking financial status from the student main menu.

Source: the student and student main menu web form.

Output: financial status web form.

Destination: SQL database server.

Require: valid log in and single click on the financial status item.

Pre-condition: Home Page and no financial information displayed.

Post-condition: connected student and student menu displayed.

Figure (2.7) Student financial status.



7.Academic status.

Function: displaying academic status.

Description: this function enables the student to show detailed academic report of academic information from the beginning until the current semester.

Input: clicking academic status from student main menu.

Source: the student and student main menu web form.

Output: academic status report.

Destination: SQL database server.

Require: valid log in and single click on academic status in student main menu.

Pre-condition: Home Page and no academic information displayed.

Post-condition: connected student and student menu displayed.

Figure (2.8) Student Academic status.



8.semester status

Function: displaying semester status.

Description: this function enables the student to see courses and their marks for latest semester.

Input: clicking semester status from student main menu

Source: the student and student main menu web form

Output: semester status report.

Destination: SQL database server.

Require: valid log in and correct username and password.

Pre-condition: Home Page and no semester information displayed.

Post-condition: connected student and semester status displayed.

Figure (2.9) Semester status.



9.Passed courses.

Function: displaying passed courses.

Description: this function enables the student to show his passed courses from the moment of entering the university until the latest semester.

Input: click on passed courses link from student main menu.

Source: student and his main menu web form.

Output: report of passed courses and some other information.

Destination: SQL database server.

Require: valid log in and single click on passed courses item.

Pre-condition: Home Page and no passed courses information displayed.

Post-condition: table of passed courses displayed.

Figure (2.10) Passed courses.



9.1 Filtering criteria.

Function: select filtering criteria.

Description: this function used to show specific courses according some condition; such as filtering courses according study year. So only the courses studied at that year displayed by choosing the year from dropdown list that shown automatically after clicking the year checkbox.

Input: select filtering method in year, semester or course type and pick the value of selected method from dropdown list that becomes visible automatically.

Source: student and his passed courses web form.

Output: table of passed courses and some other information.

Destination: SQL database server.

Require: selecting filtering criteria.

Pre-condition: student menu web form.

Post-condition: table of passed courses displayed.

Figure (2.11) Filtering criteria.



10.My boards.

Function: displaying student boards.

Description: this function specialized to offer special area that display each student boards only, so no other students can see these boards.

Input: click on my boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on my boards item in the students main menu.

Pre-condition: single click on my board's item in student's main menu.

Post-condition: table containing student boards

Figure (2.12) Student Boards.



11. Course boards.

Function: displaying course boards.

Description: In this section the student can show all of his courses boards and he can filter these boards by their date and he can specify what the course board he want. Consequently, he can show it or show all the boards.

Input: click on course boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, author, subject, course name and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on course boards item in the students main menu.

Pre-condition: single click on course boards item in student's main menu.

Post-condition: table containing course boards

Figure (213) Courses boards.



12.Department boards.

Function: display department boards.

Description: In this section the student can show all of his department boards and he can filter these boards by their date and he can specify what the department board he wants.

Input: click on department boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, subject, department name, author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on department boards item in the students main menu.

Pre-condition: single click on department boards item in student's main menu.

Post-condition: table containing department boards

Figure (2.14) Departments boards.



13.College boards.

Function: display college boards.

Description: In this section the student can show all of his college boards and he can filter these boards by their date and he can specify what the college boards he wants.

Input: click on college boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, subject, college name, subject, college name author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on college boards item in the students main menu.

Pre-condition: single click on college boards item in student's main menu.

Post-condition: table containing college boards.

Figure (2.15) Colleges boards.



14.Change password.

Function: change password.

Description: In this function the student has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

Input: click on change password in student's main menu.

Source: student and his main menu web form.

Output: change my password web form that contains a small frame including textboxes to insert old password, new password and then confirm it.

Destination: change my password web form.

Require: valid log in and single click on change password item in the students main menu.

Pre-condition: single click on change password item in student's main menu.

Post-condition: change password web form.

Figure (2.16) Change password..



14.1 Update changed password process.

Function: Update change password process.

Description: this sub-function enables the student to confirm the changing process by clicking update button to save new password in database.

Input: click on update password button in change my password web form.

Source: student and change my password web form.

Output: new changed password saved in database.

Destination: SQL database server.

Require: inserting old password correctly.

Pre-condition: filling old password, new password textboxes correctly.

Post-condition: new password saved in database.

Figure (2.17) Update changed password process.



15. Questionnaire.

Function: General voting.

Description: From the student main menu web form, student can show the question to be answered and he can show the questionnaire results and he can answer in yes, no or maybe.

Input: select answer by click on some radio button.

Source: student and his main menu web form.

Output: save answer in the database.

Destination: SQL database server.

Require: choose answer.

Pre-condition: entering question for voting and answer it.

Post-condition: save the result in database.

Figure (2.18) Questionnaire.



16. New boards.

Function: new boards.

Description: In this function, the student can see all recent boards types related to him. By default these boards since three days ago and the student can insert the number of days ago to check the issued boards within this time period.

Input: number of days and click on generate button.

Source: student and his main menu web form.

Output: issued boards within the specified time period.

Destination: SQL database server.

Require: filling since number of days textbox.

Pre-condition: student menu web form.

Post-condition: student menu web form with intended boards.

Figure (2.19) New boards.



17. Social guide center.

Function: social guide center.

Description: This function devoted to social support system for students; here the administrator (social guide) can provide the students with solutions and suggestions and focus the light on sensitive problems that the students face in their academic life.

Input: single click on the question link.

Source: Administrator.

Output: proposed answer.

Destination: SQL database server.

Require: choosing the question.

Pre-condition: student menu web form.

Post-condition: explanation web form.

Figure (2.20) Social guide center.



18. Documents.

Function: this function enables the student to print some documents.

Description: From here the student can print his academic status with finished hours and marks, and he can print paper that insures that this student is regular students in the university.

Input: single click on the document name link.

Source: Administrator.

Output: document body.

Destination: SQL database server.

Require: choosing document name.

Pre-condition: student menu web form.

Post-condition: document displayed.

Figure (2.21) Students documents.



19. Insert boards.

Function: Insert boards.

Description: this function Provides administrator the ability to insert boards.

In this section the administrator can insert new boards by providing the primary

Input: click on insert board link from administrator main menu and fill Date, Due-to date, Board title, Board number and the author textboxes and the board text.

Source: Administrator and insert new board web form.

Output: save the board and its information in database in the database.

Destination: SQL database server.

Require: choose board body location type new one in board body text.

Pre-condition: fill all textboxes and check board body radio button, then type the board body and click insert button.

Post-condition: saved board in database.

Figure (2.22) Insert boards



20. Delete boards.

Function: Delete boards.

Description: this function Provides administrator the ability to delete boards. This form specialized to enable the administrator to delete expired boards that exceeds the specified due-to date.

Input click on delete board link from administrator main menu and check the radio button to define the condition that the board will deleted according it.

Source: Administrator and delete boards web form.

Output: deleted board and its information from database.

Destination: SQL database server.

Require: select board type.

Pre-condition: get boards in target board's drop down list and then specify which expired boards to delete.

Post-condition: deleted board from database.

Figure (2.23) Delete boards.



20.1. Get boards.

Function: Get boards.

Description: this function attaches expired boards to be displayed in target boards drop down list.

Input: check on expired boards, board name or board type radio buttons.

Source: Administrator and delete boards web form.

Output: display all expired boards.

Destination: boards to delete drop down list

Require: select board type.

Pre-condition: click on delete board link from administrator main menu.

Post-condition: select boards to delete and click delete button to complete the process.

Figure (2.24) Get boards.



20.2. Browse boards.

Function: Brows boards.

Description: this function used if the board body is saved on the local server, so administrator can brows the board instead of typing it again.

Input click on brows button to specify path of board location.

Source: Administrator and delete boards web form.

Output: the path of the board filled in path textbox.

Destination: path textbox.

Require: select board type.

Pre-condition: get boards in target board's drop down list and then specify which expired boards to delete.

Post-condition: deleted board from database.

Figure (2.25) Browse boards.



21. Update boards.

Function: Update boards.

Description: Provides the administrator the ability to update boards; this form used to make some editing to existing boards.

Input: click on update board link from administrator menu.

Source: administrator and update boards form.

Output: updated board.

Destination: SQL database server.

Require: administrator should identify what he going to update and specify the board location if it exist.

Pre-condition: select board name from all available boards.

Post-condition: confirm the process by pressing on change button.

Figure (2.26) Update boards.



22. Insert Marks.

Function: Insert marks.

Description: this function Provides administrator the ability to insert marks of courses that offered at the end of current semester.

Input: click on insert marks link from administrator menu and choose the course number and student number then insert marks.

Source: administrator and insert marks web form.

Output: save courses marks for each student registered in the previous specified course.

Destination: SQL database server.

Require: select course number.

Pre-condition: select course number and student number.

Post-condition: displaying students and insert marks then submit the process.

Figure (2.27) Insert marks.



23. Add new college.

Function: Add new college.

Description: Here administrator can add a brief description for the new college and the college number determined automatically and he can insert the college name and click insert button to save the information in the database.

Input: click Add college link from administrator menu and college name and description but the college number determined automatically.

Source: administrator and add new college web form

Output: save college information in colleges table in the database.

Destination: SQL database server.

Require: college number and name.

Pre-condition: click on add college from administrator menu.

Post-condition: save college information in database by clicking insert button.

Figure (2.28) Add new college.



24. Insert new major.

Function: Insert new major.

Description: This form enables the administrator to insert new major at any college or department, a brief description can be added and the primary major information can be added too.

Input: major number, name, number of credit hours, outline year and department name.

Source: administrator and insert new major web form

Output: save major information in majors table in the database.

Destination: SQL database server.

Require major number, name, number of credit hours outline year and department name.

Pre-condition: click on insert new major from administrator menu.

Post-condition: save major information in database by clicking insert button.

Figure (2.29) Insert new major.



25. Questionnaire analysis.

Function: Questionnaire analysis.

Description: Administrator can provide students with the questionnaire results and voting analysis that often come from database and calculated automatically, the student can vote one time only.

Input: click questionnaire analysis link from administrator main menu.

Source: administrator and questionnaire analysis web form.

Output: display questionnaire result.

Destination: SQL database server.

Require: None.

Pre-condition: entering question for voting.

Post-condition: upload the question.

Figure (2.30) Questionnaire analyses.



26. Control panel.

Function: Control panel.

Description: From this link the administrator gain control over all the portal processes, links and forms, he can activate or deactivate any link or control in the students accounts.

Input: student, instructor or portal settings.

Source: administrator and control panel web form.

Output: activate or deactivate student, instructor settings and other portal links and processes.

Destination: student and instructor main menus and d other portal settings.

Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

Figure (2.31) Control panel.



26.1. Apply student settings.

Function: Apply student settings.

Description: this function used to apply settings that selected for manage student account to allow or deny him from performing some activities.

Input: select setting.

Source: administrator and control panel web form.

Output: activate or deactivate student settings.

Destination: student main menu.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

Figure (2.32) Applying student settings.



26.2. Apply instructor settings.

Function: Apply instructor settings.

Description: this function used to apply settings that selected for manage instructor account to allow or deny him from performing some activities.

Input: select setting.

Source: administrator and control panel web form.

Output: activate or deactivate instructor settings.

Destination: instructor main menu.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

Figure (2.33) Applying instructor settings.



26.3. Apply Portal settings.

Function: Apply Portal settings.

Description: this function used to apply settings that selected for manage portal such as delete expired boards automatically or allow browsing the Portal inside the university only.

Source: administrator and control panel web form.

Output: activate or deactivate Portal settings.

Destination: Portal settings.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

Figure (2.34) Applying portal settings.



28. Change password.

Function: change password.

Description: In this function the administrator has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

Input: click on change password in administrator main menu.

Source: student and his main menu web form.

Output: change my password web form that contains a small frame including textboxes to insert old password, new password and then confirm it.

Destination: Change password web form.

Require: valid log in and single click on change password item in the administrator main menu.

Pre-condition: single click on change password item in administrator's main menu.

Post-condition: change password web form.

Figure (2.35) Change password.



29. Insert boards by instructor.

Function: Insert boards.

Description: this function Provides instructor the ability to insert boards. In this section the instructor can insert new boards by providing the primary board data, the board type might be student or course only.

Input: click on insert board link from instructor main menu and fill Date, Due-to date, Board title, Board number and the author textboxes and the board text.

Source: Administrator and insert new board web form.

Output: save the board and its information in the database.

Destination: SQL database server.

Require: choose board body location type or write new one in board body text.

Pre-condition: fill all textboxes and check board body radio button, then type the board body and click insert button.

Post-condition: saved board in database by clicking insert button.

Figure (2.36) Insert board by instructor.



30. Insert Marks by instructor.

Function: Insert marks.

Description: this function Provides instructor the ability to insert marks by himself of courses that offered by him only.

Input: click on insert marks link from instructor menu and choose the course number and student number then insert marks.

Source: instructor and his insert marks web form.

Output: save courses marks for each student registered in the previous specified course.

Destination: SQL database server.

Require: select course number.

Pre-condition: select course number and student number.

Post-condition: displaying students and insert marks then submit the process.

Figure (2.37) Insert mark by instructor.



32. My regular students.

Function: My regular students.

Description: In this function instructor has the ability to generate list of his regular student in one of his courses.

Input: course number.

Source: instructor and his main menu web form.

Output: list of regular students.

Destination: SQL database server.

Require: valid log in and single click on my regular students item in the instructor main menu.

Pre-condition: single click on change password item in instructor's main menu.

Post-condition: list of regular students.

Figure (2.38) My regular students



33. Log in as instructor.

Function: log in as instructor.

Description: identifying the instructor by using valid username and password.

Input: instructor username and password

Source: instructor and log in form.

Output: instructor main menu.

Destination: SQL database server.

Require: valid log in and valid username and password.

Pre-condition: Home Page and disconnected instructor.

Post-condition: connected instructor and his main menu displayed.

Figure (2.39) Login as instructor.



34. Help.

Function: Help.

Description: this link used to conduct the portal visitors how to benefit form services and facilities provided by the portal.

Input: single click on help link on top of home page.

Source: Any PAP visitor.

Output: help content.

Destination: SQL database server.

Require: None.

Pre-condition: single click on help link.

Post-condition: help content.

Figure (2.40) Help.



35. Search engine.

Function: search engine.

Description: This search engine limited to courses, colleges, majors, employees and boards, this search limited on portal database.

Input: complete or part of the subject name.

Source: Any PAP visitor.

Output: table of required search.

Destination: SQL database server.

Require: complete or part of the subject name.

Pre-condition: click portal search.

Post-condition: search content.

Figure (2.41) Search engine.



36. FAQs.

Function: FAQs.

Description: This link used to serve students with dynamic help that provided by portal specialists that including common Frequently Asked Questions and try to answer them..

Input: single click on the question.

Source: Any PAP visitor and FAQs section of home page.

Output: suggested answer.

Destination: SQL database server.

Require: None.

Pre-condition: click the question.

Post-condition: the answer.

Figure (2.42) FAQs



37. General activities.

Function: general activities.

Description: This area contains current general activities take place or that expected to occur.

Input: single click on activity name.

Source: Any PAP visitor and general activities section of home page.

Output: the activity illustration.

Destination: SQL database server.

Require: None.

Pre-condition: click the activity name.

Post-condition: detailed activity explanation.

Figure (2.43) General activities.



38. Honor board.

Function: honor board.

Description: This corner displays scrolled marquee with student name links of honored students. To show details of each student you have to click on name link.

Input: single click on the honored student name.

Source: Any PAP visitor.

Output: table of student name, major and accumulative average.

Destination: SQL database server.

Require: None.

Pre-condition: click the student name name.

Post-condition: table of student name, major and accumulative average.

Figure (2.44) Honor board.



39. Boards marquee.

Function: board's marquee.

Description: this function used to show the latest boards for publics since seven days.

Input: single click on the board name.

Source: Any PAP visitor.

Output: board body.

Destination: SQL database server.

Require: None.

Pre-condition: click the board name.

Post-condition: board body.

Figure (2.45) Boards marquee.



40.My style.

Function: My style.

Description: this function used to customize the appearance of each student account in the Portal according student tastes and desires.

Input: single click on my style link in the student main menu.

Source: student.

Output: preferred style appearance.

Destination: SQL database server.

Require: None.

Pre-condition: click on my style link in student menu.

Post-condition: target style.

Figure (2.46) My style.



41. Graduate projects.

Function: Graduate projects.

Description: displays all graduate projects and available document in a specific major by choosing major name, then all related projects appear in a table.

Input: major name.

Source: any PAP visitor.

Output: project body.

Destination: SQL database server.

Require: None.

Pre-condition: choose project name.

Post-condition: target project.

Figure (2.47) Graduate projects.



2.4 Validation Criteria:

To avoid potential user errors we have to meet all required validations to guide the user during his work with portal and obtain maximum benefit from the portal, these validation criteria listed below:

- Passwords must be at least six character length for all portal accounts.
- Passwords shouldn't contain any space.
- Passwords shouldn't contain any special characters (hyphen, semicolon, etc).
- Passwords may be the same for PAP users but username should be different.
- Only administrator have authentication to make changes on PAP database.
- Each student, administrator and instructor should have a unique ID session at the moment they log in to PAP site.
- Login names should not contain special characters (hyphen or semicolon).
- Login name shouldn't contain spaces.
- Each user input should be checked against all constraints.
- Voting allowed only one time for students and instructors for the same question.
- forgetting password; all fields must be filled.
- When issuing new boards; the following fields is required:
 - Board title
 - due to date.
 - Board type.
 - Board body.
 - Destination.
- When inserting new college the following fields are required:
 - College name.
 - College type.
 - College description.
- When insert mark or edit it; it should be between 40 t0 100.
- When deleting board, we have to select deletion criteria we will delete according with.



2.5 Information Description :

2.5.1 dataflow diagram :

Level 0: System Dataflow Diagram

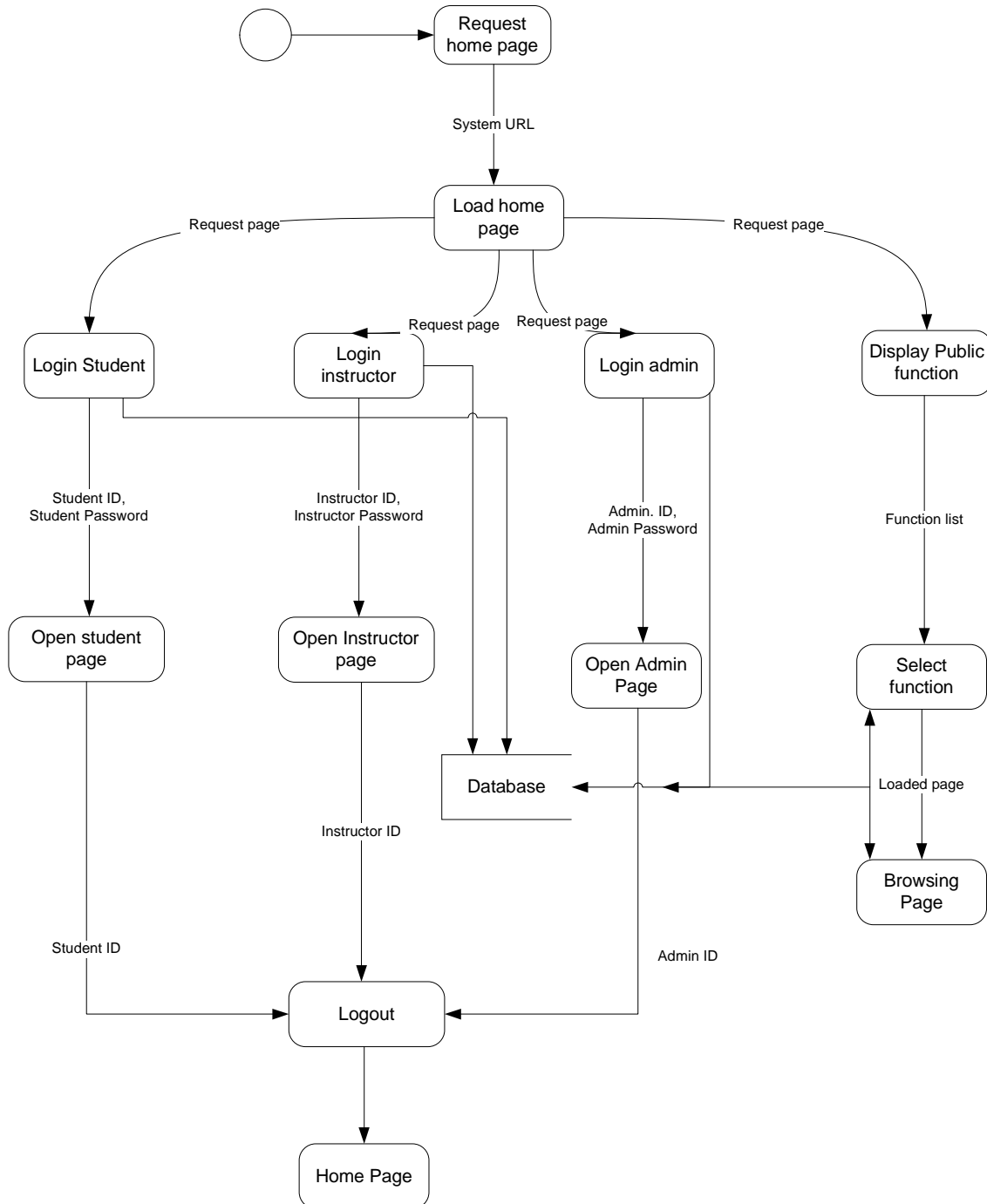


Figure (2.48) Dataflow Diagram level 0.



2.5.2 Data Dictionary:

Stored Procedures	
<u>Entity Name</u>	<u>Description</u>
Vote_yes	This procedure insert in vote_result table value 1 for yes and 0 for no and maybe.
Vote_no	This procedure insert in vote_result table value 1 for no and 0 for yes and maybe.
Vote_maybe	This procedure insert in vote_result table value 1 for maybe and 0 for no and yes.
usernamev	This procedure retrieve all students username in login_std table.
Updatevote	This procedure insert new question and its number in vote table
Updateset1	This procedure insert 1 for set_s in setting table.
Updateset0	This procedure insert 0 for set_s in setting table.
Updatepwd_std	This procedure update the password for students.
Updatepwd_admin	This procedure update the password for administrator.
Updatepwd_ins	This procedure update the password for instructor.
updatemark	This procedure update student mark.
updateduetodate	This procedure update the due to date of the board.
updatebrdt	This procedure update the board text body.
Updatebrdb	This procedure update the board attachment body.
Stdboards_str	This procedure retrieve student boards.



<u>Entity Name</u>	<u>Description</u>
Std_no	This procedure retrieve the student number for the current student name.
Semesterm_str	This procedure retrieve the course number and name and its mark that registered for student for current semester
Returnstdinmjr	This procedure retrieve all students in specific major.
returnnewbrdinstd	This procedure retrieve new students boards.
returnnewbrdindpt	This procedure retrieve new departments boards.
returnnewbrdincrs	This procedure retrieve new courses boards.
returnnewbrdincol	This procedure retrieve new colleges boards.
Tawjihi_branch	This procedure retrieve the number of tawjihi_branch for current tawjihi_branch name.
returnnewbrd	This procedure retrieve all new boards number according specific period.
returnmycrsbrd	This procedure retrieve all my new courses boards.
Returnmajors	This procedure return majors name.
returnlikemjr	This procedure return all majors info. That like the inserted major name.
returnlikeemp	This procedure return all employee info. That like the inserted employees name.
returnlikecrs	This procedure return all courses info. That like the inserted course name.
returnlikecol	This procedure return all colleges info. That like the inserted college name.
returnlikebrd	This procedure return all boards info.



	That like the inserted board name.
returnexpbrd	This procedure retrieve all expired boards.
returnduetodate	This procedure return the due to date for the board.
returndptbrdno	This procedure return the numbers of all department boards.
returncrsreg	This procedure return all courses registered for the student.
returncrsno	This procedure return the course number according its name.
returncolbrdno	This procedure return all colleges boards numbers according college name.
returnbrdno	This procedure return board number according board name.
returnbrdname	This procedure return all boards names.
Returnbrd_name	This procedure return the board name for the board number inserted.
Profile_str	This procedure retrieve all information in profile page for specific student.
Passedcrs_str	This procedure retrieve all finished hours and marks for student.
myaccount	This procedure return the student password.
mjrno	This procedure return the major number.
mainboards	Return all general (bulliten) boards.
liststdname	This procedure return students full name.
liststdno	This procedure retrieve all students numbers or usernames.
listtawname	This procedure return tawjihi branch name.
listmjrname	This procedure return all majors name.
listdptno	This procedure return all dept. numbers.



<u>Entity Name</u>	<u>Description</u>
listdptname	This procedure retrieve all departments names.
listcrsno	This procedure return all courses numbers.
listcrsname	This procedure retrieve all courses names.
listcolno	This procedure retrieve all colleges numbers.
listcolname	This procedure retrieve all colleges names.
isinstdbrd	This procedure check if the specific board in students boards or not, return 1 or 0.
isindptbrd	This procedure check if the specific board in departments boards or not, return 1 or 0
isincolbrd	This procedure check if the specific board in colleges boards or not, return 1 or 0
isincrsbrd	This procedure check if the specific board in courses boards or not, return 1 or 0
insertstdstyle	This procedure update student style by insert new values in students table for style attribute.
insertstdb	This procedure insert board number to std_brd table.
insertnewmjr	This procedure insert to majors table new major.
insertnewcol	This procedure insert to colleges table new college.
insertdptbrd	This procedure insert board number to dpt_brd table.
insertcolbrd	This procedure insert board number to col_brd table.



<u>Entity Name</u>	<u>Description</u>
insertcrsbrd	This procedure insert board number to crs_brd table.
insertbt	This procedure insert into boards table new board with text body.
insertbb	This procedure insert into boards table new board with attachment body.
Infoverify2	This procedure return all hours and its marks that student finished.
Infoverify1	This procedure return the student id, tawjihi average, and full name for verify.
Filterdptboards_str	This procedure filter the departments boards as selected departments and retrieve its info.
Filtercolboards_str	This procedure filter the colleges boards as selected college and retrieve its info.
Filtercrsboards_str	This procedure filter the courses boards as selected course and retrieve its info.
Fillsemno_str	Retrieve all semesters numbers .
Filldptname_str	This procedure retrieve all departments name.
Fillcrstype_str	This procedure retrieve all courses types.
dptno	This procedure retrieve department number according inserted name.
Dptboards_str	This procedure return all departments boards.
deletestdbrd	This procedure delete boards from std_brd table.
deletebrd	This procedure delete boards from boards table.
deletedptbrd	This procedure delete boards from dpt_brd table.



<u>Entity Name</u>	<u>Description</u>
deletecrsbrd	This procedure delete boards from crs_brd table.
deletecolbrd	This procedure delete boards from col_brd table.
Current_str	This procedure retrieve all courses that registered for student at current semester (current schedule).
crsno	This procedure return the course number according its name that inserted.
Crshours_str	This procedure return the course credit hours.
colno	This procedure return the college number according its name that inserted.
chkvoting	This procedure check if the student already voting before or not, return 1 or 0.
chktextnull	This procedure check if the board text body is null or not, return 1 or 0.
chkresultmjr	This procedure check if the major that inserted to search about it is exist in database or not, return 1 or 0.
chkresultemp	This procedure check if the employees that inserted to search about it is exist in database or not, return 1 or 0.
chkresultcrs	This procedure check if the course that inserted to search about it is exist in database or not, return 1 or 0.
chkresultdpt	This procedure check if the department that inserted to search about it is exist in database or not, return 1 or 0.
chkresultcol	This procedure check if the college that inserted to search about it is exist in database or not, return 1 or 0.



<u>Entity Name</u>	<u>Description</u>
chkresultbrd	This procedure check if the board that inserted to search about it is exist in database or not, return 1 or 0.
chkpic	Check if there is a picture stored for this student in database or not.
Chkdocnull	This procedure check if there is a document attached with the project or it is null.
chkattachmentnull	This procedure check if the board attachment body in boards table is null or not.
chkbrd	This procedure check if the board is exist in board table or not.
Balance_str	This procedure retrieve all financial balance information for the student.
alltrue	This procedure check if all data inserted to get the account online is valid or not.
academic	This procedure retrieve all academic status information from database for the student.
3true	This procedure return all finished hours information from passed table that student take it for calculate his accumulative average.
2true	This procedure return all information about finished hours from passed table that student take it in all semesters..
23true	This procedure check from the two previous procedures(2true,3true) together if its result is valid or not .



<u>Entity Name</u>	<u>Description</u>
1true	This procedure return all finished hours information from passed table that student take it in given year.
13true	This procedure check from the two previous procedures(1true,3true) together if its result is valid or not .

Table (2.1) Stored procedures.



Sessions (Functions)	
Entity Name	Description
Session("b")	Function that keep the student number(username) between pages.
Session("a")	Function that keep the student, and administrator name between pages.
Session("sem_no")	Function that keep the semester number between pages.
Session("ac_year")	Function that keep the academic year between pages.
Session("colno")	Function that keep the college number between pages.
Session("dptno")	Function that keep the department number between pages.
Session("insname")	Function that keep the instructor name between pages.
Session("pass_std")	Function that check if the student is login successful.
Session("pass_admin")	Function that check if the administrator is login successful.
Session("pass_ins")	Function that check if the instructor is login successful.
Session("brdreq")	Function that check which type of boards is required.
Session("selected")	Function that checked if value from drop down list is selected or not.

Table (2.2) Sessions.



Applications (Global Functions)	
Entity Name	Description
Application("count")	Counter that count the web site activity.
Application("count1")	Counter that count the page activity.
Application("marks_login1")	Check if the administrator or instructor login to marks page.
Application("marks_login2")	Check if more than one instructor is login to marks page, and count them.

Table (2.3) Applications.



2.5.3 System Interface description:

According to the user interface, the System will have an interface with three user categories; we have previously classified them as students, administrator, and instructors.

In this section we describe the system interface with each user category:

A. Student interface:

☒ Input:

Student input represented in making queries of information from database, this means that his access to database is restricted. Restricting the student access aimed at keeping and securing data integrity and consistency. He can obtain his information only by providing his username and password.

The student input represented in inserting his picture if he graduated and can vote on some question and can select his interface.

☒ Output:

the Portal provide student with all of his university information such as academic and financial information ...etc.

☒ GUI:

Our Portal support stronger student interface, this feature applied in my style function that enable each student to choose and customize the appearance of his PAP account.

Coloring system provided by Portal is comfortable and avoid eye stress.

B. Instructor interface:

☒ Input:

PAP system authorize instructor to insert part of student academic information and only for students he teach them for the current semester, such as insert student marks, and boards for target student.

**☒ Output:**

the new inserted boards by instructor appear in the scrolled marquee that visible only in the student account, and the instructor can prepare a report of his regular students, in addition to is new password output.

☒ GUI:

The screens that instructors use are to be smooth and meaningful.

*C. Administrator interface:***☒ Input:**

All administrator pages must be secure, the administrator will have the accessibility to the data in database tables, and this will be done through input tools to facilitate him to modify records easily and smoothly.

Validation techniques are needed to ensure that predefined input types and ranges are not violated.

☒ Output:

The output for all administrator transactions will be reflected on the system database tables thereby site data contents, so no specified output is to be produced for administrators except reports which should be available.

☒ GUI :

An administrator should be provided with a login page and others to enable him to modify site database records and site data contents.



2.5.4 Database Requirements :

Tables

1. Academic status:

- Academic number.
- warning.
- Dismiss.
- Delay
- Student level.
- Regular.
- Graduated.
- Registered hours.
- Passed hours.
- Community work hours.
- Semester number.
- Academic year.
- Study system.
- Student number.
- Student average.

2. Boards.

- Board number.
- Birth date.
- Due to date.
- Text body.
- Attachment body.
- Author.
- Subject.

3. College boards.

- College number.
- Board number.



4. Course boards.

- Course number.
- Board number.

5. Department boards.

- Department number.
- Board number.

6. Student boards.

- Student number.
- Board number.

7. Colleges.

- College number.
- College name.
- Description

8. Courses.

- Course number.
- Course name.
- Credit hours.
- Description.

9. course type.

- Course number.
- Course type.

10. Major courses.

- Course number.
- Major number.

11. departments.



- Department number.
- Department name.
- Description.
- College number.
- Community work hours.
- Academic number.

12. employee class.

- Employee number.
- Course number.
- Semester number.
- Class number.

13. employee.

- Employee number.
- Employee name.
- Department number.
- Student number.
- ID number.

14. financial status.

- Financial number.
- Assistant.
- Scholarship.
- Others.
- Credit.
- Debit.
- Student number.

15. login administrator.

- Username.
- Password.



16. login instructor.

- Employee number.
- Password.

17. login student.

- Student number.
- password.

18. majors.

- Majors number.
- Major name.
- Hours.
- Description.
- Department number.
- Outline.

19. offered courses.

- Course number.
- Class number.
- Time.
- Room number.
- Semester number.
- Academic year.

20. passed course.

- Student number.
- Course number.
- Semester number.
- Academic year.
- Class number.
- Mark.
- College number.



21. setting.

- Setting number.
- Setting status.
- Setting date.

22. students.

- Student number.
- First name.
- Student name.
- Third name.
- Last name.
- Birth year.
- Birth month.
- Birth day.
- Birth place.
- Nationality.
- Social status.
- Student ID.
- Tawjihi average.
- Gender.
- Address.
- Telephone number.
- Major number.
- Academic number.
- Tawjihi number.
- Picture.

23. study system.

- Student number.
- Study name.

24. tawjihy branch.

- Tawjihi number.



- Tajikhy name.

25. vote.

- Vote number.
- Vote body

26. vote result.

- Vote number.
- Voting yes.
- Voting no.
- V maybe.
- Who.

27. Student style:

- student number.
- Main header.
- Name header.
- Title header.
- Box.
- Center box.
- Box header.
- Center-box header.
- Logout.

28. Projects :

- Student number
- Project number
- Project name
- Project body
- Employee number



Views

1. academic_s (academic status):

- std_fname (student first name).
- std_sname (student second name).
- std_tname (student third name).
- std_lname (student last name).
- Birth_p (birth place).
- Ta_name (Tawjihy name).
- Tawjehy_avg (tawjihi average).
- Warning (warned or not).

2. balance (financial balance):

- assistant (assistance value).
- Scholarship(scholarship value).
- Others.
- Credit.
- Debit.
- Stdno (student number).
- Semno (semester number).
- Ac_year (academic year).

3. basic_info (basic information):

- std_no (student number).
- std_fname (student first name).
- std_sname (student second name).
- std_tname (student third name).
- std_lname (student last name).
- Tel_no (telephone number).
- Address.

4. board_view (board view):

- date (issue date).



- Author (author name).
- Title.
- Board_no (board number).

5. calcavg (calculate average):

- std_no (student number).
- Mark (mark value).
- Crs_no (course number).
- Co_no (college number).
- Crd_hours (credit hours number).
- Std_fname (student first name).
- Std_sname (student last name).
- Std_tname (student third name).

6. col_brd (college board):

- board_date (board issue date).
- Author (author name).
- Title (title).
- College_no (college number).

7. col_info (college information):

- college (college name).
- College_no (college number).

8. colleges_boards (colleges boards):

- author (author name).
- Subject (subject title).
- Col_no (college number).
- Brd_no (board number).
- Coll_name (college name).
- Brd_date (board date).
- Bodyt (text body).
- Bodyb (attachment body).

**9. courses_boards (courses boards):**

- author (author name).
- Subject (subject title).
- Crs_no (course number).
- Brd_no (board number).
- Brd_date (board date).
- Crs_name (course name).
- Bodyt (text body).
- Bodyb (attachment body).

10. crs_boards (courses boards):

- Crs_name (course name).
- Brd_date (board date).
- Auther (author name).
- Subject (subject title).
- Std_no (student number).
- Crs_no (course number).
- Brd_no (board number).

11. crs_info (course information):

- Course_no (course number).
- Course (course name).
- Credit_hours (credit hours number).

12. crs_reg (courses registered):

- Crs_name (course name).
- Std_no (student number).
- Crs_no (course number).

13. current_schedule (current schedule):

- Time (time).
- Room_no (room number).



- Crs_name (course name).
- Crs_no (course number).
- Sem_no (semester number).
- Std_no (student number).
- Class_no (class number).
- Ac_year (academic year).

14. departments_boards (department boards):

- Author (author name).
- Subject (subject title).
- Dpt_no (department number).
- Dpt_name (department number).
- Brd_no (board number).
- Brd_date (board issue date).
- Bodyt (text body).
- Bodyb(attachment body).

15. dpt_brd (department boards):

- Board_date (board issue date).
- Author (author name).
- Title (title).
- Dpt_no (department number).

16. emp_account (employee account):

- Emp_no (employee number).
- Emp_name (employee name).
- Dpt_name (department name).
- St_name (study name).
- Pwd (password).
- Id_no (ID number).

17. emp_info (employee information):

- Employees (employee name).



- Science degree (scientific degree).
- Department (department name).
- Employees_no (employee number).

18. emp_view (employee view):

- Emp_no (employee number).
- Crs_no (course number).
- Class_no (class number).
- Std_no (student number).

19. login_view (login view):

- std_fname (student first name).
- Std_sname (student second name).
- Std_tname (student third name).
- Std_lname (student last name).
- Pwd (password).
- Std_no (student number).
- Dpt_no (department number).
- Col_no (college number).

20. loginins (log in as instructor):

- emp_no (employee number).
- Pwd (password).
- Emp_name (employee name).

21. main_boards (main boards):

- due_to (due to date).
- Subject (subject name).
- Brd_no (board number).
- Brd_date (board date).
- Bodyt (text body).
- Bodyb (attachment body).
- Author (author).

**22. mark_hours (marks and hours to calculate average):**

- crd_hours (credit hours).
- Std_no (student number).
- Mark (mark).

23. mjr_info (major information):

- major (major name).
- Credit_hours (credit hours value).
- Outline (year of major plan).
- Department (department name).
- Major_no (major number).

24. my_crsboards (course boards):

- Crs_no (course number).
- Brd_no (board number).
- Bodyt (text body).
- Bodyb (attachment body).
- Author (author name).
- Subject (subject title).
- Brd_date (board initial date).
- Std_no (student number).

25. passed (passed courses):

- Crs_no (course number).
- Std_no (student number).
- Co_t (course type).
- Co_no (course number).
- Crs_name (course name).
- Crd_hours (course credit hours).
- Mark (course mark).
- Sem_no (semester number).

**26. profile (student profile):**

- std_fname (student first name).
- Std_sname (student second name).
- Std_tname (student third name).
- Std_lname (student last name).
- Address (address).
- Telno (telephone number).
- Mjr_name (major name).
- Outline (major plan year).

27. sem_hours (number of registered semester hours):

- Std_no (student number).
- Hours_reg (registered hours).
- Hours_p (passed hours).

28. semester_marks (semester marks):

- Crs_name (course name).
- Crd_hours (credit hours).
- Crs_no (course number).
- Std_no (student number).
- Mark (course mark).
- Ac_year (academic year).
- Sem_no (semester number).
- Co_no (college number).

29. std_brd (student boards):

- Board_date (initial board date).
- Author (author name).
- Title (board title).
- Std_no (student number).

30. stdpaper (student confirmation sheet):

- Std_fname (student first name).



- Std_sname (student second name).
- Std_tname (student third name).
- Std_lname (student last name).
- St_name (study name).
- Mjr_name (major name).
- Std_no (student number).

31. student_boards (student boards):

- Author (author name).
- Subject (subject title).
- Std_no (student number).
- Std_sname (student second name).
- Std_fname (student first name).
- Std_tname (student third name).
- Std_lname (student last name).
- Brd_no (board number).



2.5.5 System Context Model:

This diagram explain relationship in abstract between the Portal system and other environment systems without details.

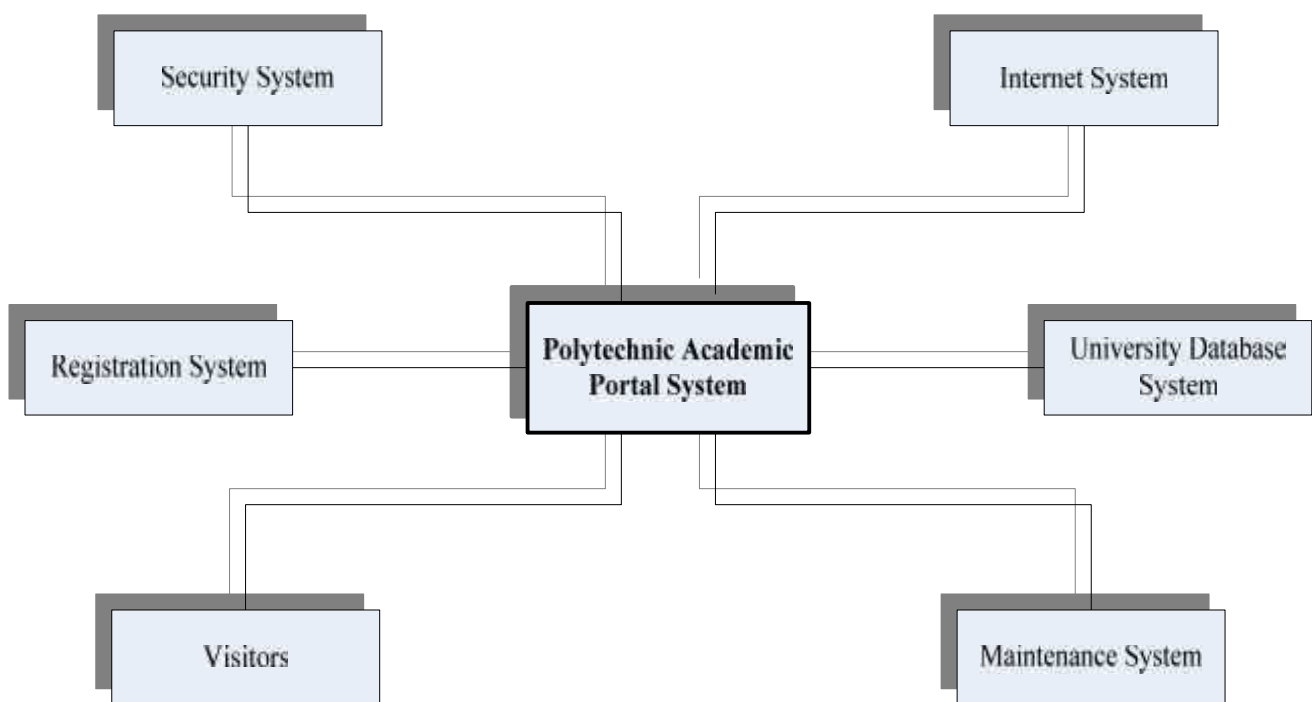


Figure (2.49) System Context Model



3.1 Introduction

This chapter describes the system design that have objects design for each module and the topics in this chapter is: (description, interface, flowchart, constraints or validation, and user interface design) , I/O design for designing all input /output screens , and database design show database model and all table constraints, test plan , and program languages and coding .

3.2 Functional design

functional design for each module should described in the software system, accordingly with the description of the interface, the constraints, and the user interface design in which we use means of diagramming to help us understand it:

These functions design divided to:

1. Student Functions design
2. Administrator Functions design
3. Instructor Functions design
4. Public Function design



A . Student Functions Design :

1- login student:

- a. Description : This function that gives the students authentication or enabling them to login to them accounts .
- b. Interface:
 - ◆ Input : student ID (Student number), student Password
 - ◆ Output : Student home page if your id and password is true , error message if id or password is invalid.
- c. Constraints:
 - ◆ Only students and administrator are authenticated to login to this page.
 - ◆ Password must be checked. and it considered as a character case
 - ◆ Password must be at least 6 characters.
 - ◆ Username must be numbers only (student academic number).



d. Flowchart:

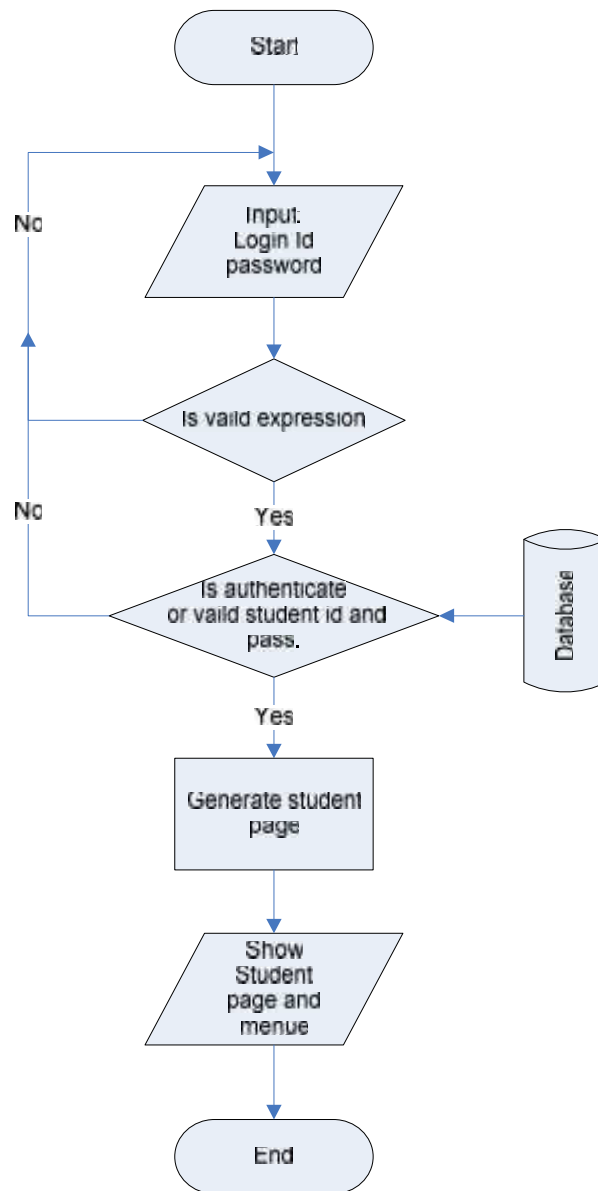
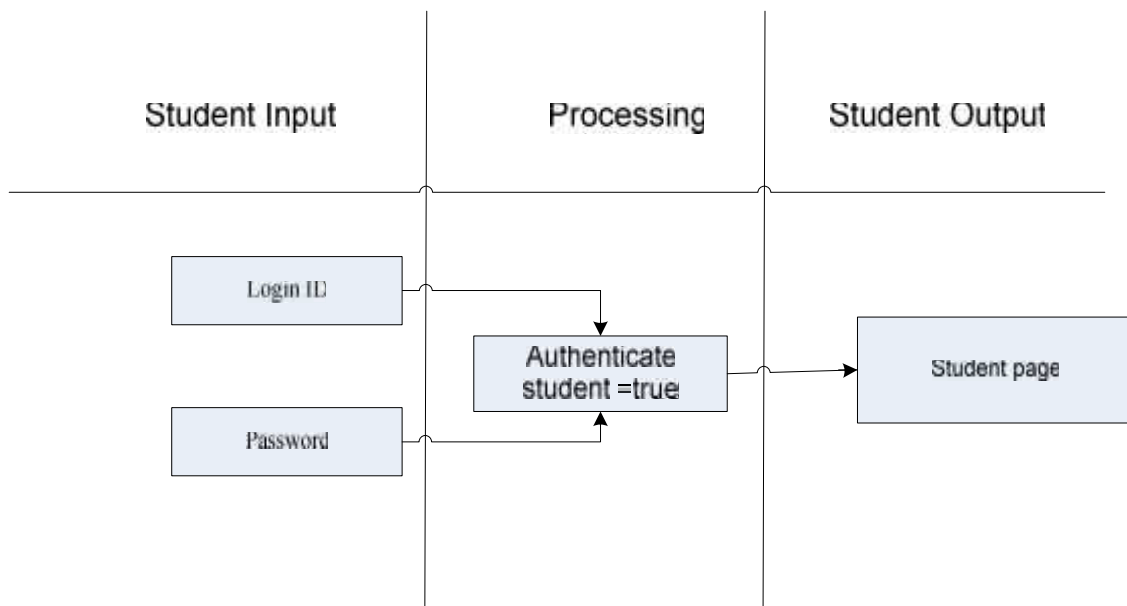


Figure (3.1) Student login Operation.



>> User interface design



2- *Forgot my Password:*

a. Description :This function is important and allow the student to retrieve his password if he forgot it, and the student can get his account after the system checked his username and then asked he some questions that must be answered correctly.

b. Interface:

- ◆ Input: student username, to ensure that he has an account on PAP.
- ◆ Input : if the previous username is valid then he will be asked about : student id card Number , Tawjihi Average, Accumulative Average, Birth Date



- ◆ Output : if he answered all questions correctly the system will send the student account to his e-mail, but in our system he will show his account (username, password, group) because there is no connection yet with university mail server, else s\error message will be appear.

c. Constraints:

- ◆ Student id card number must be numbers and 9 digit.
- ◆ Accumulative average and tawjihi average must be numbers.
- ◆ Birth date format (month, day, year).



d. Flowchart:

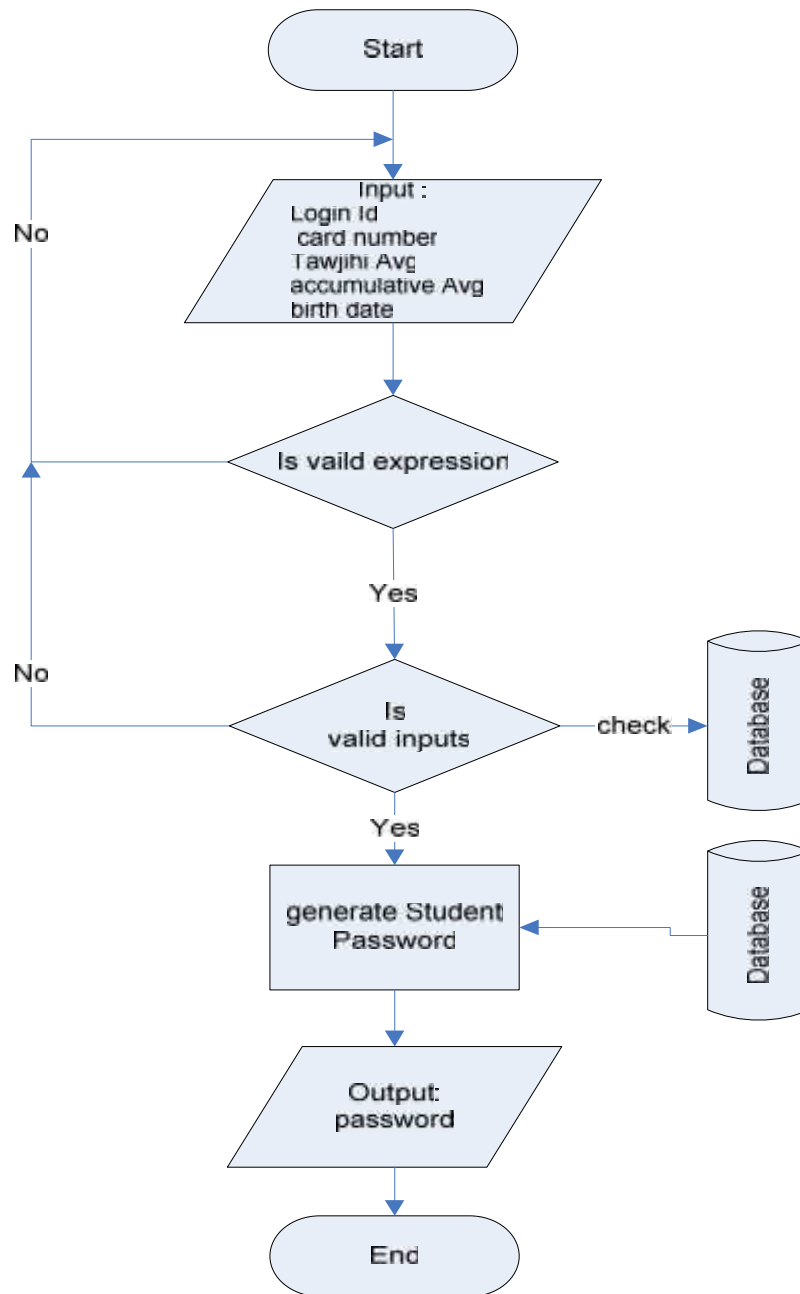
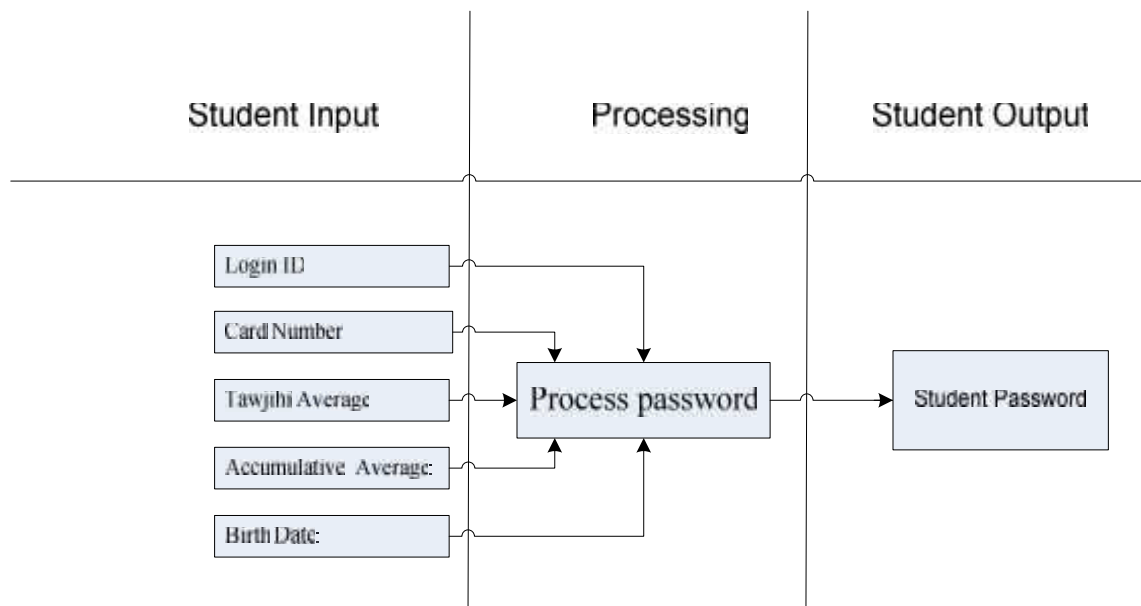


Figure (3.2) Forgot My password Operation.



>> User interface design



3- Student Profile :

- a. Description :This function show the student profile in the university that have name, address of student ,and academic information, and there is an area that allow to student to insert him picture if he want, and if he was graduated he will be able to insert his graduate project document to portal. This function dependent on the student id (username), the session take student Id (username) and show student profile according it .

b. Interface:

- ◆ Input : Session(student Id)
- ◆ Output : Full Student Name, Address, Tel. No., Academic No, Scientific Degree , Major, Outline.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information displayed is read only and the student can't edit it, just his picture and document can edit them.

d. Flowchart:

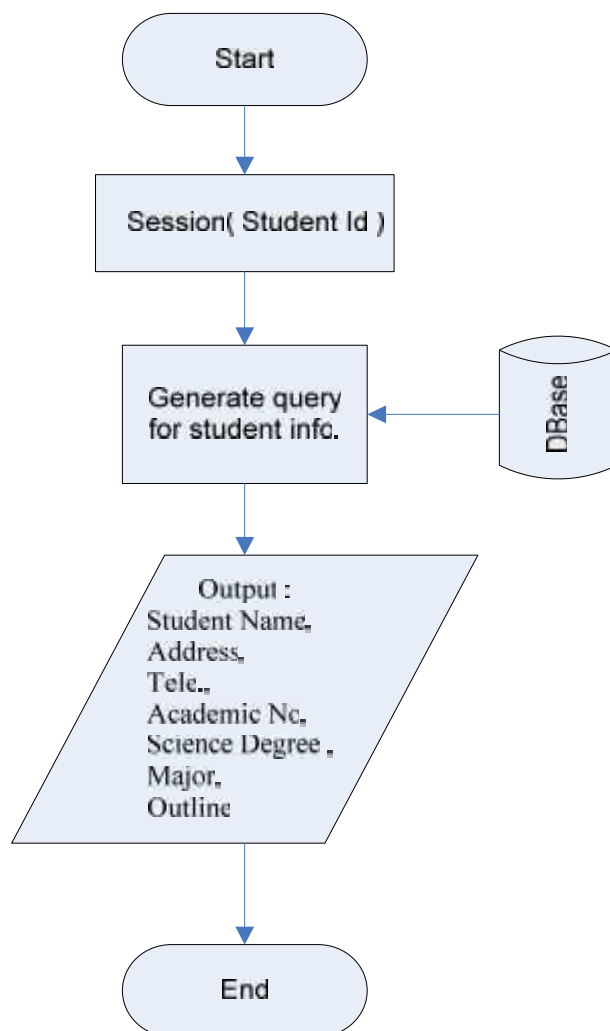
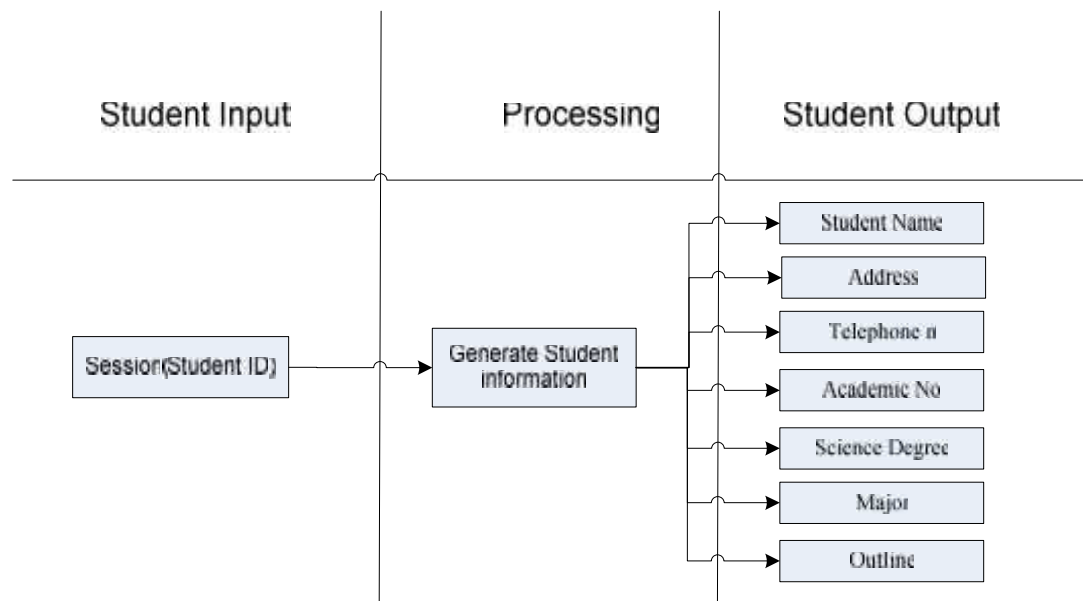


Figure (3.3) Student profile Operation.



>> User interface design



4- Current Schedule :

- a. Description :This function show the student Current Schedule in the university that have Course No., Semester, Course Name, Instructor, Section, Time, Room No. This function dependent on the student id (username), the session take student Id and show current schedule according it .

b. Interface:

- ◆ Input : Session(student Id)
- ◆ Output : Course No., Semester, Course Name, Instructor, Section, Time, Room No.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ Student ser can't change any information (all information read only).

d. Flowchart:

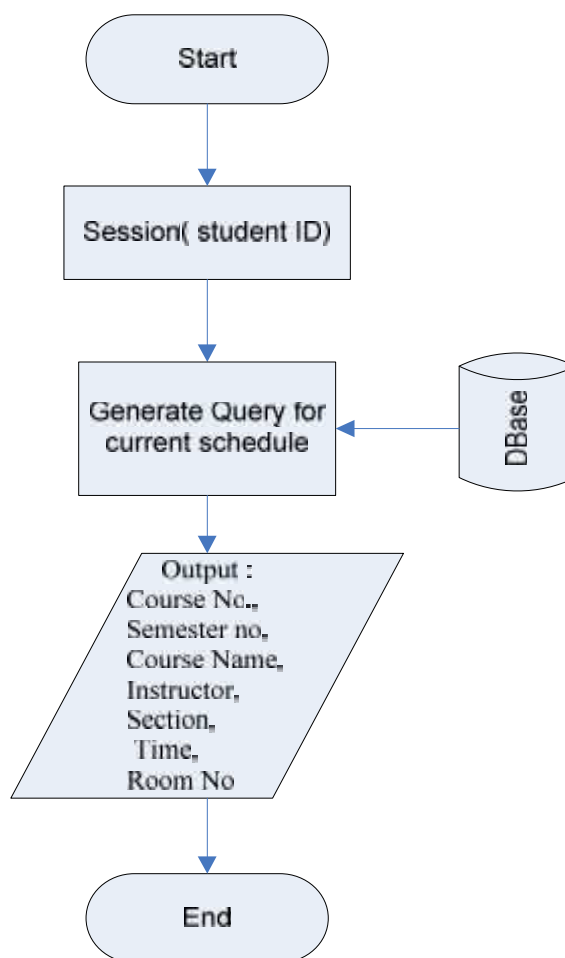
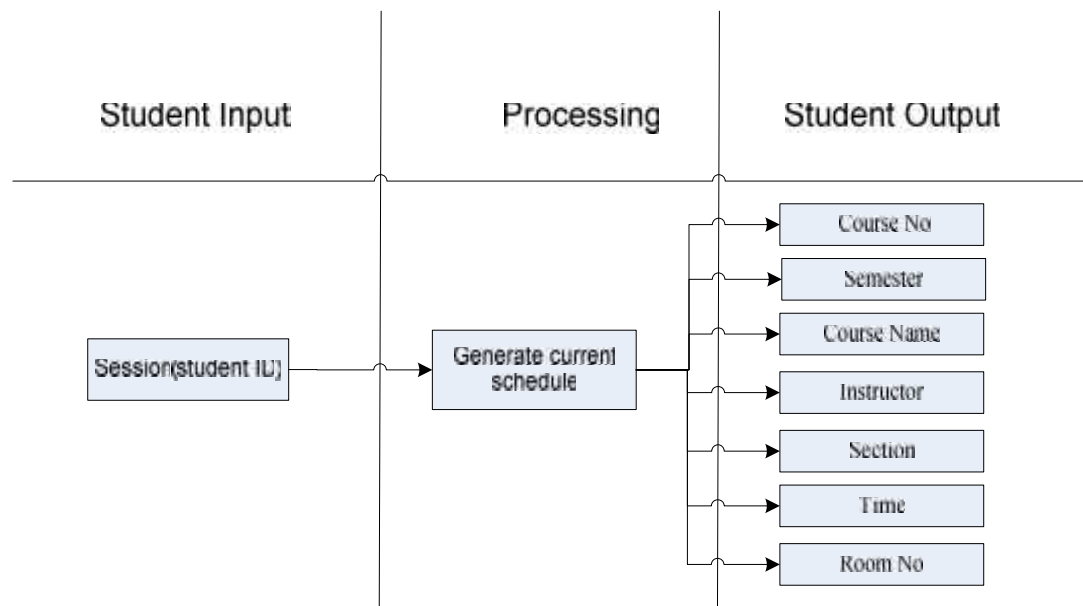


Figure (3.4) Student Current Schedule Operation.



>> User interface design



5- *Financial Balance:*

a. Description :This function show the financial balance for student in the university that have Financial Aids (Assistance, Scholarship, Other) , Balance(Credit, Debit), Semester Balance and Total Balance.

b. Interface:

- ◆ Input : student Id in session ,semester number in session
- ◆ Output : Financial Aids (Assistance, Scholarship, Other), Balance(Credit, Debit), Semester Balance and Total Balance



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information here is read only.

d. Flowchart:

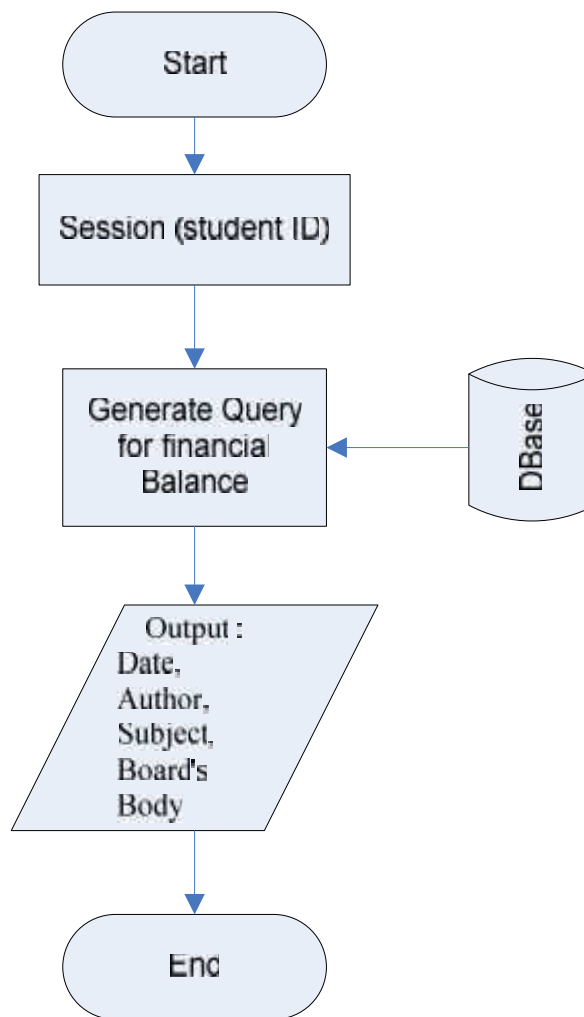
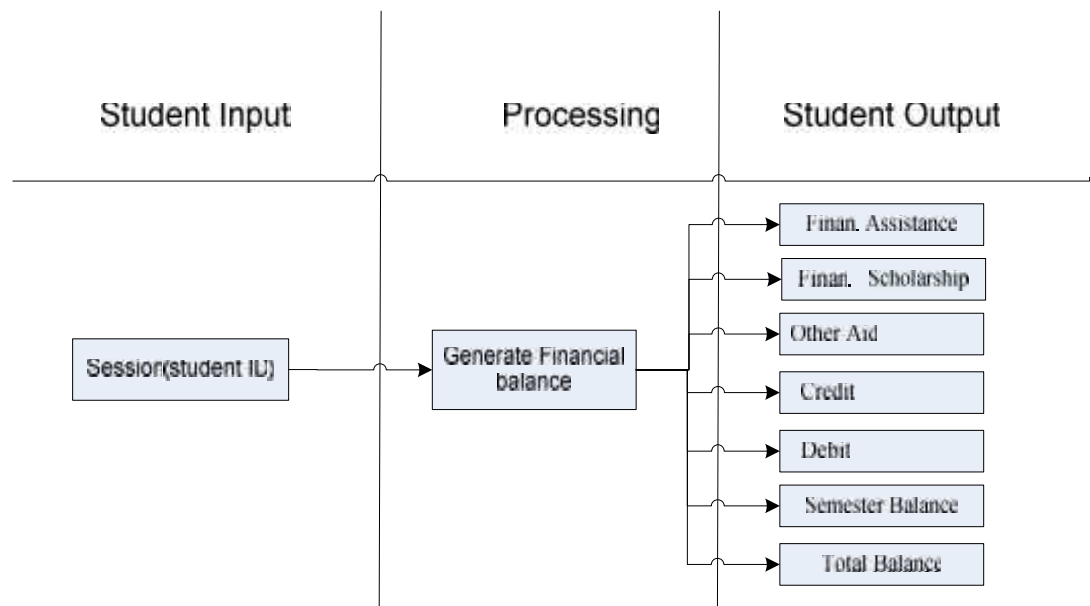


Figure (3.5) Student Financial Balance Operation.



>> User interface design



6- Academic Status:

- a. Description :This function show the academic status for each student in the university that have major degree, tawjihi branch and average, community work hours (passed and remains) department name and college name, average accumulate and major, academic alert,

b. Interface:

- ◆ Input : Session(student ID)
- ◆ Output : major degree, tawjihi branch, tawjihi average, community work hours (passed and remains), department name and college name, average accumulate and average major, academic alert, Dismiss, Delay, Level, Regular, Graduated.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information here is read only.

d. Flowchart:

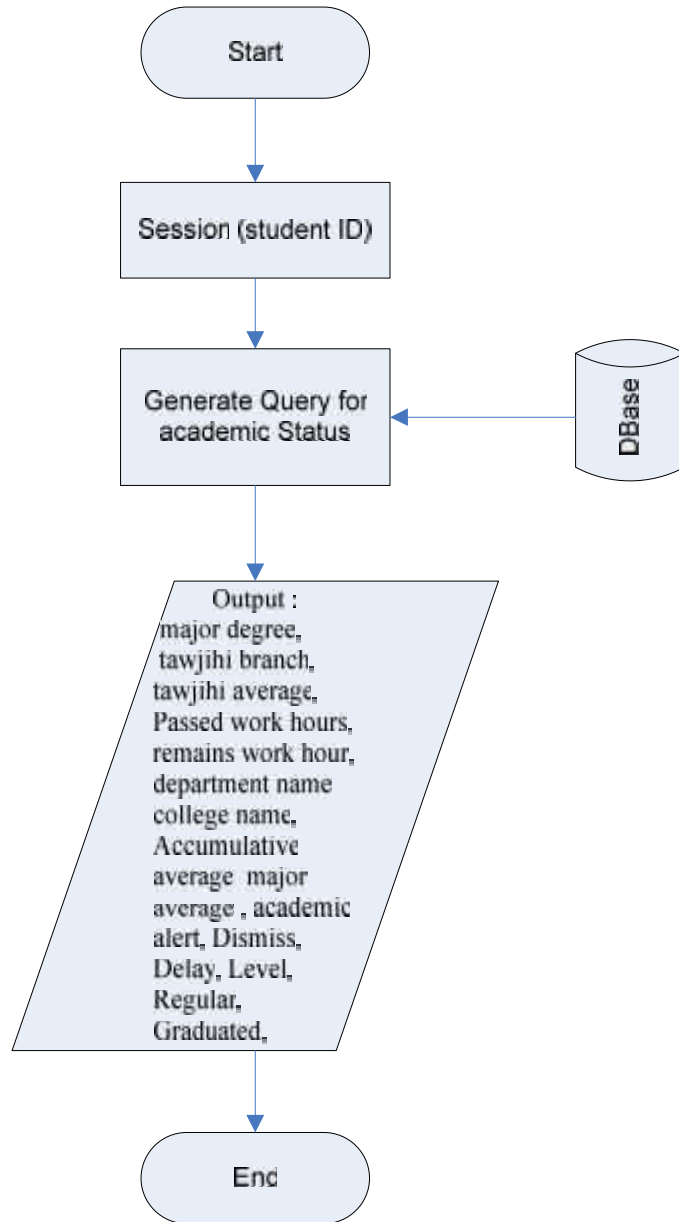
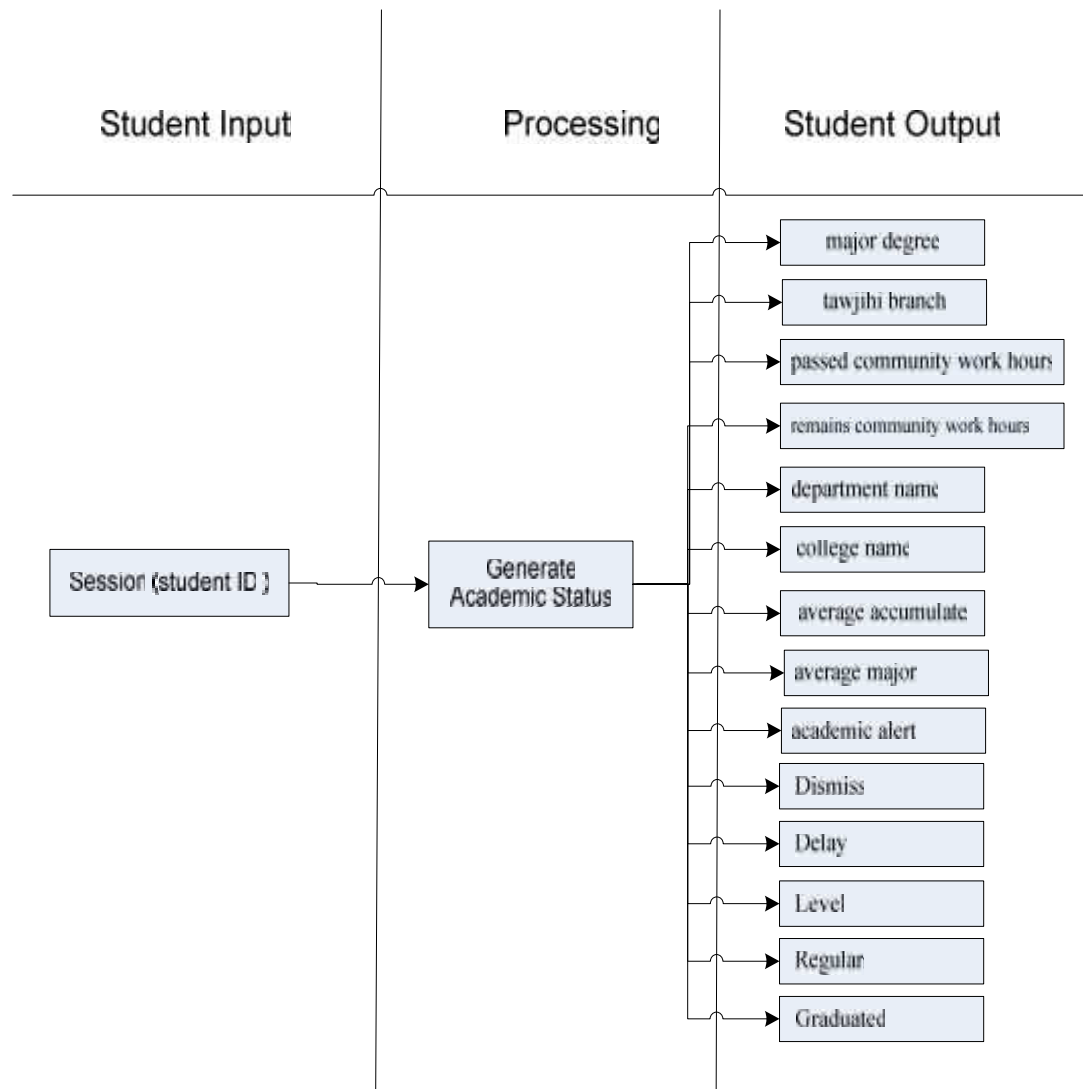


Figure (3.6) Student academic status



>> User interface design



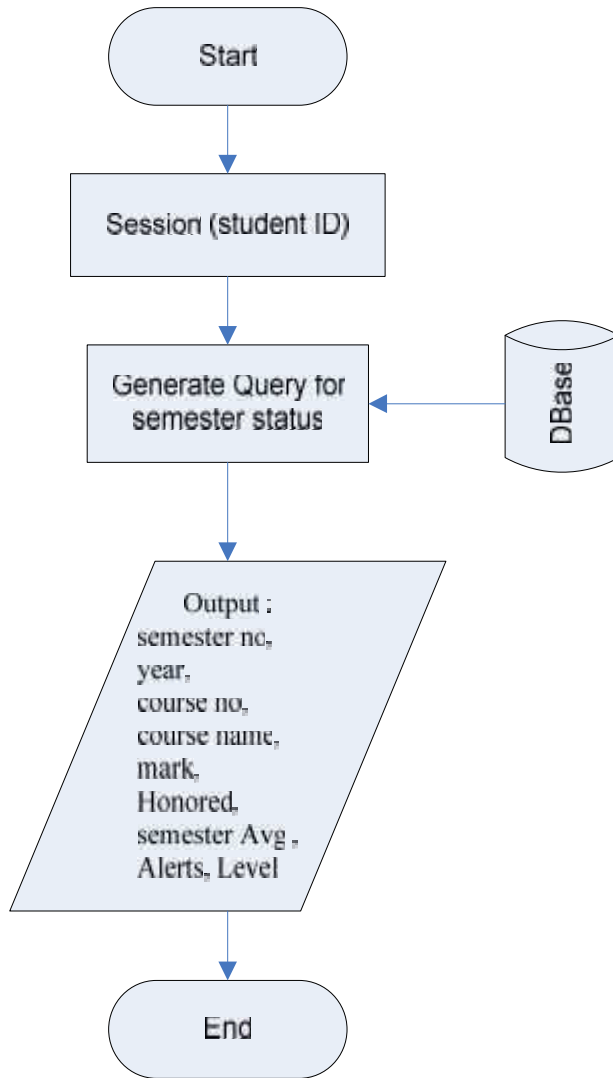


7- Semester Status:

- a. Description :This function show the semester status for registered student, and this page show the finished semester marks , the semester average, hours registered and hours passed, and if the student have an honor or not according his average .
- b. Interface:
 - ◆ Input : student Id in session, semester number in session.
 - ◆ Output : semester number, year, course no, course name, mark, Honored, Semester average.
- c. Constraints:
 - ◆ Authenticated student can only show this page.
 - ◆ All information is read only.



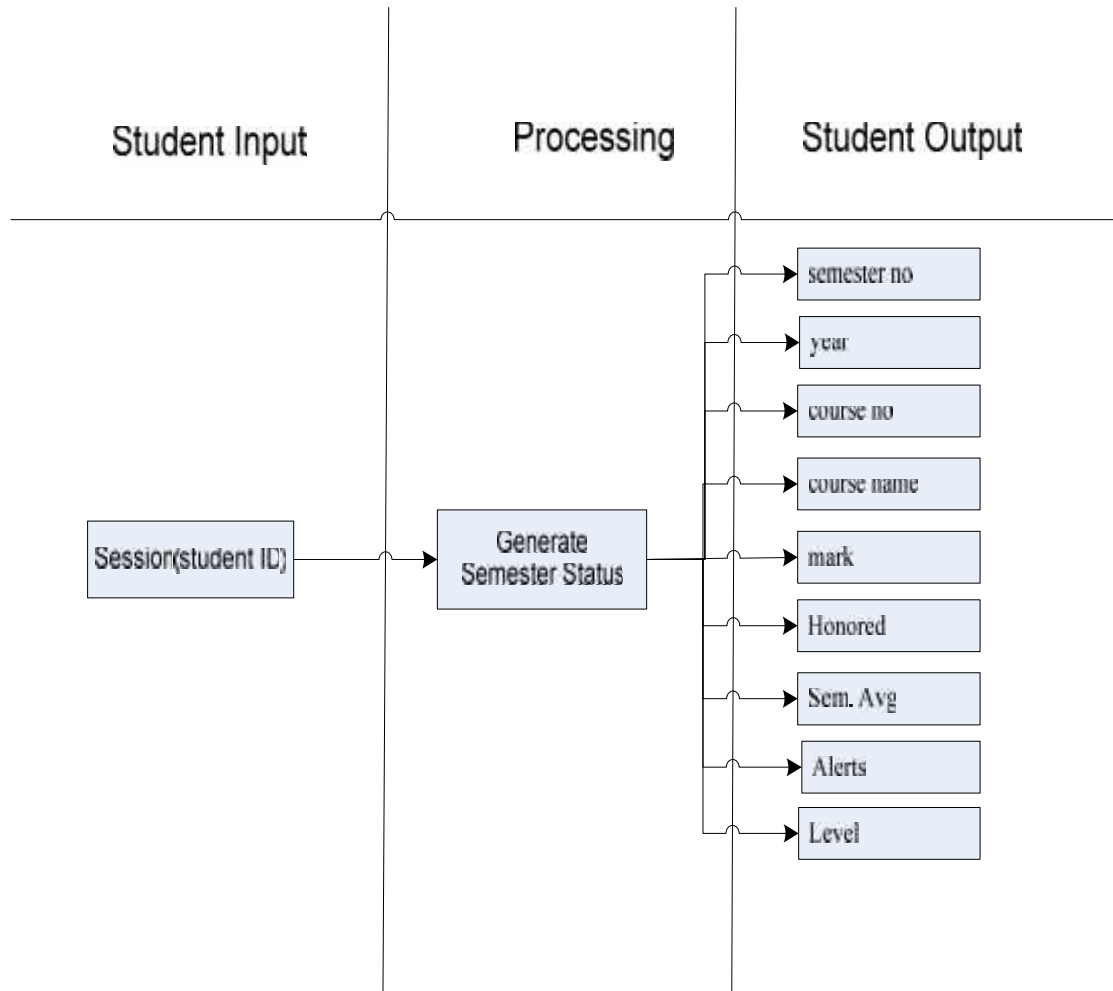
d. Flowchart:



Figure(3.7) student semester status operation.



>> User interface design





8- *Passed Courses:*

a. **Description :** This function shows the finished hours and marks for the student, the student can filter this information according to semester, year, course type, and the student can show his accumulative average, number of passed hours, and number of remains, and he can show if he completes the graduate requirements or not..

b. **Interface:**

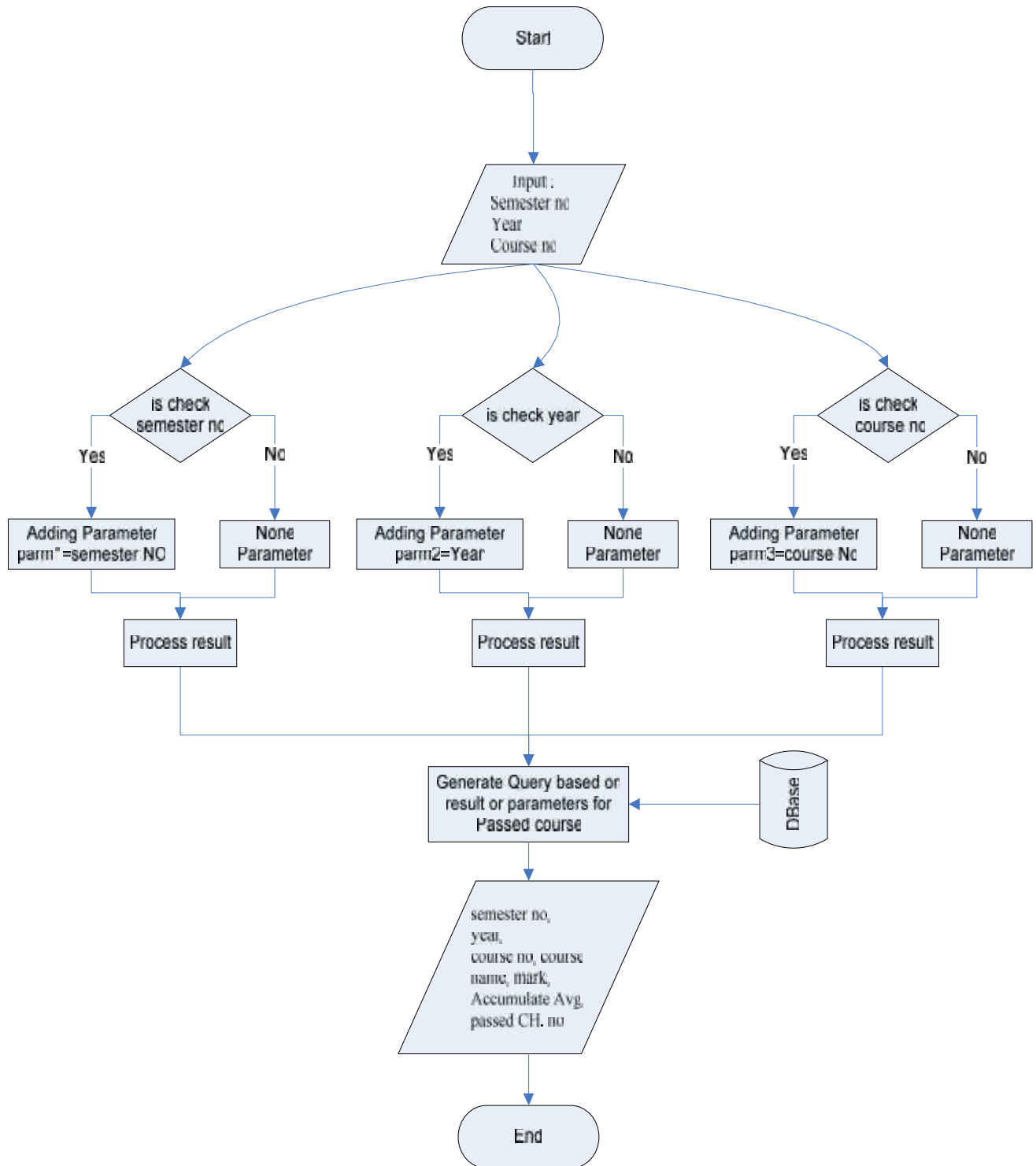
- ◆ **Input :** Session(student Id) , semester number in session, year in session.
- ◆ **Output :** semester number, year, course no, course name, course type, course mark, credit hours passed and remains number, accumulate avg..

c. **Constraints:**

- ◆ Authenticated student can only show this page..
- ◆ All information is read only.



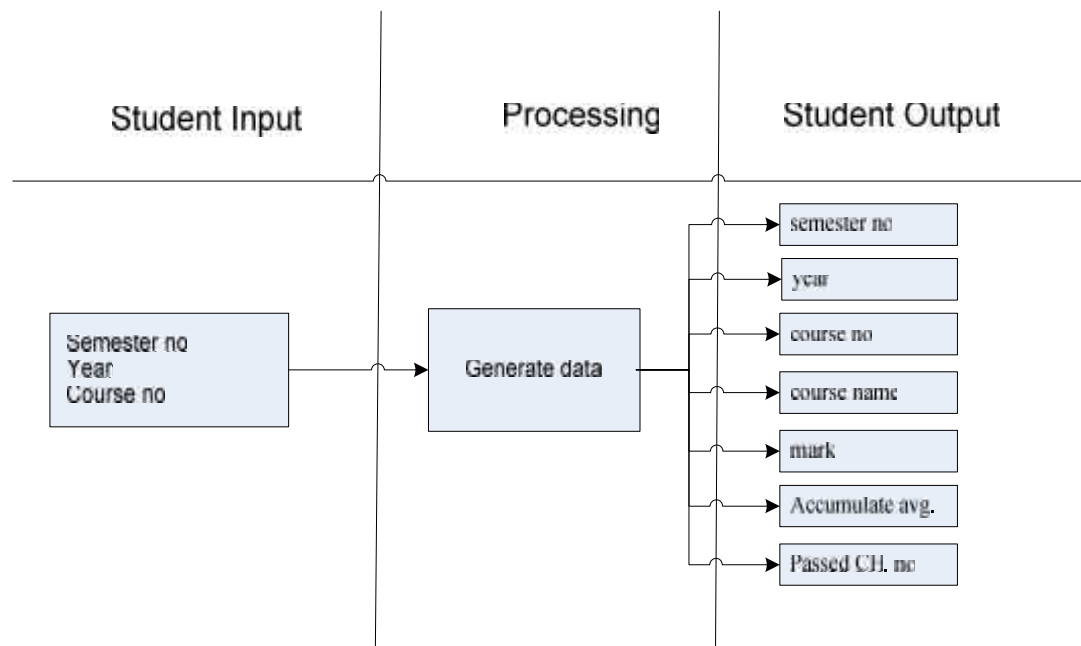
d. Flowchart:



Figure(3.8) passed courses operation.



>> User interface design



9- My Boards:

a. Description :This function show the student boards only for current student

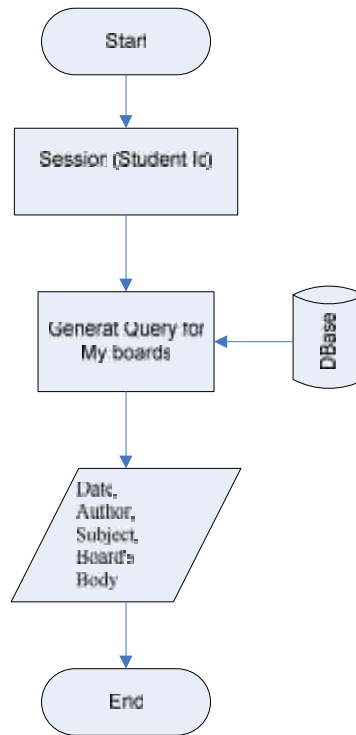
b. Interface:

- ◆ Input : Session (student ID)
- ◆ Output : Date, Author, Subject, Board's Body

c. Constraints:

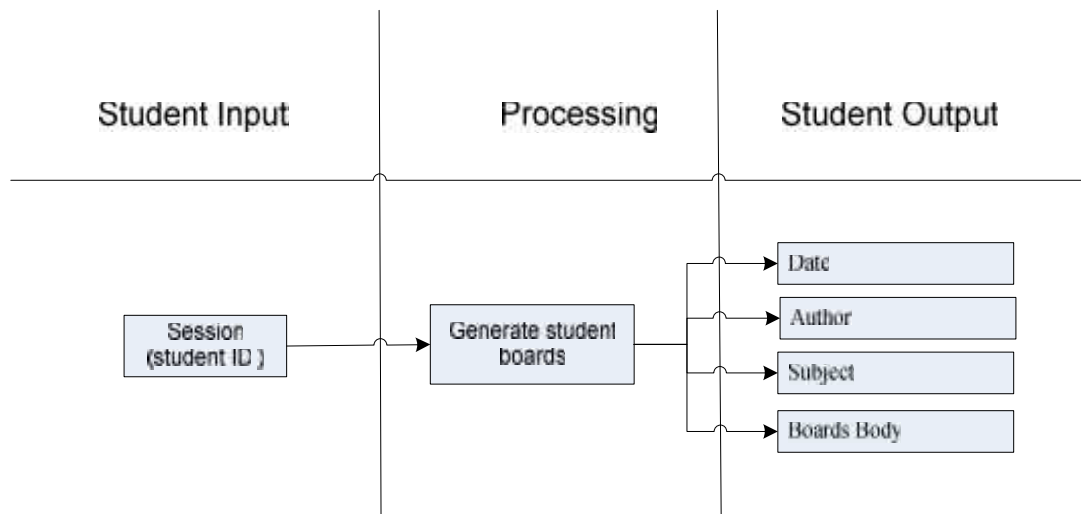
- ◆ Authenticated student only can show this page.
- ◆ All information is read only.

d. Flowchart:



Figure(3.9) My boards operation.

>> User interface design



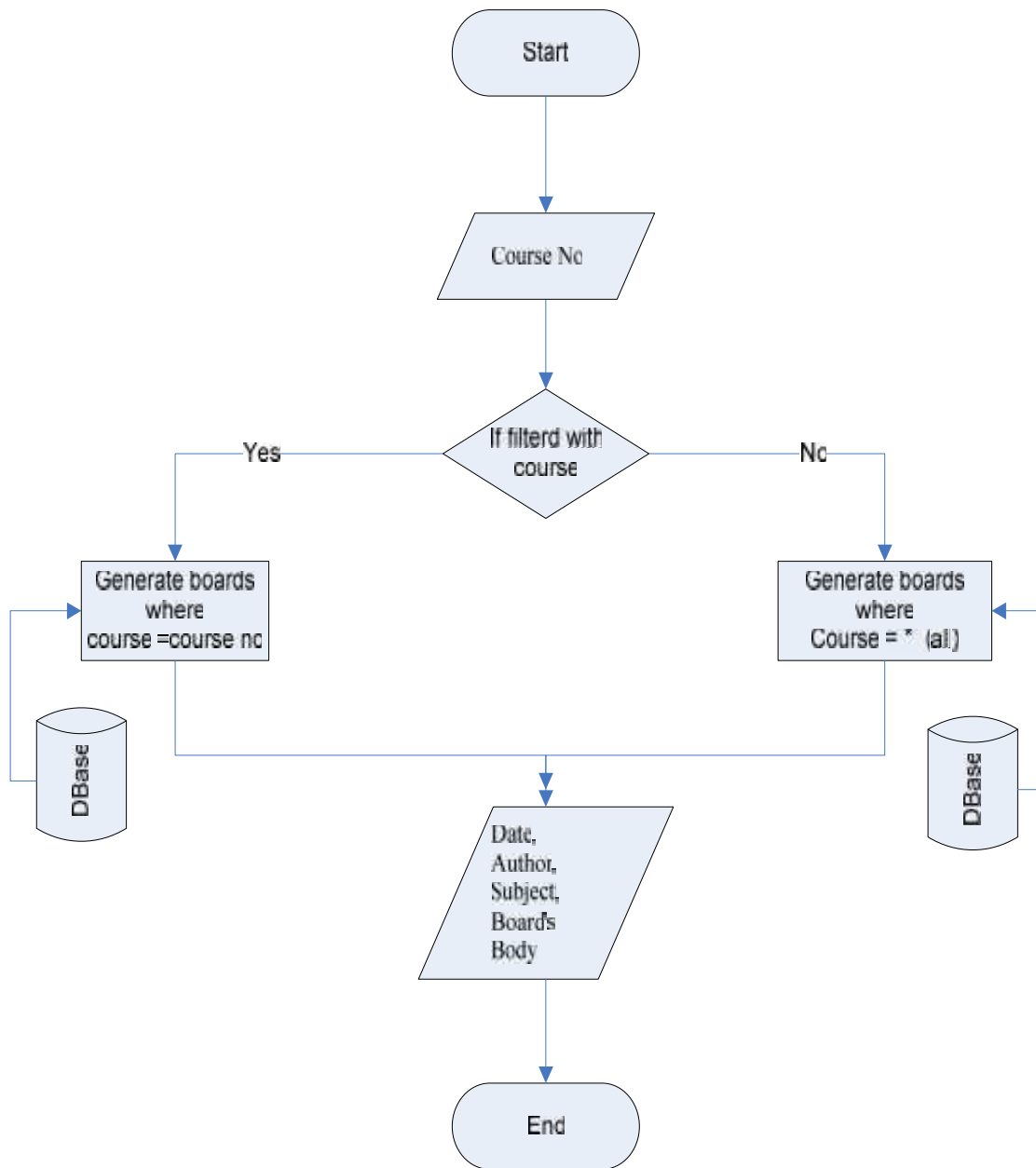


10- *Course Boards:*

- a. **Description :** This function show the courses boards for the current student courses .
- b. **Interface:**
 - ◆ **Input :** session(student Id), semester number in session, year in session.
 - ◆ **Output :** Date, Author, Subject, Board's Body
- c. **Constraints:**
 - ◆ **Authenticated student only can show this page.**
 - ◆ **All information is read only.**



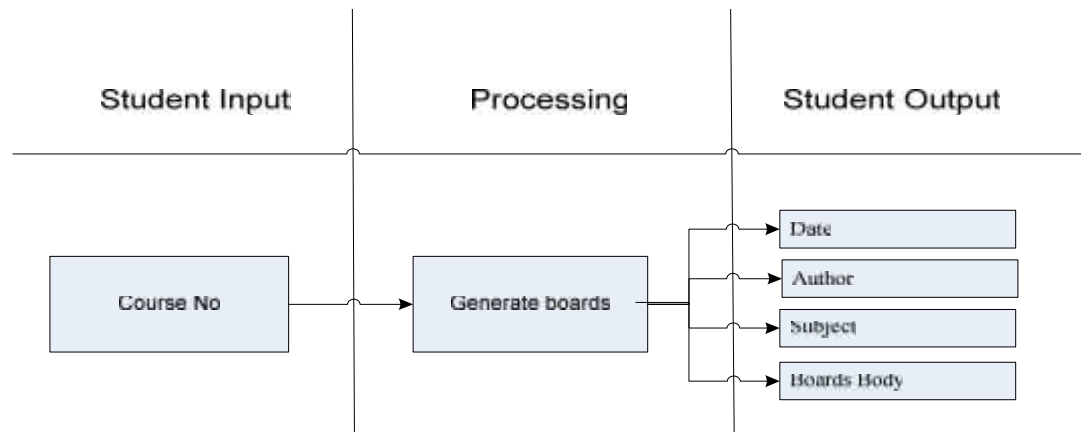
d. Flowchart:



Figure(3.10) Course boards operation.



>> User interface design

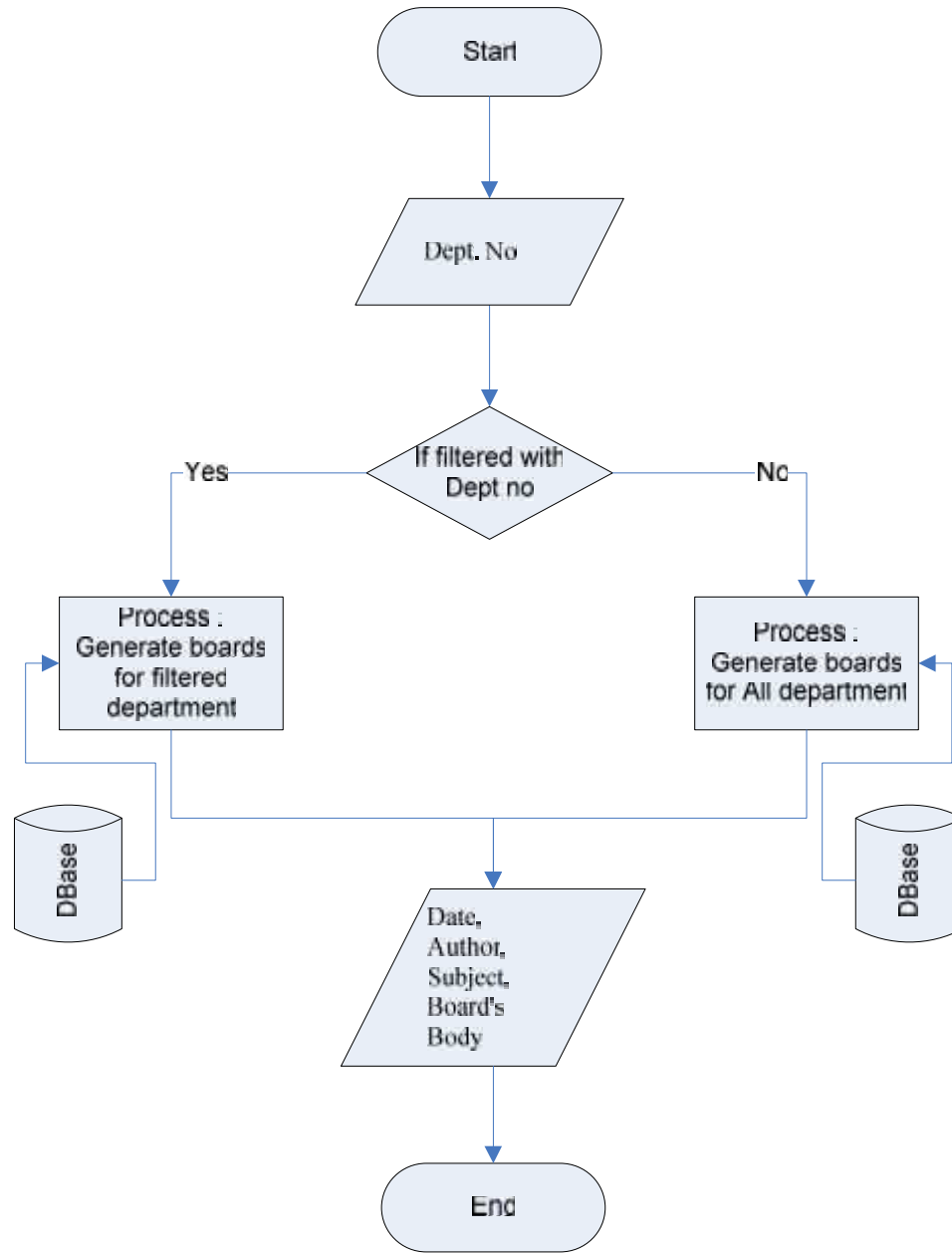


11- Department Boards:

- a. Description :This function show the department boards for the student in a specific department.
- b. Interface:
 - ◆ Input : session (student Id) , department number in session.
 - ◆ Output : Date, Author, Subject, Board's Body
- c. Constraints:
 - ◆ Authenticated student only can show this page.
 - ◆ All information is read only.



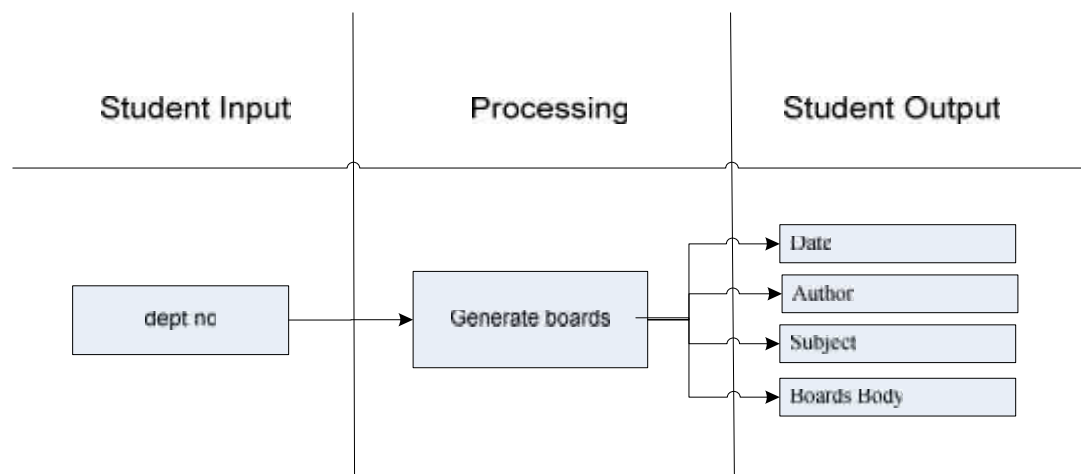
d. Flowchart:



Figure(3.11) Department boards operation.



>> User interface design



12- College Boards:

a. description :

This function show the college boards for a student in specific college.

b. interface:

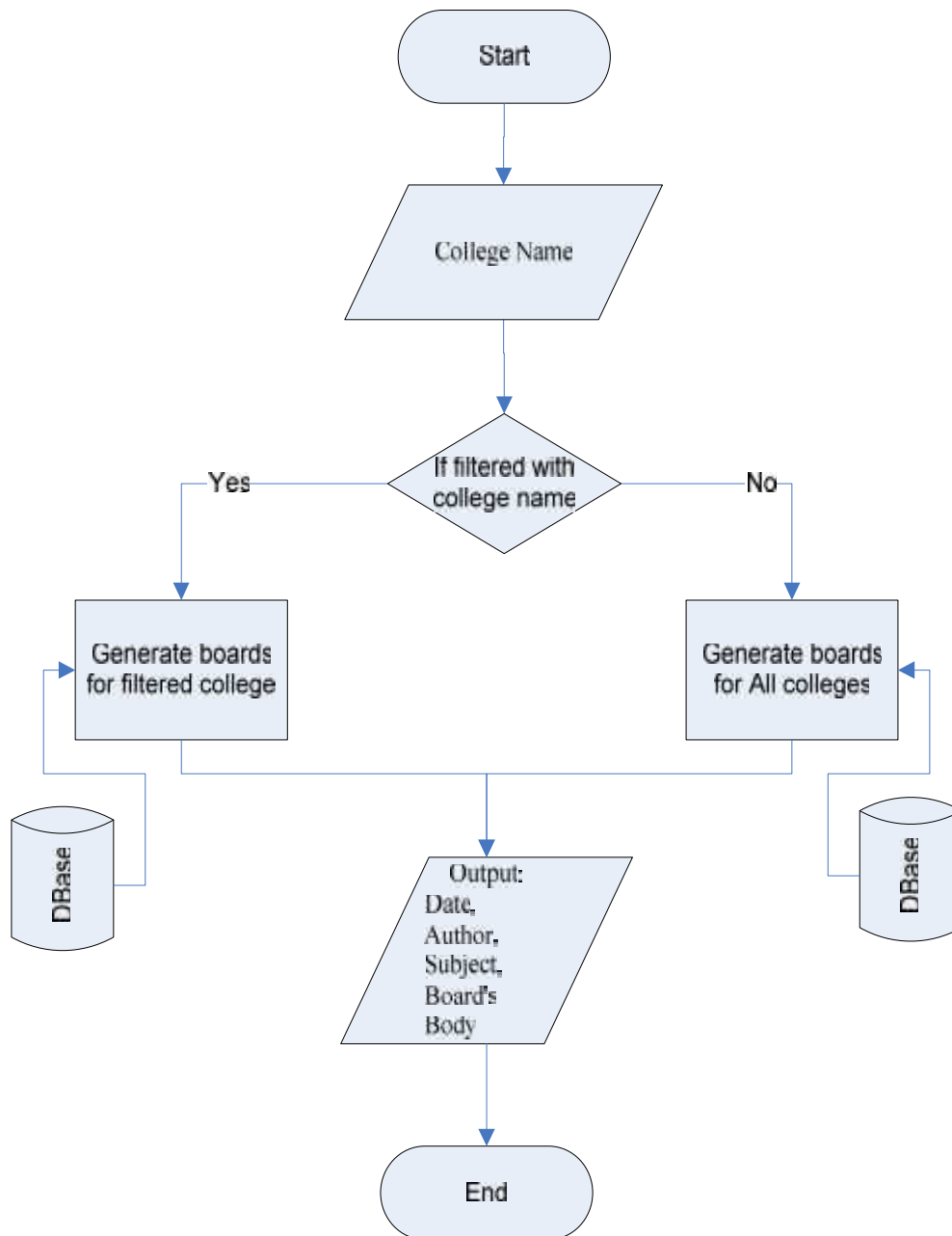
- ◆ Input : session(student Id) , college number in session
- ◆ Output : Date, Author, Subject, Board's Body

c. Constraints:

- ◆ Authenticated student only can show this page.
- ◆ All information is read only.



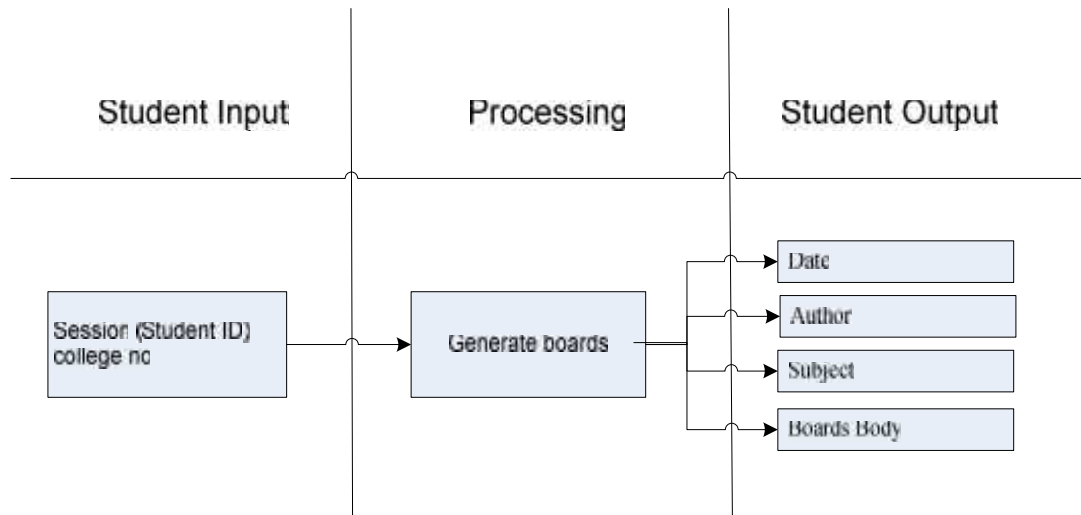
d. Flowchart:



Figure(3.12) College boards operation.



>> User interface design

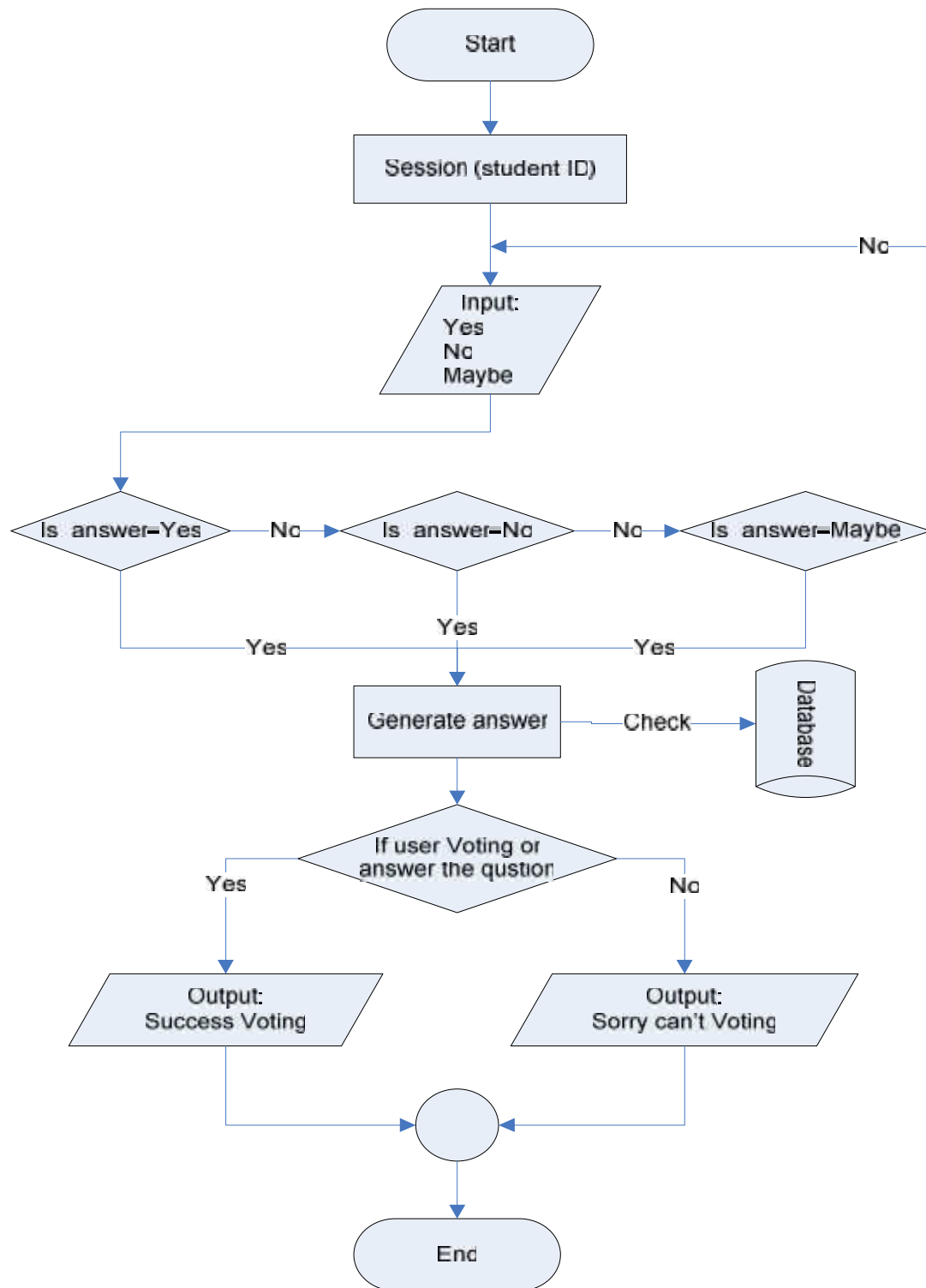


13- Student voting:

- a. description :This function enable the student to answer specific question for one time only.
- b. interface:
 - ◆ Input : selection only one answer (yes, or no, maybe)
 - ◆ Output : if he vote for first time, message with process done will appeared, else message that tell you that you were voting before will appeared.
- c. Constraints:
 - ◆ Student can't voting more than one time for the same question, and only one answer.
 - ◆ Student only read question and can't edit it.



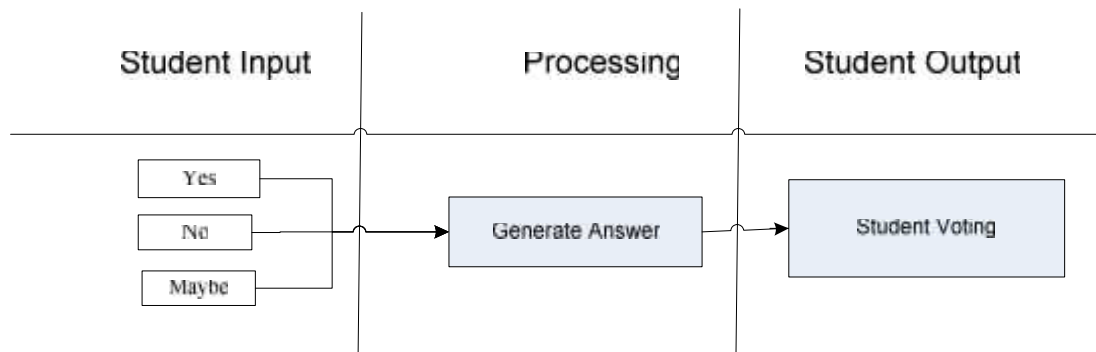
d. Flowchart:



Figure(3.13) Student voting operation.



>> User interface design



14- Chang Password :

a. Description :this function enable student to change his password .

b. Interface:

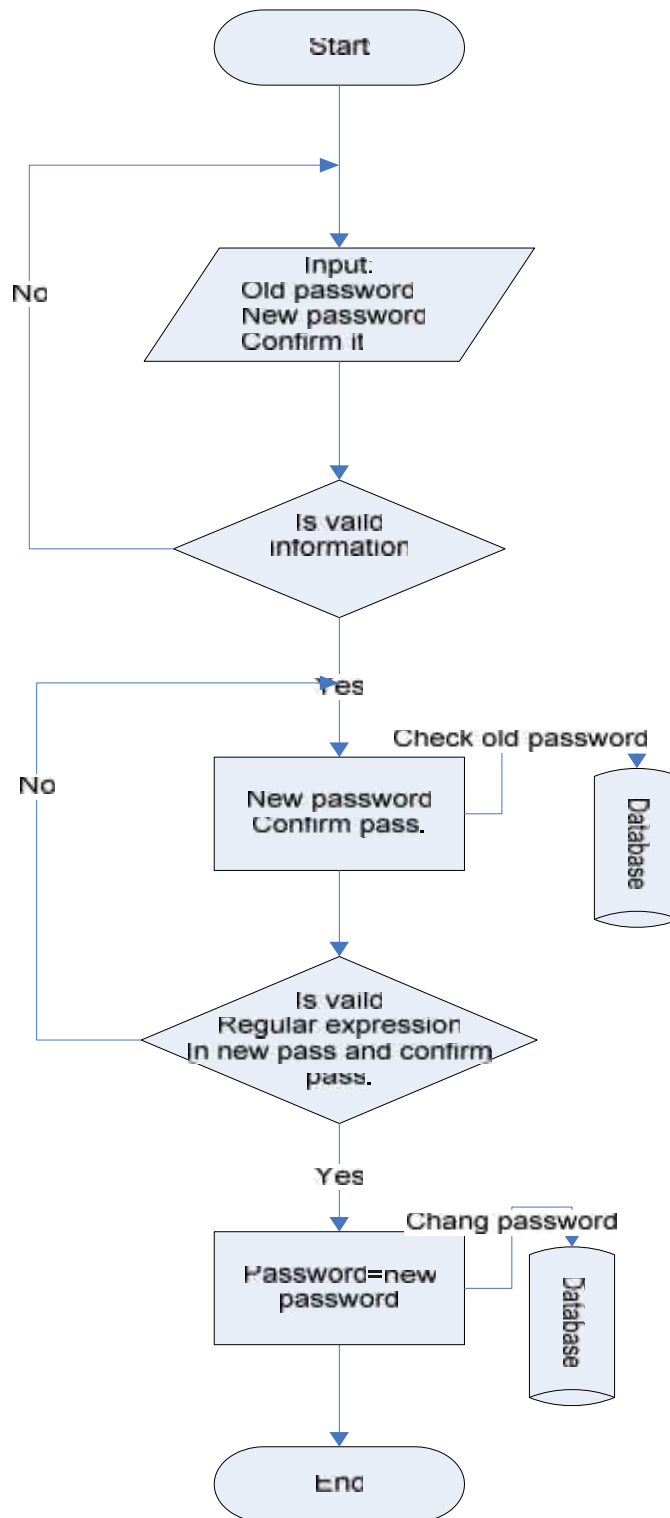
- ◆ Input : student id in session, old password, new password, confirm new password
- ◆ Output : new password (your password was updated)

c. Constraints:

- ◆ New password and its confirmation must match.
- ◆ The new password will take place at the next login .
- ◆ New password must be at least 6 characters.



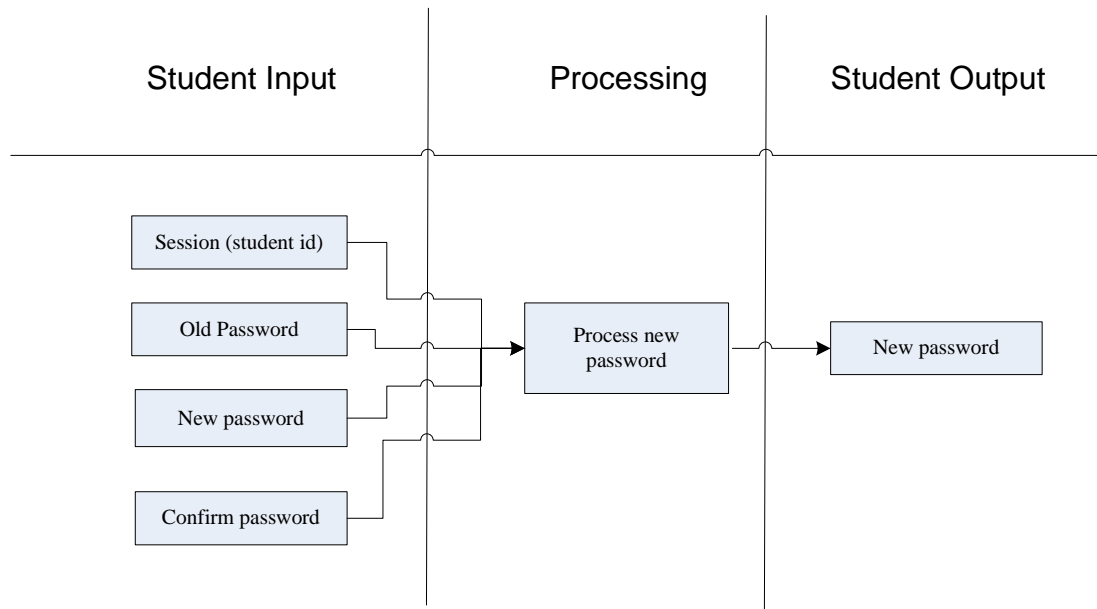
d. Flowchart:



Figure(3.14) Change password operation.



>> User interface design

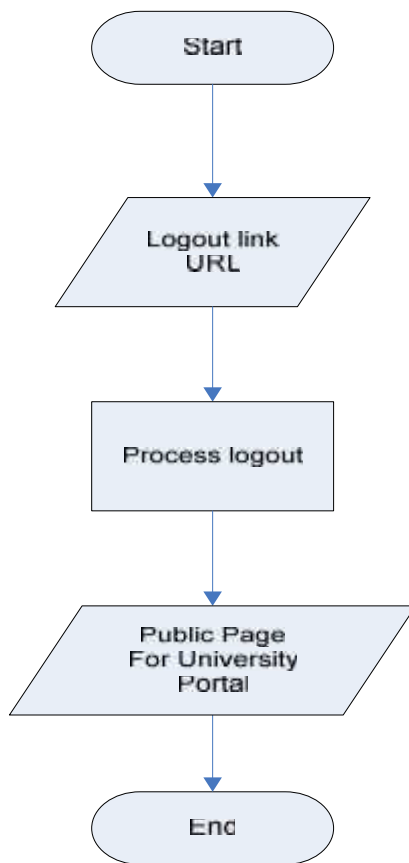


15- *Logout Student* :

- a. Description: This function logging student out of his session, his pass session will expired by take 0 value, and he go back to home page.
- b. Interface:
 - ◆ Input: Click on logout link.
 - ◆ Output: Home page.
- c. Constraints: None.

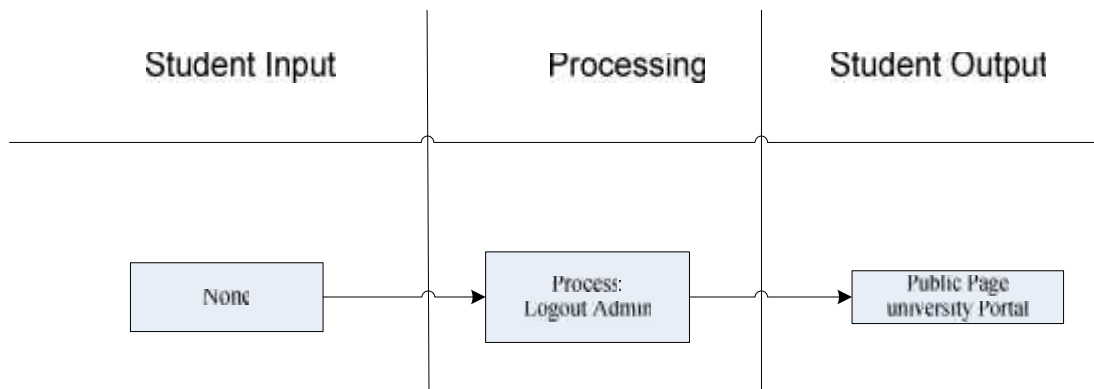


d. Flowchart:



Figure(3.15) Student logout operation.

>> User interface design;



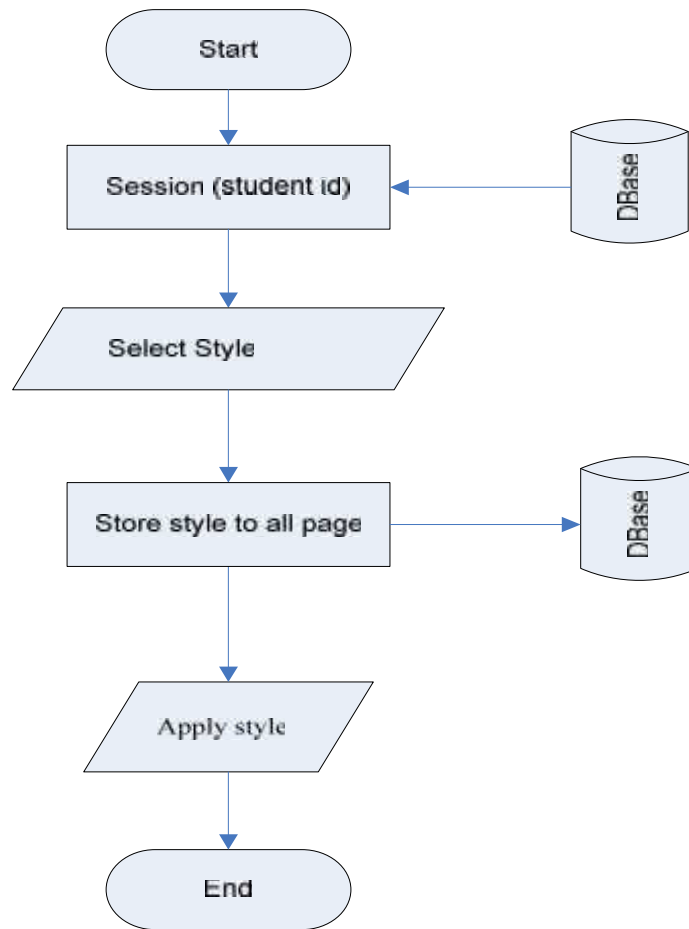


16- *My Style:*

- a. Description :This function enable student to customize his account interface by selecting one style from given list and apply it.
- b. Interface:
 - ◆ Input: Student id (username) in session, select his style.
 - ◆ Output: New interface style.
- c. Constraints:
 - ◆ Only authenticated student can show this page.
 - ◆ Students only can change there style.
 - ◆ New style apply only for the student who select it.
 - ◆ New style will apply after refresh the page.

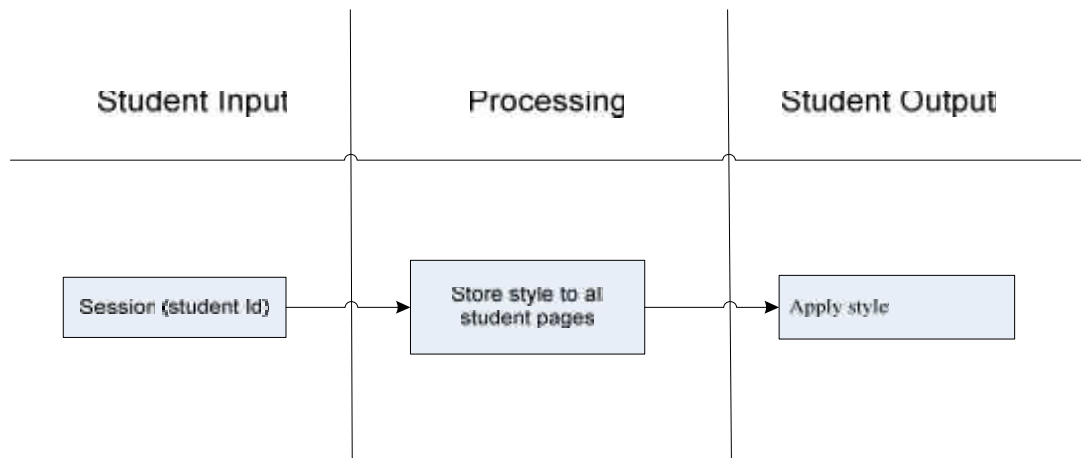


d. Flowchart :



Figure(3.16): My style operation

>> User interface design:





B. Administrator Functions design

1- Login Administrator :

a. Description :This function enable Administrator to login to the portal.

b. Interface:

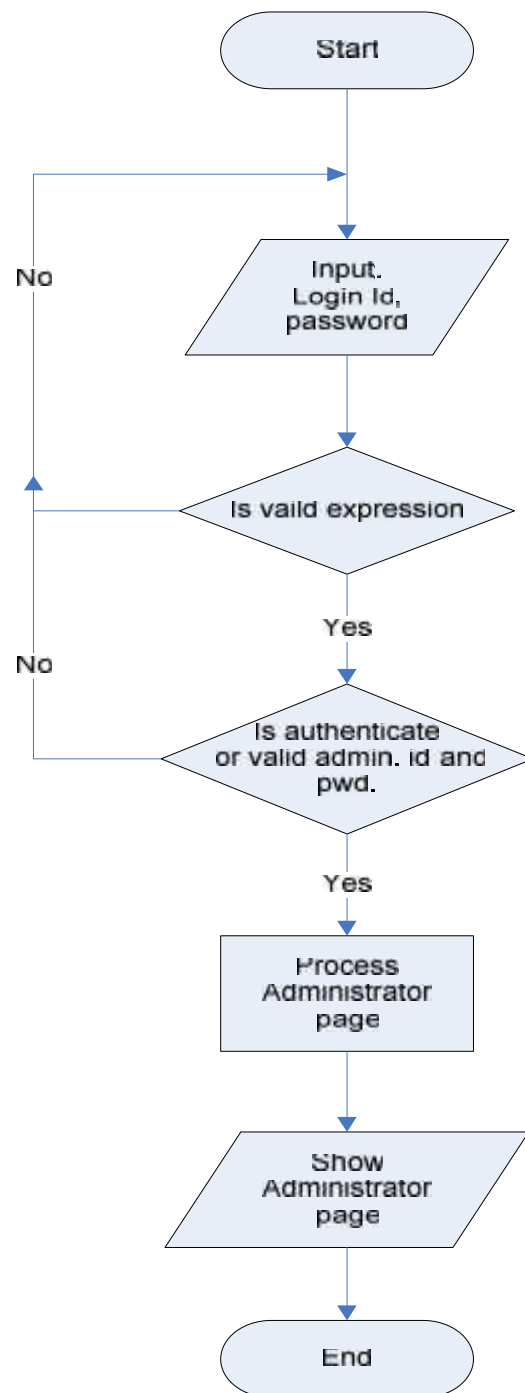
- ◆ Input : login id, password
- ◆ Output : Administrator desktop (home page).

c. Constraints:

- ◆ The password should be a character with at least 6 characters.
- ◆ The password must be encrypted.



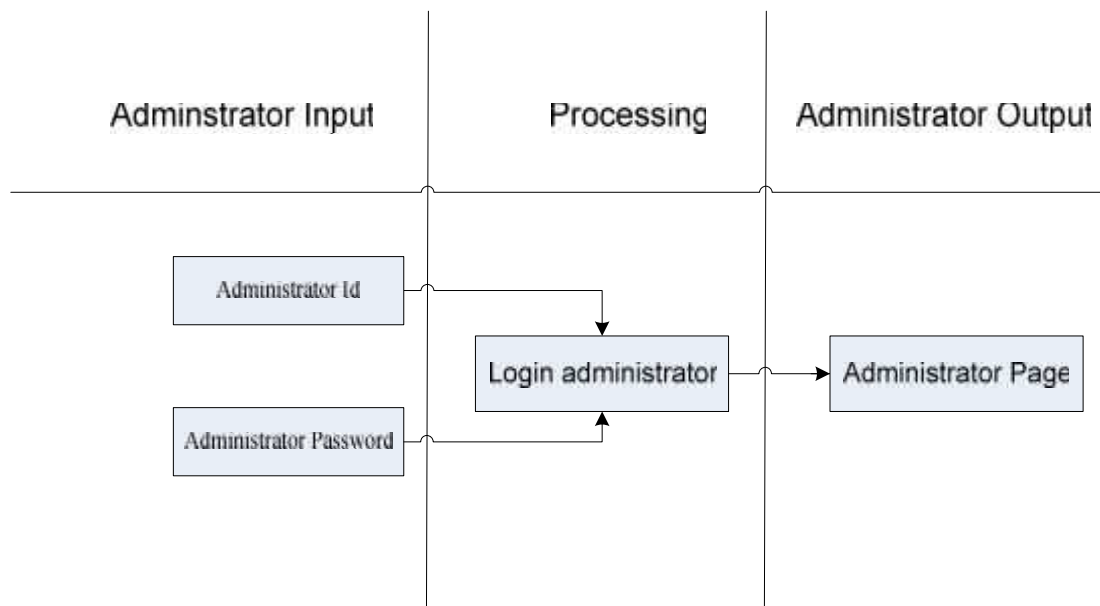
d. Flowchart:



Figure(3.17) Log in administrator operation.



>> User interface design



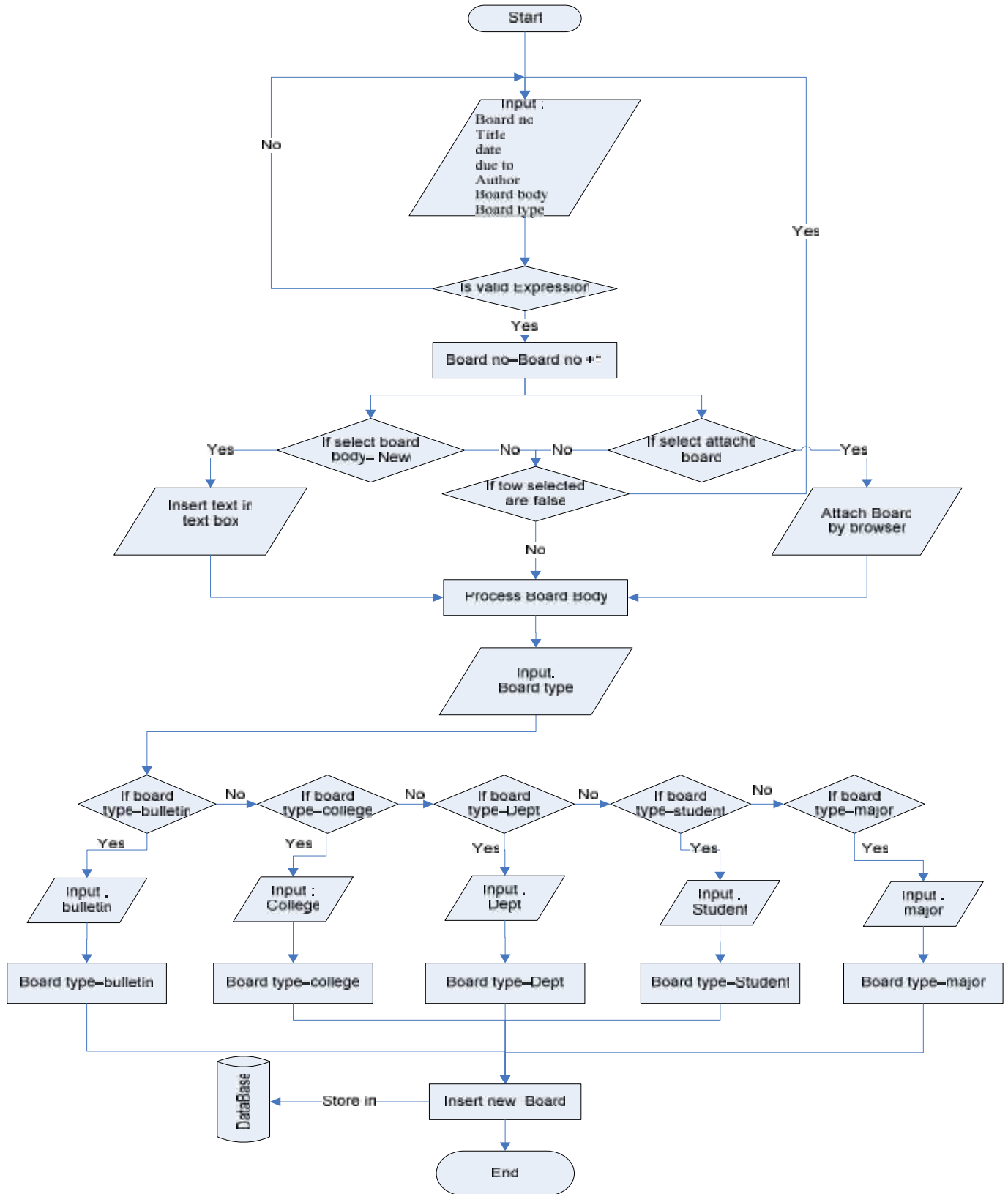
2- Insert boards :

- a. description :This function enable Administrator to upload new boards for students, colleges, departments, courses, or overall university (General boards).
- b. interface:
 - ◆ Input : Board No, Body text or attachment or both, board type, Title, issue date, due to date, Author, destination.
 - ◆ Output : New Board uploaded.
- c. Constraints:
 - ◆ Board body can be text or attachment or both.
 - ◆ Board can be uploaded for more than one destination.



- ◆ Authenticated administrator can basically upload four types of boards(student, college, department, courses), and public board
- ◆ Only authenticated administrator can show this page.

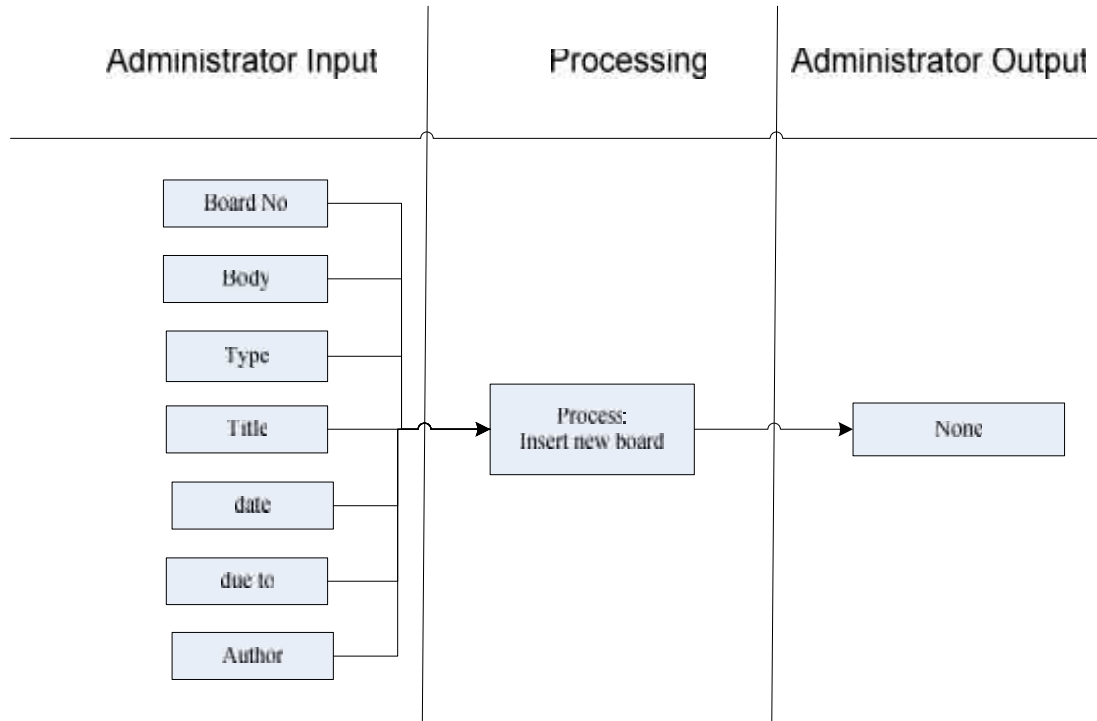
d. Flowchart:



Figure(3.18) Insert board operation.



>> User interface design



3- Delete Boards :

a. description :This function allow the Administrator to delete the boards according several criteria (expired boards, by name, boards type) .

b. interface:

- ◆ Input : Expired board ,board name ,board type then select board name to be deleted, put it in recycle bin, click delete button.
- ◆ Output : deleted board

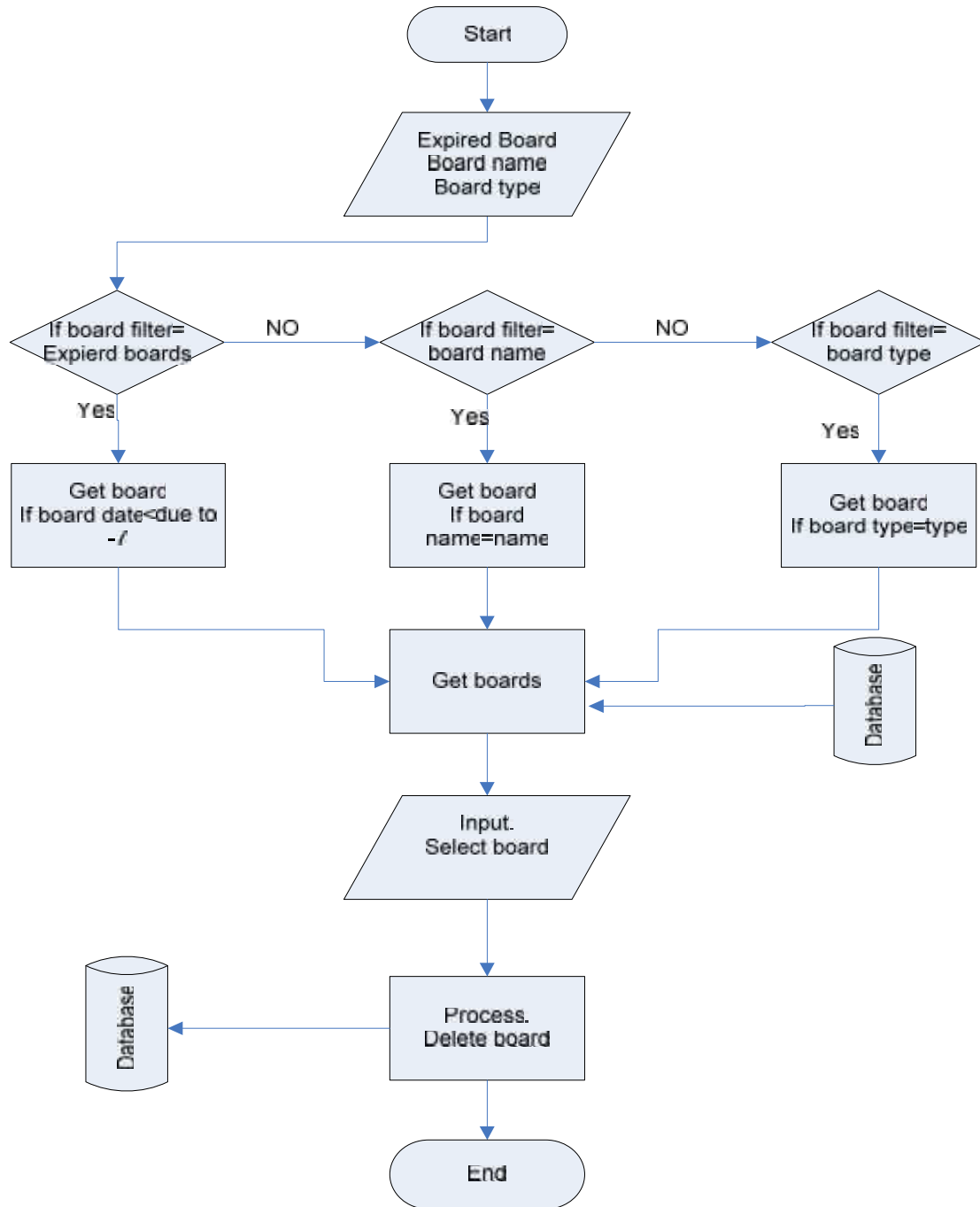
c. Constraints:

- ◆ Authenticated administrator only can delete boards.



- ◆ Boards can be deleted manually by administrator or automatically by the system (further work).

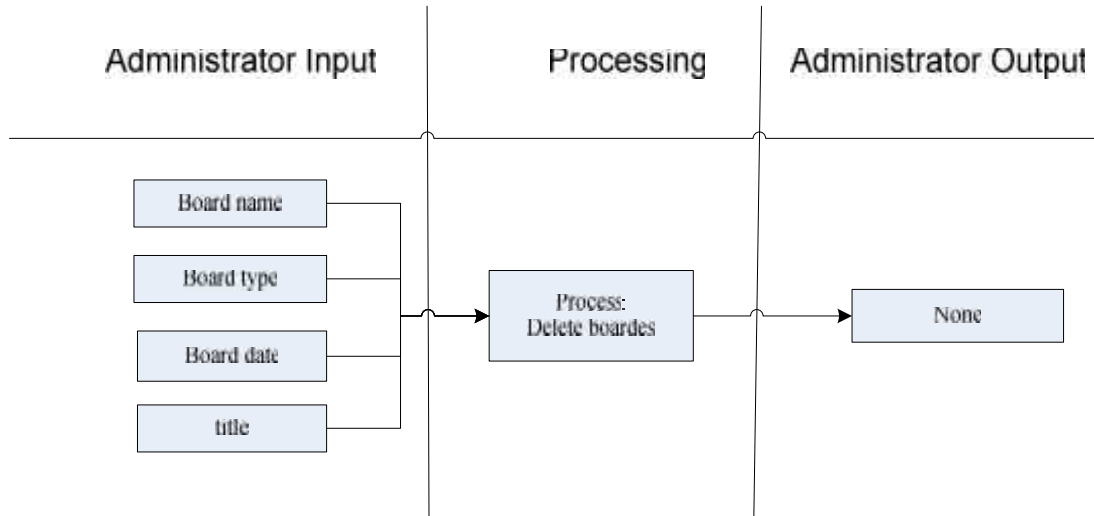
d. Flowchart:



Figure(3.19) Delete boards operation.



>> User interface design



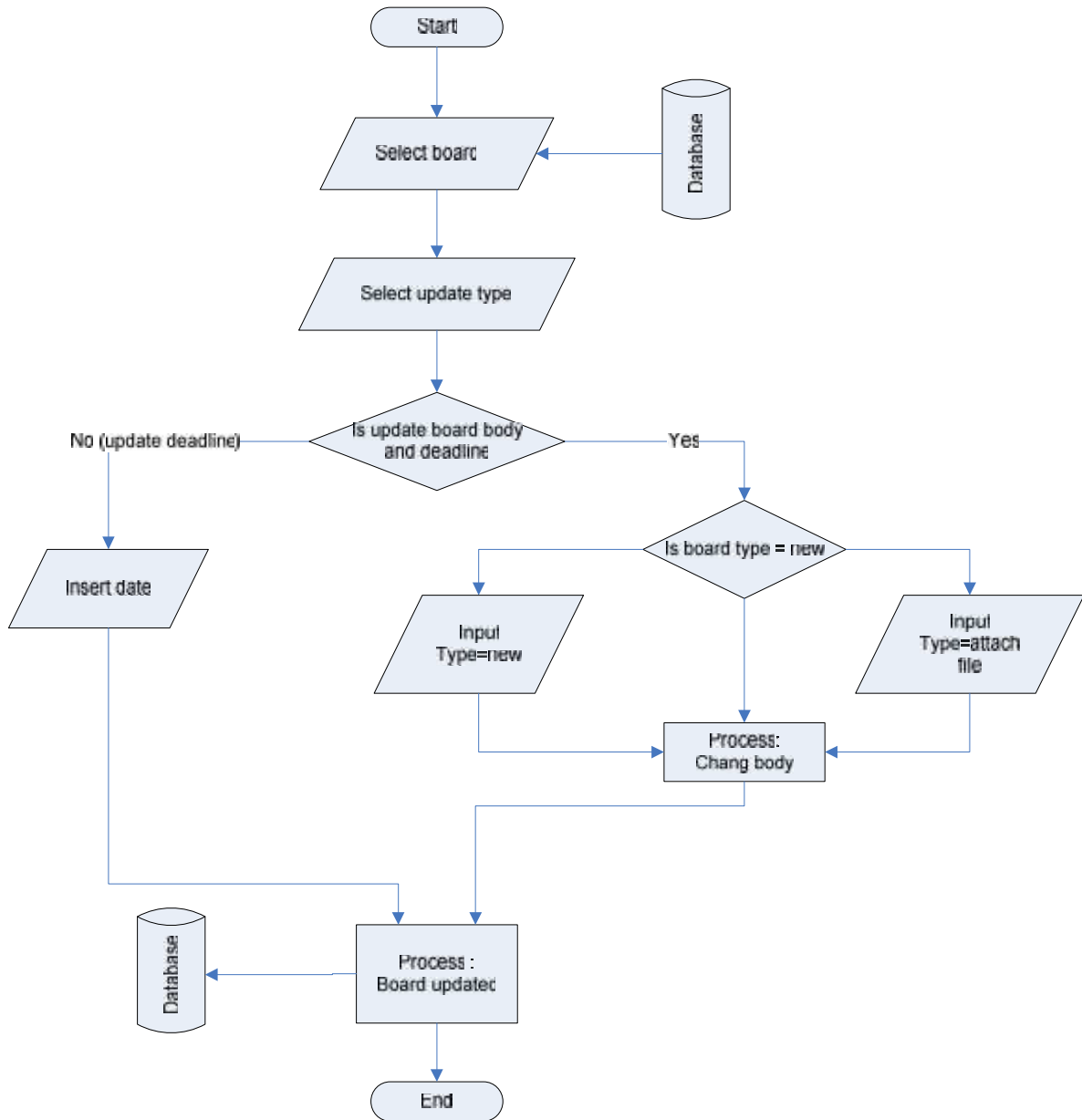
4- Update Boards :

- a. description :This function allow the Administrator to update the existing board body or board due to date or both.
- b. interface:
 - ◆ Input : board name, board title, board type(exit, or new), new board dead line, or new board body, or both.
 - ◆ Output : updated board
- c. Constraints:
 - ◆ Boards that can be updated generated automatically in drop down list when page request.



- ◆ Only authenticated administrator can update boards.
- ◆ Updated boards will be appeared after refresh the page.
- ◆ Board due to date must be in correct format (month/day/year).

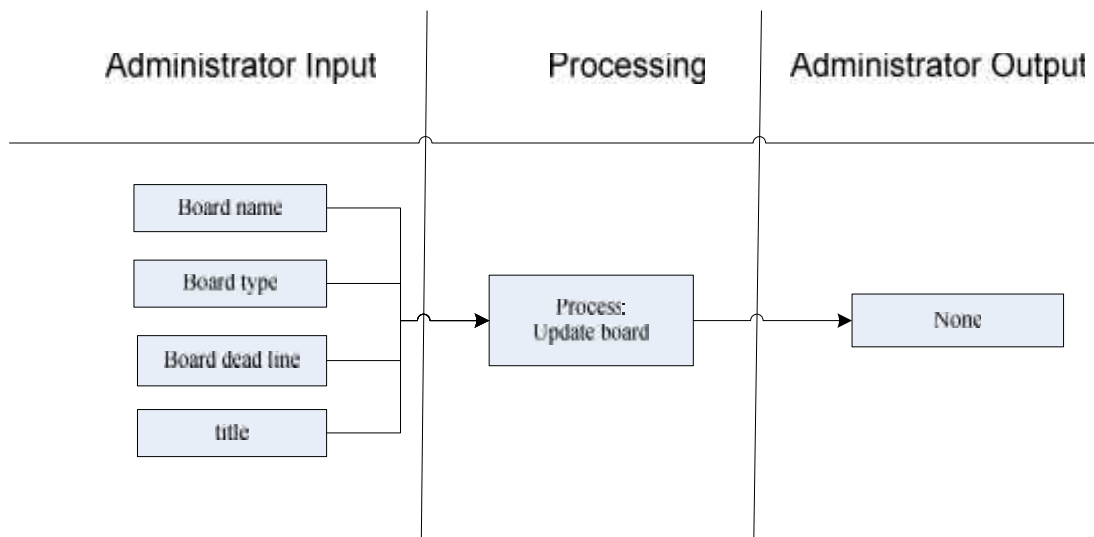
d. Flowchart:



Figure(3.20) Update boards operation.



>>User interface design

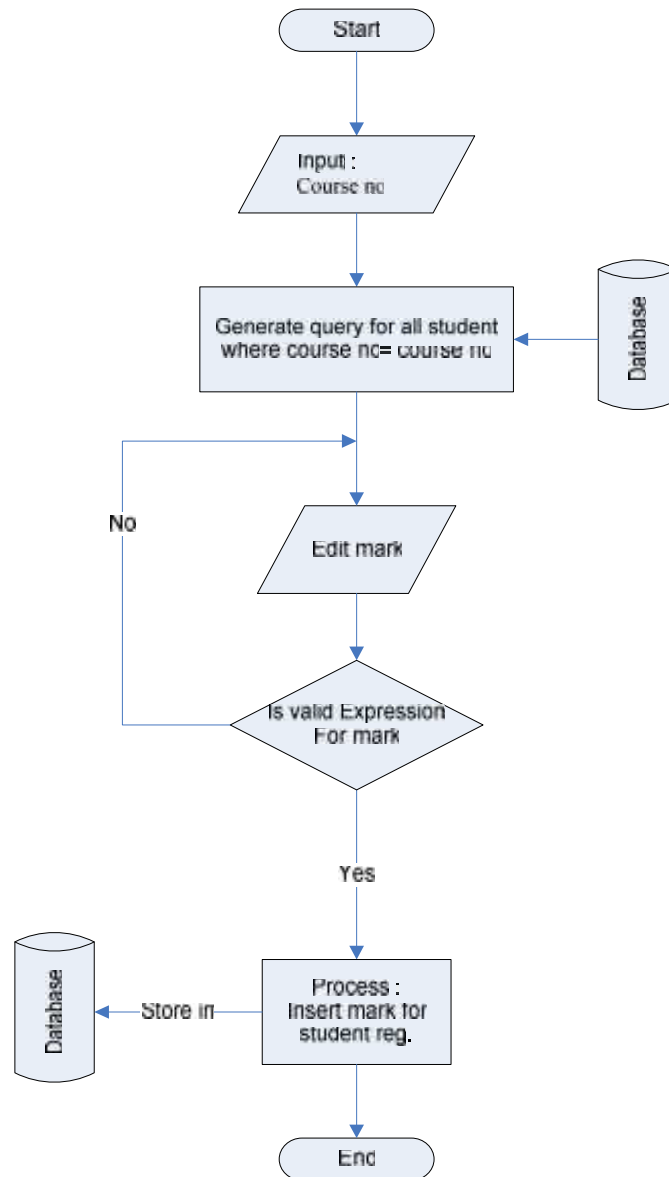


5- Insert Marks :

- a. description :This function enable Administrator to update the students marks, depending on course number.
- b. interface:
 - ◆ Input : course no, student mark .
 - ◆ Output : Update students marks
- c. Constraints:
 - ◆ All enrolled courses will appear in drop down list automatically when request the page.
 - ◆ Only students that enrolled the selected course will appear in table with there information.
 - ◆ The inserted mark can't be less then 40 or more then 100, and can't be in negative value.



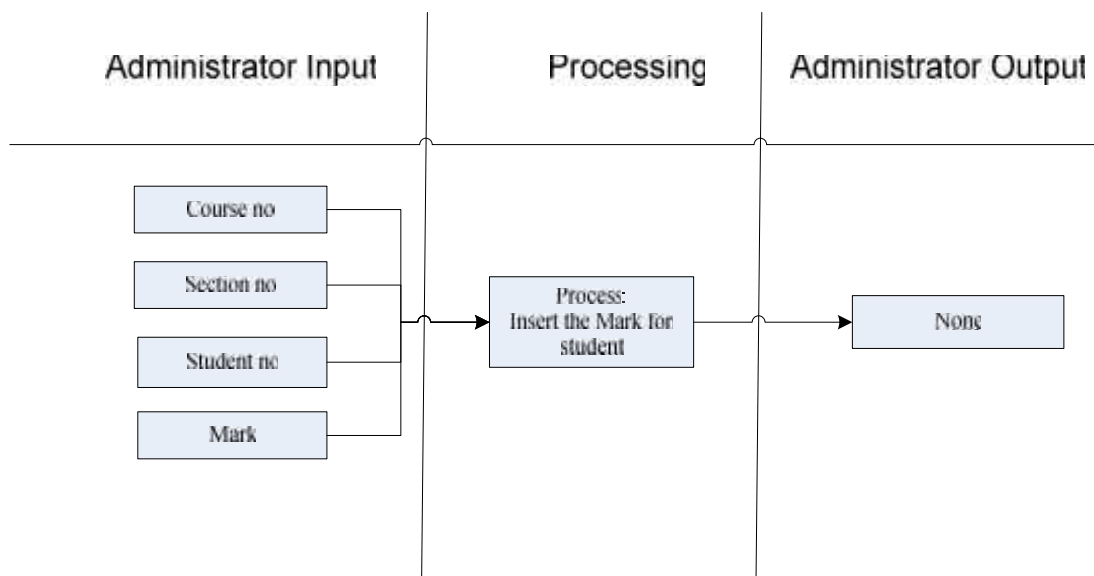
d. Flowchart:



Figure(3.21) Insert marks operation.



>>User interface design



6- Insert College :

a. description :

This function allow the Administrator to Insert the new college information to the portal.

b. interface:

- ◆ Input : College No, College Name, College Description
- .
- ◆ Output : new college

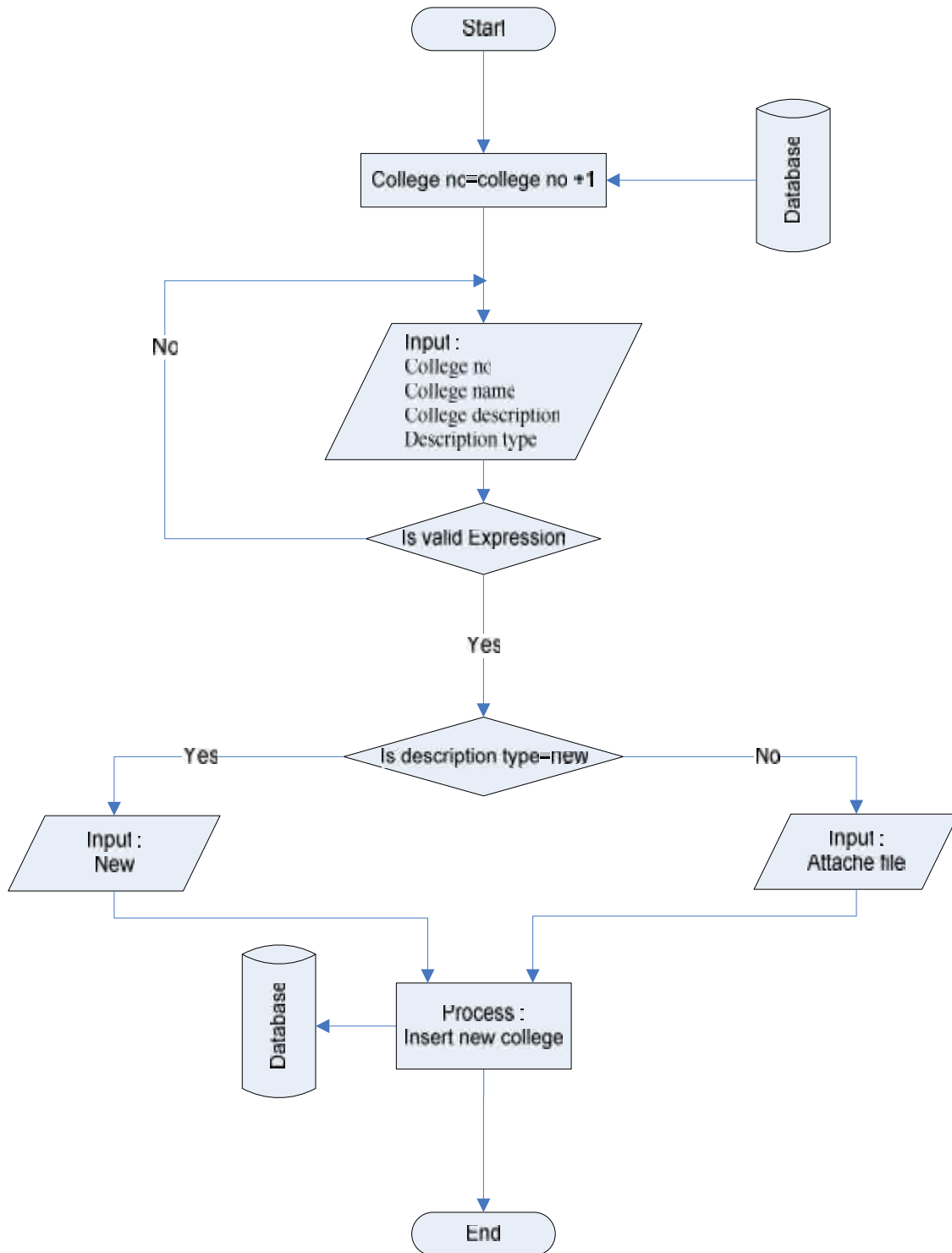


c. Constraints:

- ◆ New college number generated automatically .
- ◆ New college description can be either text or attachment file or both.
- ◆ New college name can't be number, or start with number.



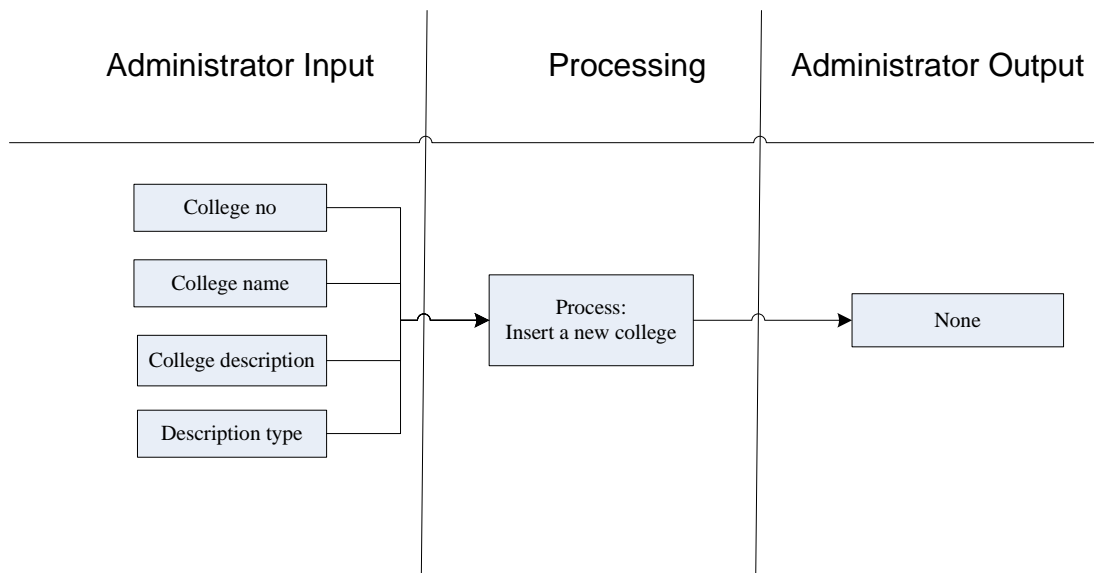
d. Flowchart:



Figure(3.22) Insert new college operation.



>>User interface design



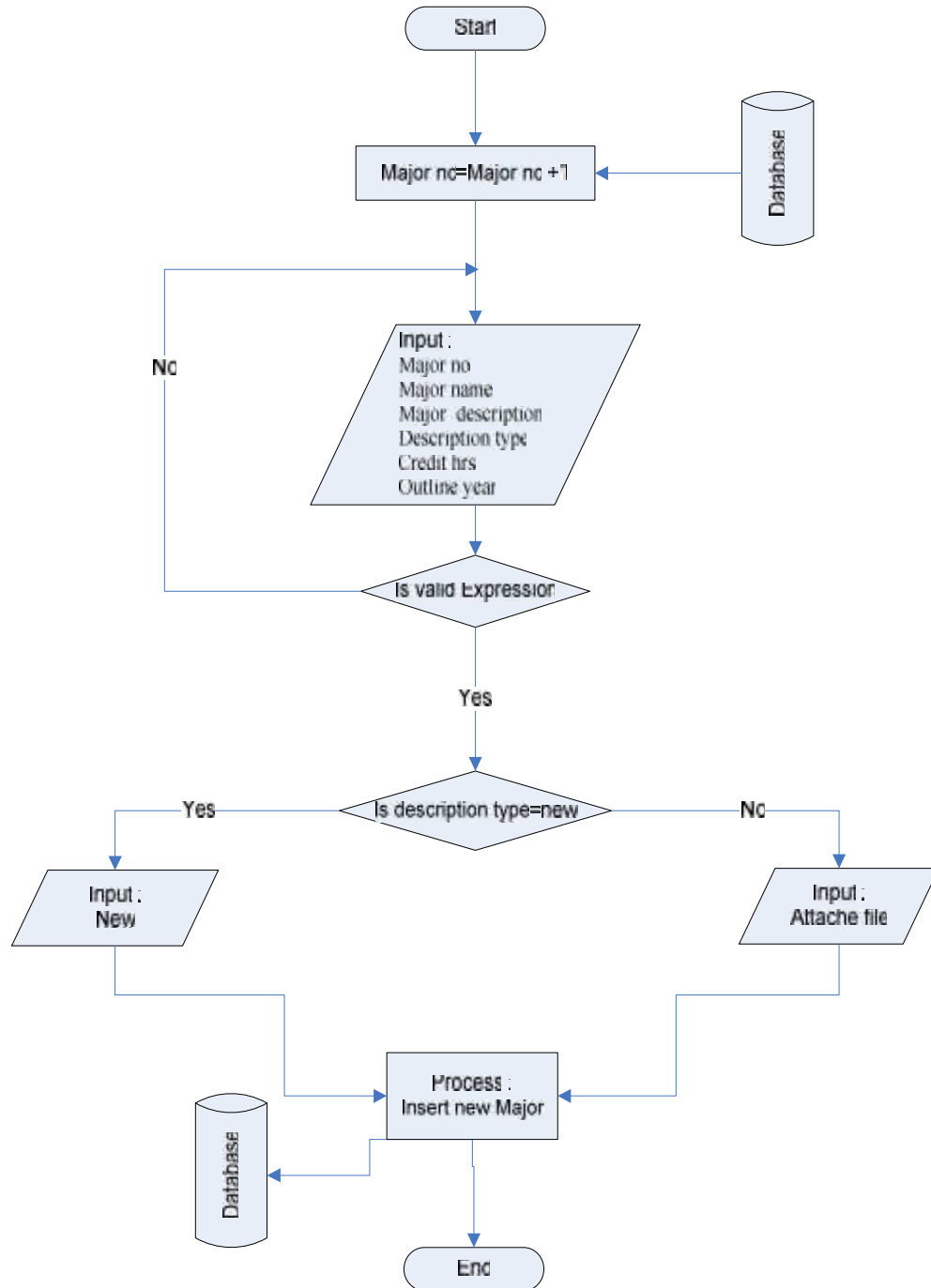
7- Insert Major :

- a. description :This function allow the Administrator to Insert a new major information to the portal.
- b. interface:
 - ◆ Input : Major No, Major Name, Major Description, Description type, Credit HRS, Outline year.
 - ◆ Output : new Major
- c. Constraints:
 - ◆ New major number generated automatically .
 - ◆ New major description can be either text or attachment file or both.
 - ◆ Outline must be number.



- ◆ Credit hours can't be less than 1 or more than 4.

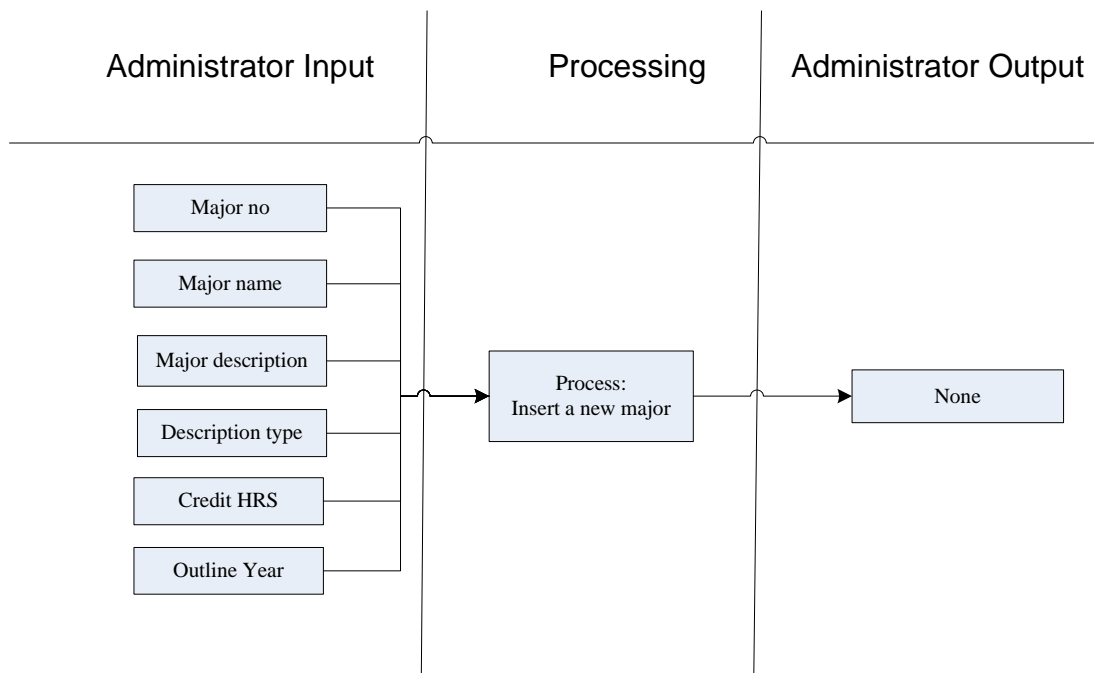
d. Flowchart:



Figure(3.23) insert major operation.



>>User interface design



8- Change Password for administrator :

a. description :This function enable Administrator to change his password .

b. interface:

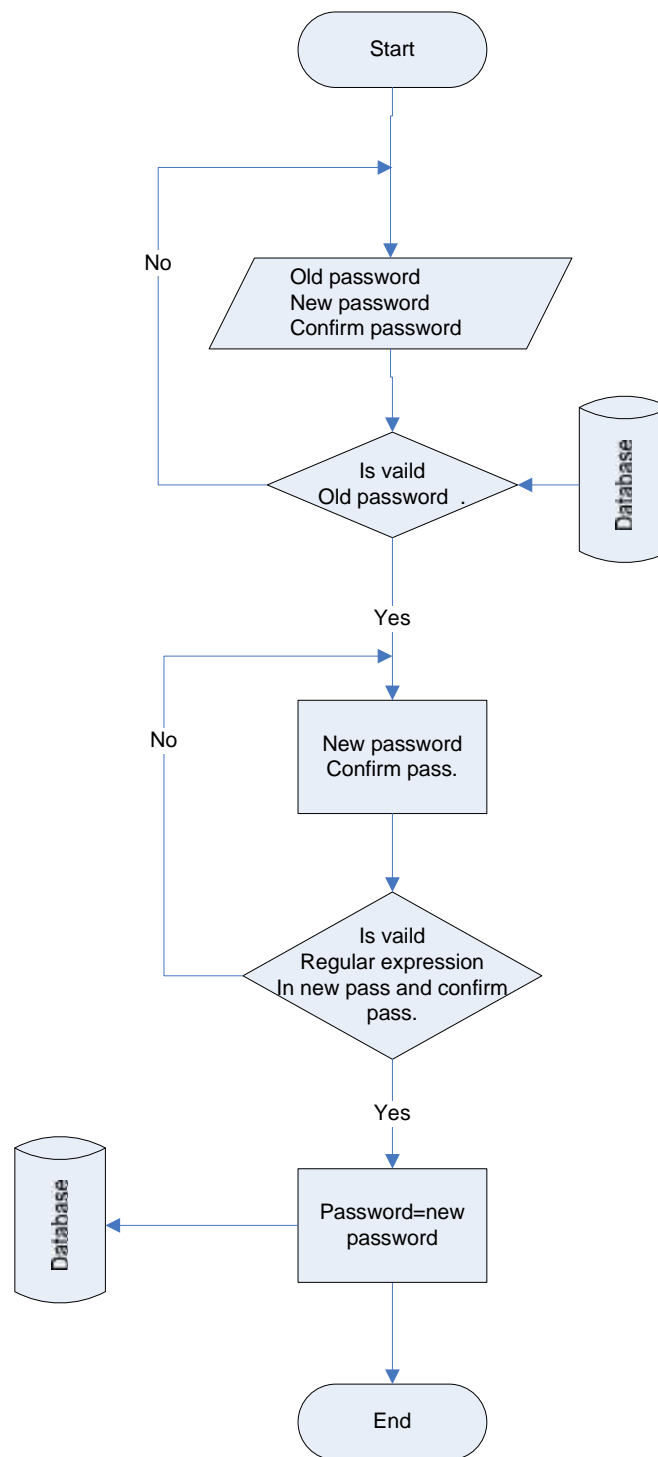
- ◆ Input : Admin id, old password, new password, confirm password
- ◆ Output : new password

c. Constraints:

- ◆ New password and its confirmation must match.
- ◆ The new password will take place at the next login .
- ◆ New password must be in character and at least with 6 character.



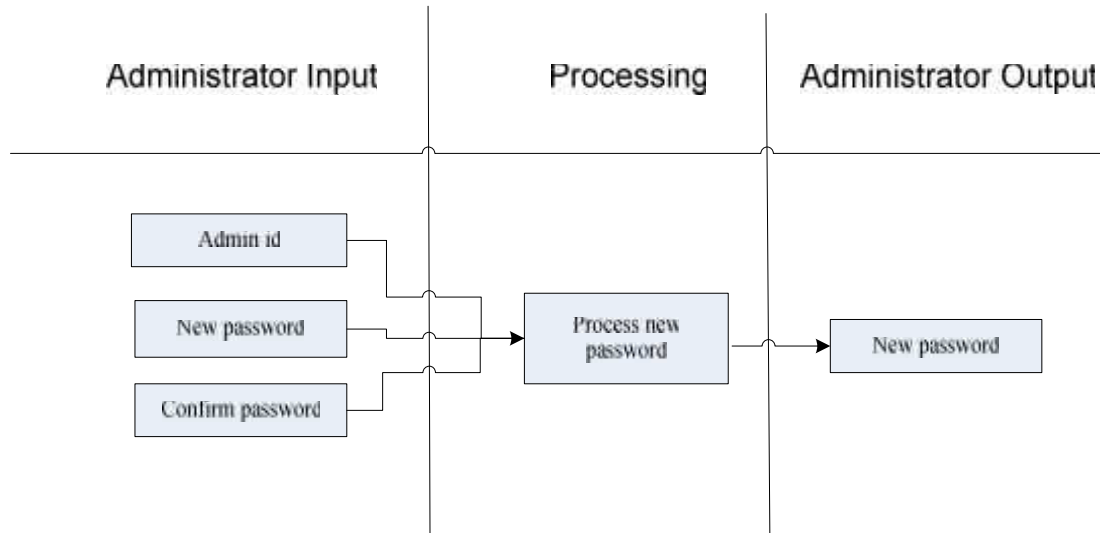
d. Flowchart:



Figure(3.24) Chang administrator's password.



>>User interface design



9- Logout administrator

a. Description:

This function make administrator logging out from his session work, and terminate his pass session, and go back to home page .

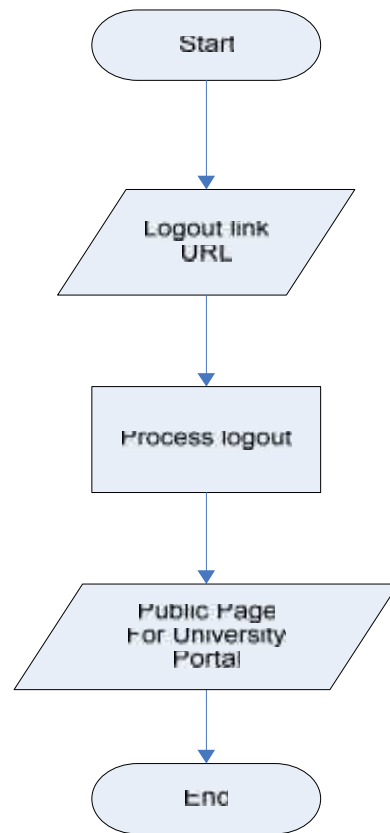
b. Interface:

- ◆ Input: Click on logout link.
- ◆ Output: home page.

c. Constraints: None.

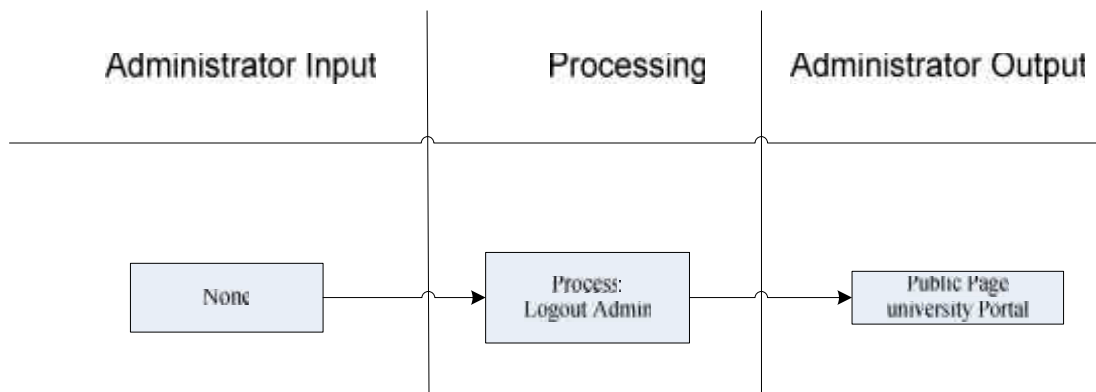


d. Flowchart:



Figure(3.25) Administrator logout operation.

>> User interface design:





C. Instructor Functions design

1- Insert board :

a. description :This function enable instructor to upload new boards Only for courses who enrolled and student who teaching them..

b. interface:

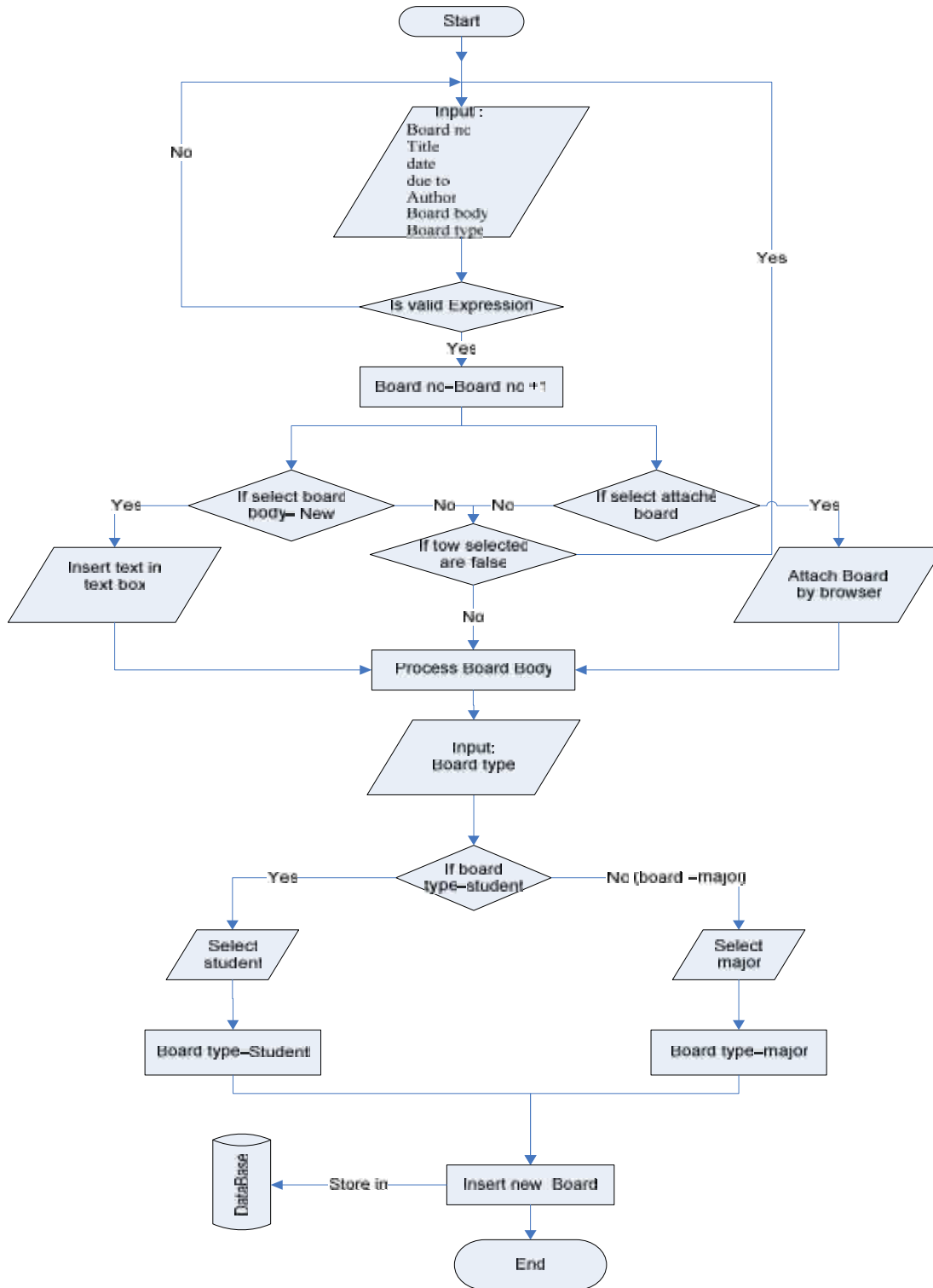
- ◆ Input : Board No, Body, Type, Title, issue date, due to date, Author, destination.
- ◆ Output : New Board.

c. Constraints:

- ◆ Instructor can only upload courses and students boards.
- ◆ Instructor can only upload boards for his students and his enrolled courses.
- ◆ Instructor have no authentication to delete or update boards.



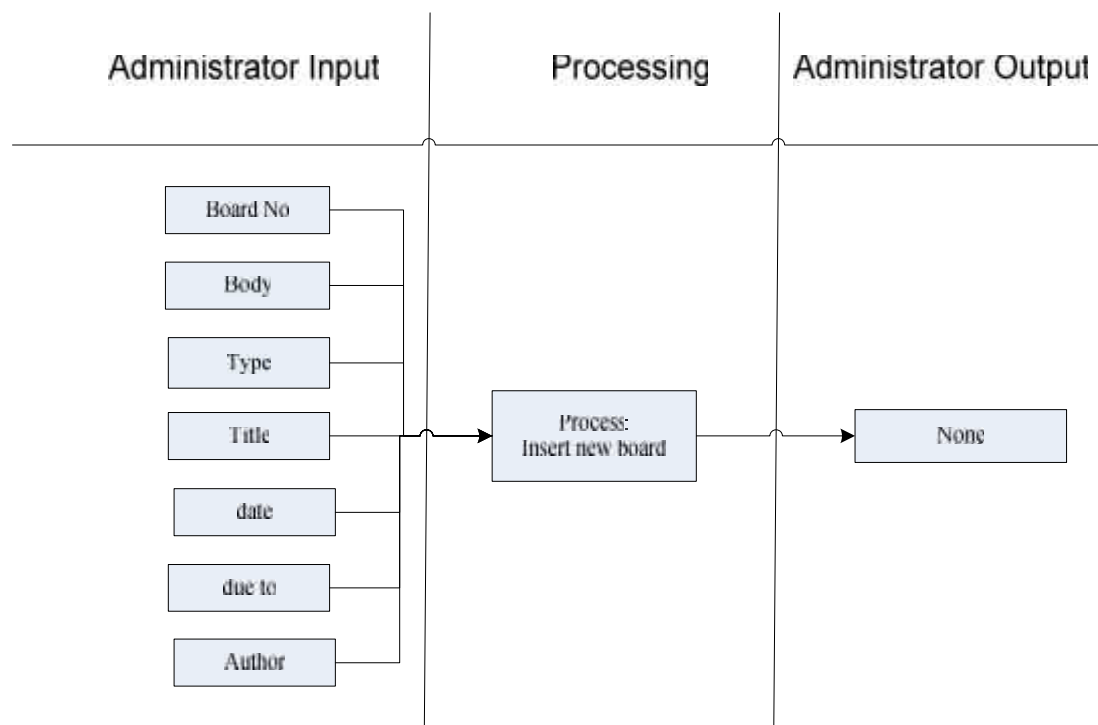
d. Flowchart:



Figure(3.26) Instructor insert board operation.



>>User interface design



2- Insert Marks :

a. description :This function enable instructor to update marks for his students in his enrolled courses according section .

b. interface:

- ◆ Input : course no, section no, student mark
- ◆ Output : updated students marks

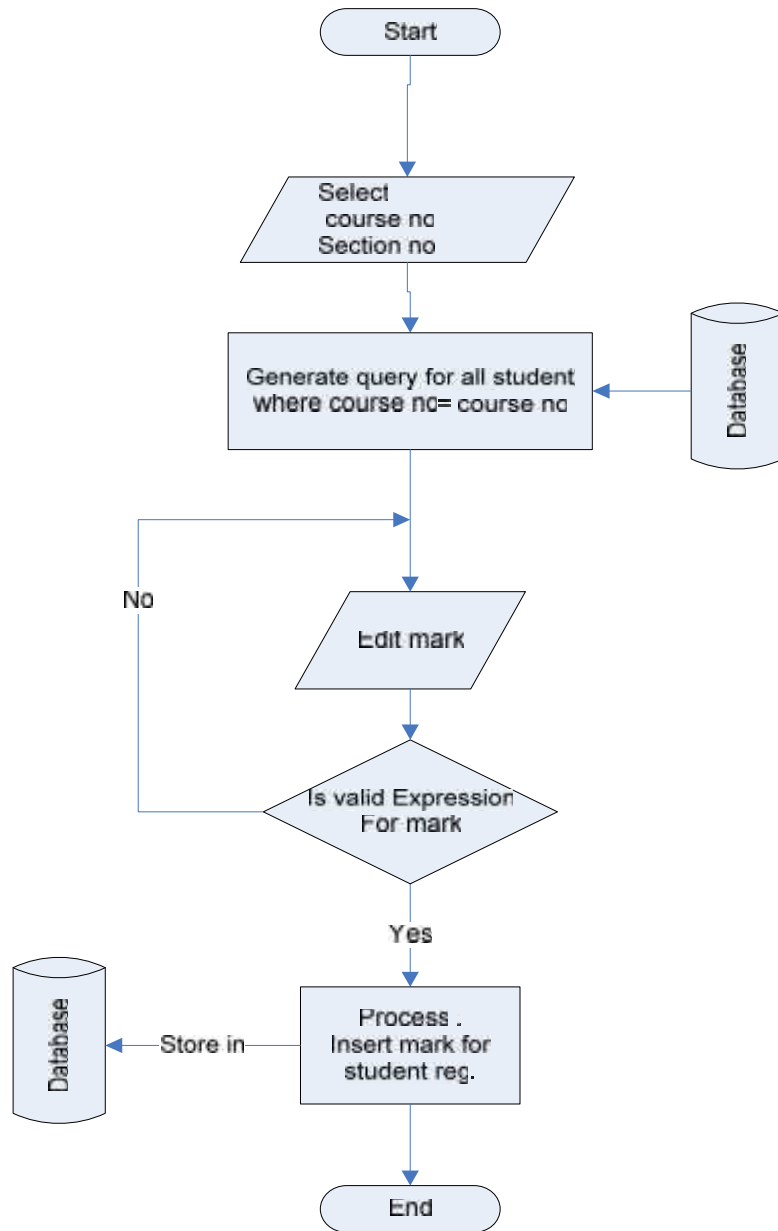
c. Constraints:

- ◆ Student mark can't be less than 40 or more than 100, or in negative value..



- ◆ Instructor can update marks only for students that enrolled his enrolled courses and specially for his sections

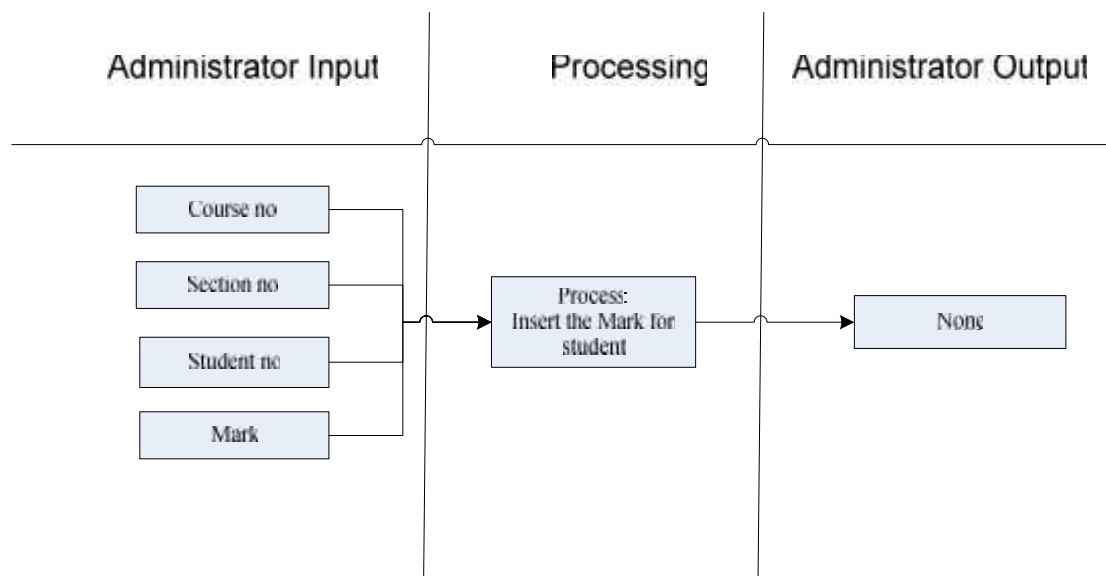
d. Flowchart:



Figure(3.27) Insert marks operation.



>>User interface design



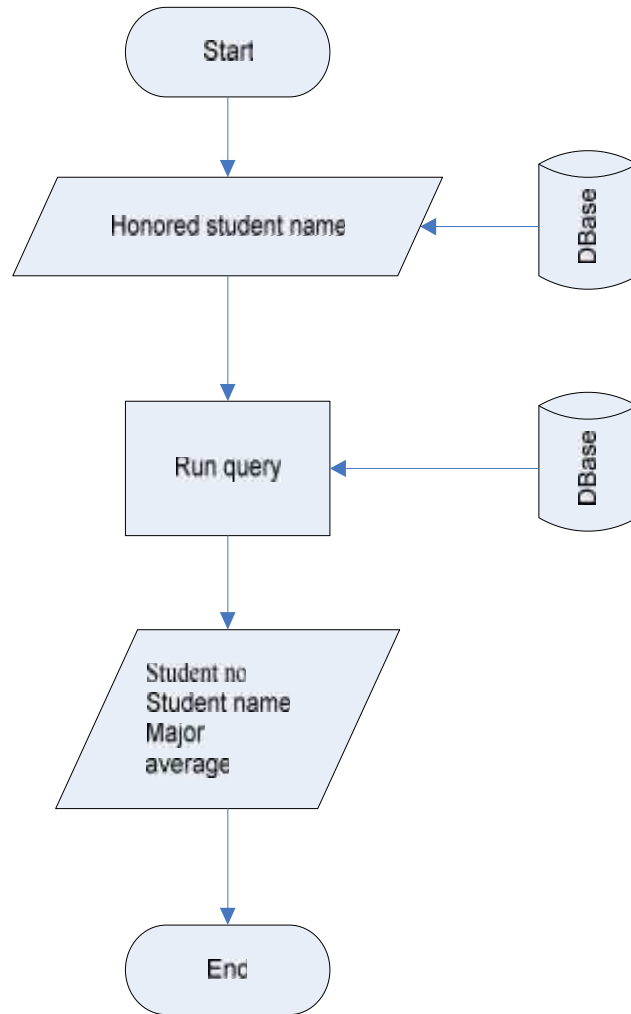
D- Public Functions design

1- Honored students :

- a. description : Display all honored students names in a vertical marquee.
- b. interface:
 - ◆ Input : student name.
 - ◆ Output : std no, std name, major, average.
- c. Constraints:
 - ◆ None.



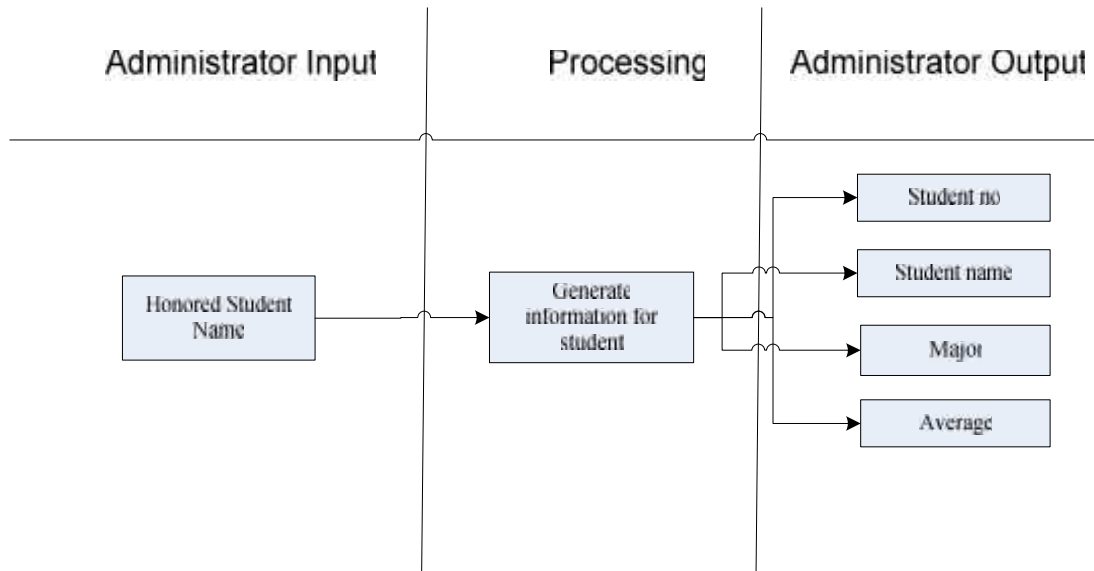
d. Flowchart



Figure(3.28) Honored students operation.



>>User interface design



2- Public search :

a. description : search that devoted to all visitors to the web site, this search restricted on portal database for colleges, courses, departments, boards and employees.

a. interface:

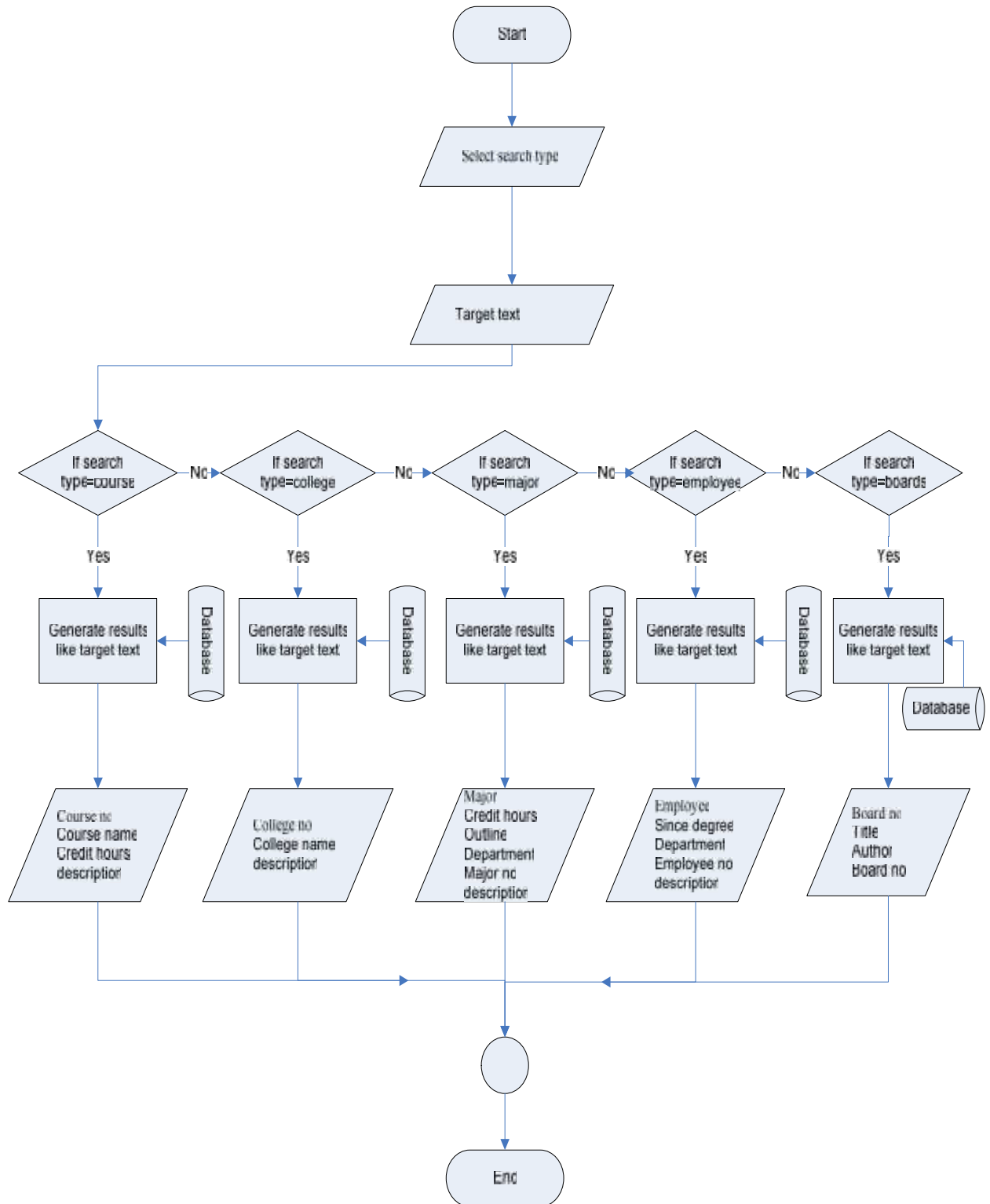
- ◆ Input: search type, target search prefix.
- ◆ Output: desired search item.

b. Constraints:

- ◆ None.



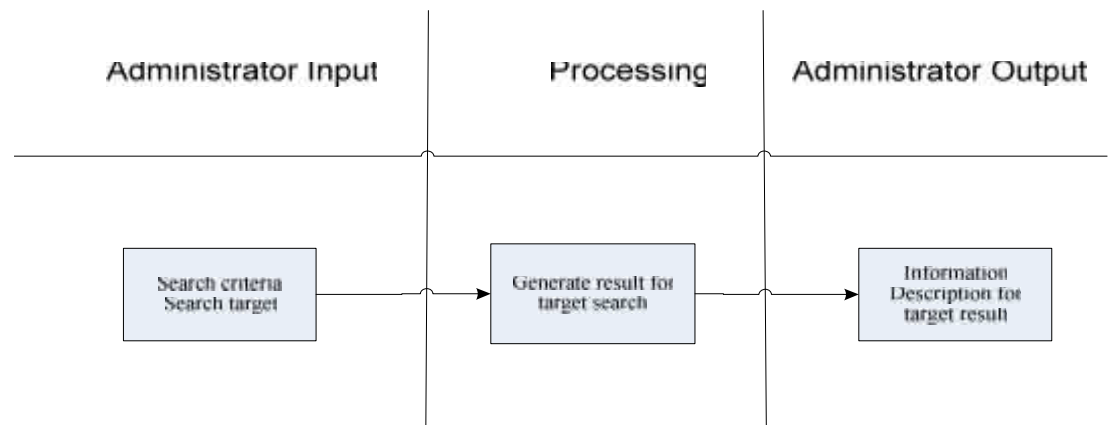
c. Flowchart



Figure(3.29) Public search operation.



>>User interface design



3- Marquee Boards :

a. description :

Displays all boards issued since seven days in a scrolled marquee on top of the home page, this marquee contains only colleges and department boards.

b. interface:

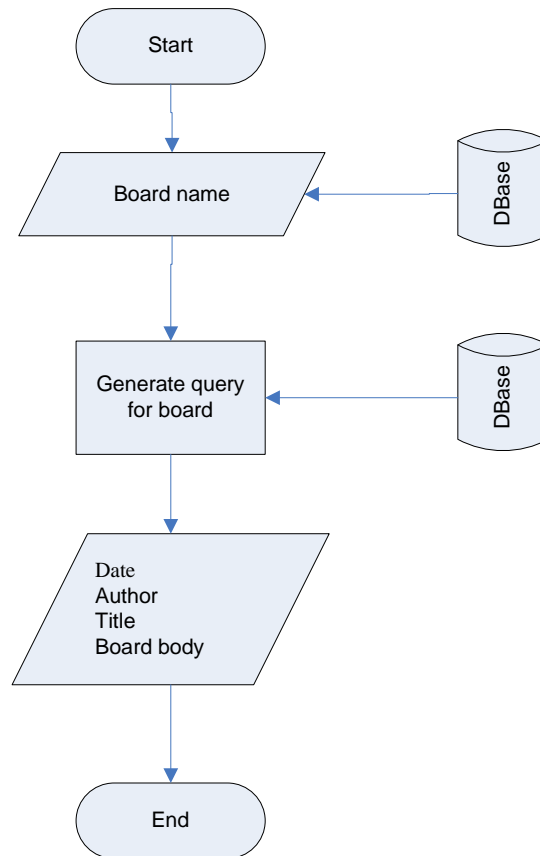
- ◆ Input: select board name from marquee.
- ◆ Output: board body.

c. Constraints:

- ◆ None.

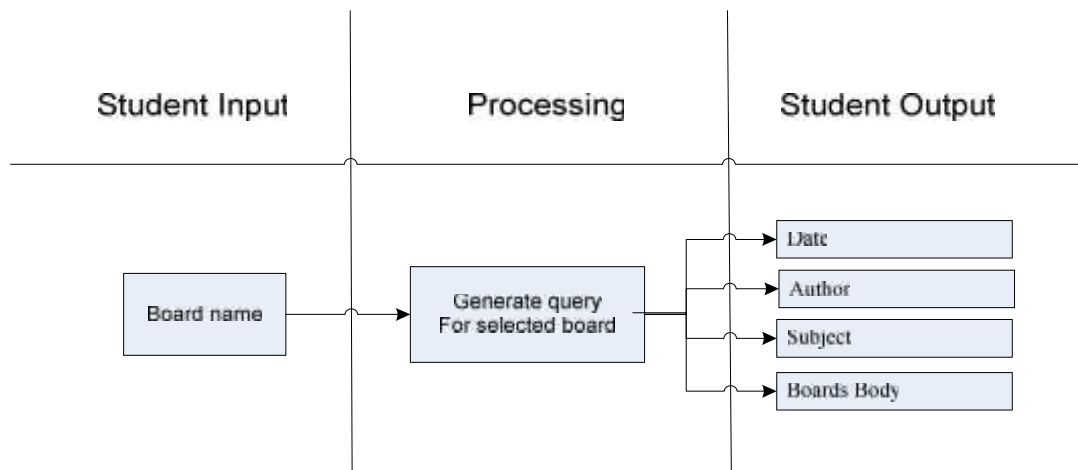


d. Flowchart



Figure(3.30) Marquee boards operation.

>>User interface design



F



4- Graduate project :

a. description :

this function enable the PAP visitor to show all graduate projects information according major name. and he can download it if it belongs to attachments type.

b. interface:

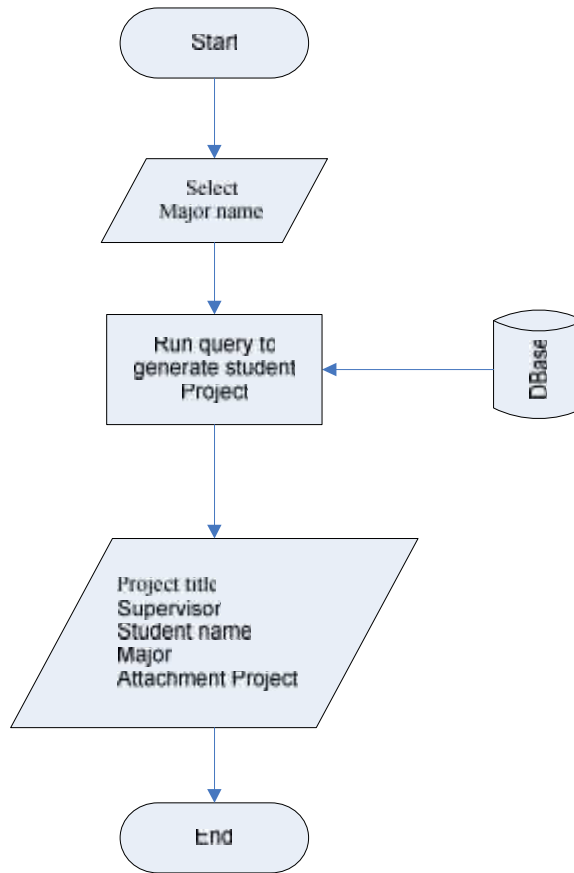
- ◆ Input: Major name.
- ◆ Output: project name, project supervisor, major, project attachment.

c. Constraints:

- ◆ None.

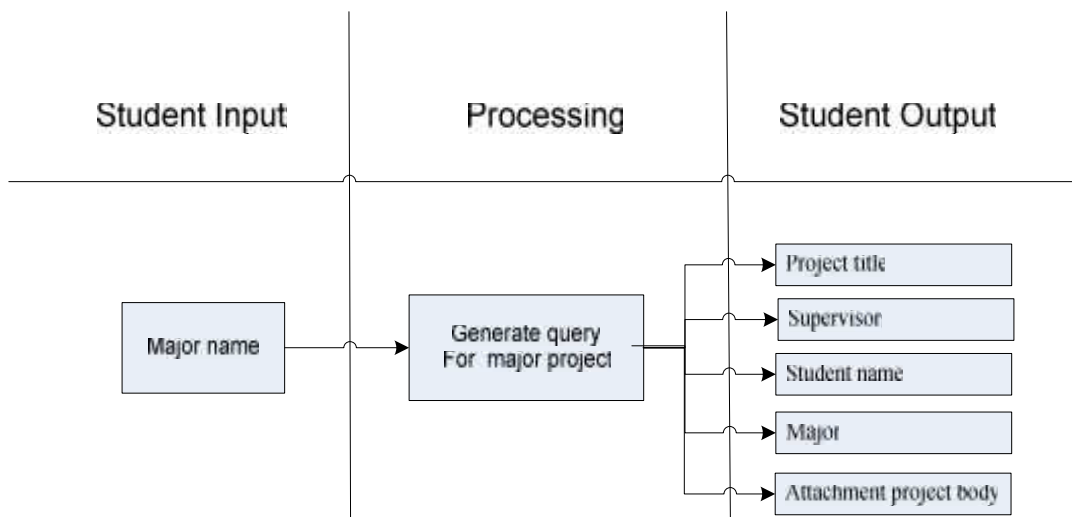


d. Flowchart



Figure(3.31) Graduate projects operation.

>>User interface design





3.3 Input Output design :

After describing requirements specification and system functions, we must show input/output screens. The designed forms shows how function works and shows how user can interact with it . and these steps of designing I/O considered as a first step for implementing and developing the software system .

In this chapter we design I/O user interface design, including student I/O design and Administrator I/O design.

A. Student interface design :

a. Student Input design:

- Student login:

This screen used to enable student to login to his account.

The image shows a screenshot of a web browser window titled "Login". The window has a blue border and standard window control buttons (minimize, maximize, close) in the top right corner. The main content area is white and contains a centered heading "Please Insert Your ID And Password". Below this heading are two input fields: "Student Name :" followed by a text box, and "Password :" followed by a text box. Below the input fields is a "Login" button. At the bottom of the form area, there is a blue underlined link that says "Forgot My Password?".

Figure (3.32) student login screen.



➤ Student Menu :

This screen contains student menu, all student services displayed here.

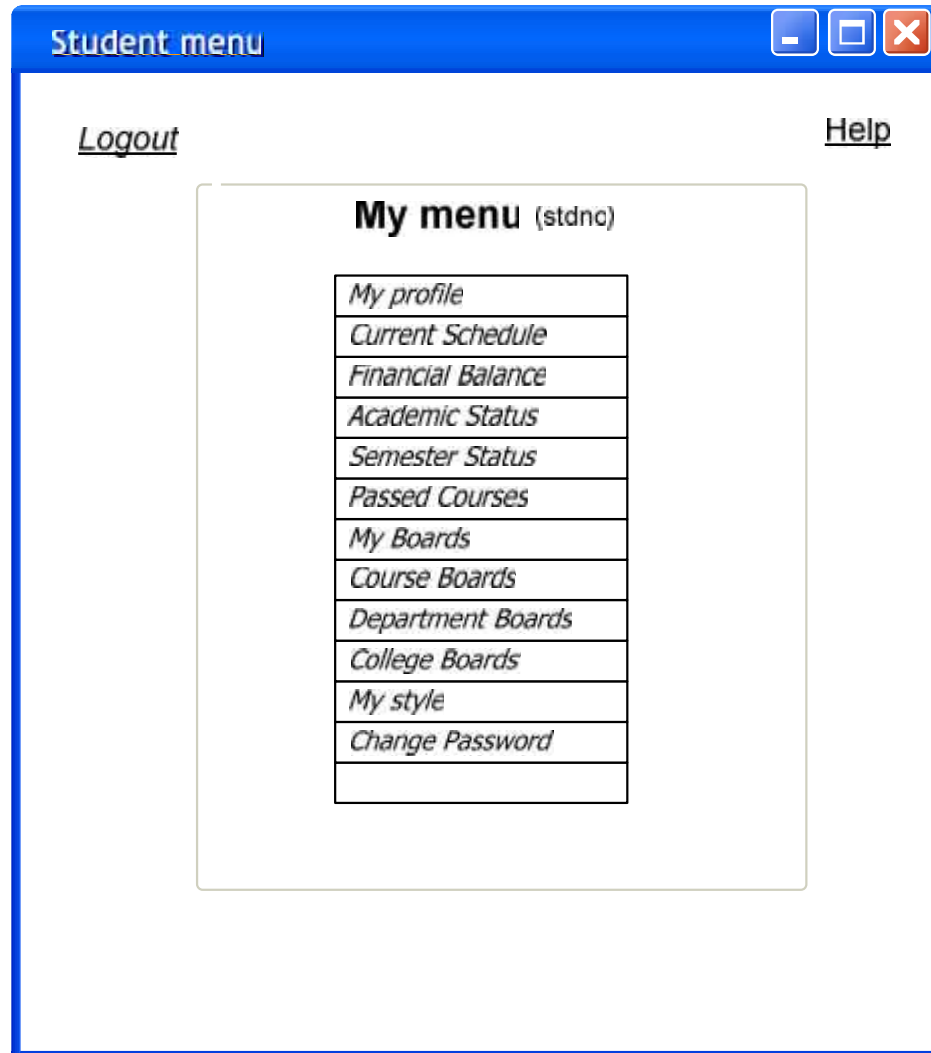


Figure (3.33) student menu.



➤ **Chang My Password:**

From this screen the student can change his password when he want.

The screenshot shows a web browser window titled "Change Password". The window has a blue header bar with the title "Change Password" and standard window control buttons (minimize, maximize, close). The main content area is white and contains the following elements:

- A title "Change Password" centered at the top of the form area.
- A form container with three input fields:
 - "Old Password" with an adjacent text input box.
 - "New Password" with an adjacent text input box.
 - "Confirm it" with an adjacent text input box.
- Three buttons at the bottom of the form: "Update", "Back", and "Home".

Figure (3.34) change password for student..



b. Student Output design :

➤ My profile :

This screen shows some of student information such as his name, address, etc... .

My Profile [stdno]

[Logout](#) [My Menu](#) [Help](#)

My Name

First Name: **Family Name :**

My Address

Address: **Telephone no:**

My Academic Inf.

Academic no: **Science Degree:** **Major :** **Outline:**

Figure (3.35) Student profile.



➤ Current Schedule :

This screen display the current schedule for enrolled courses for the student.

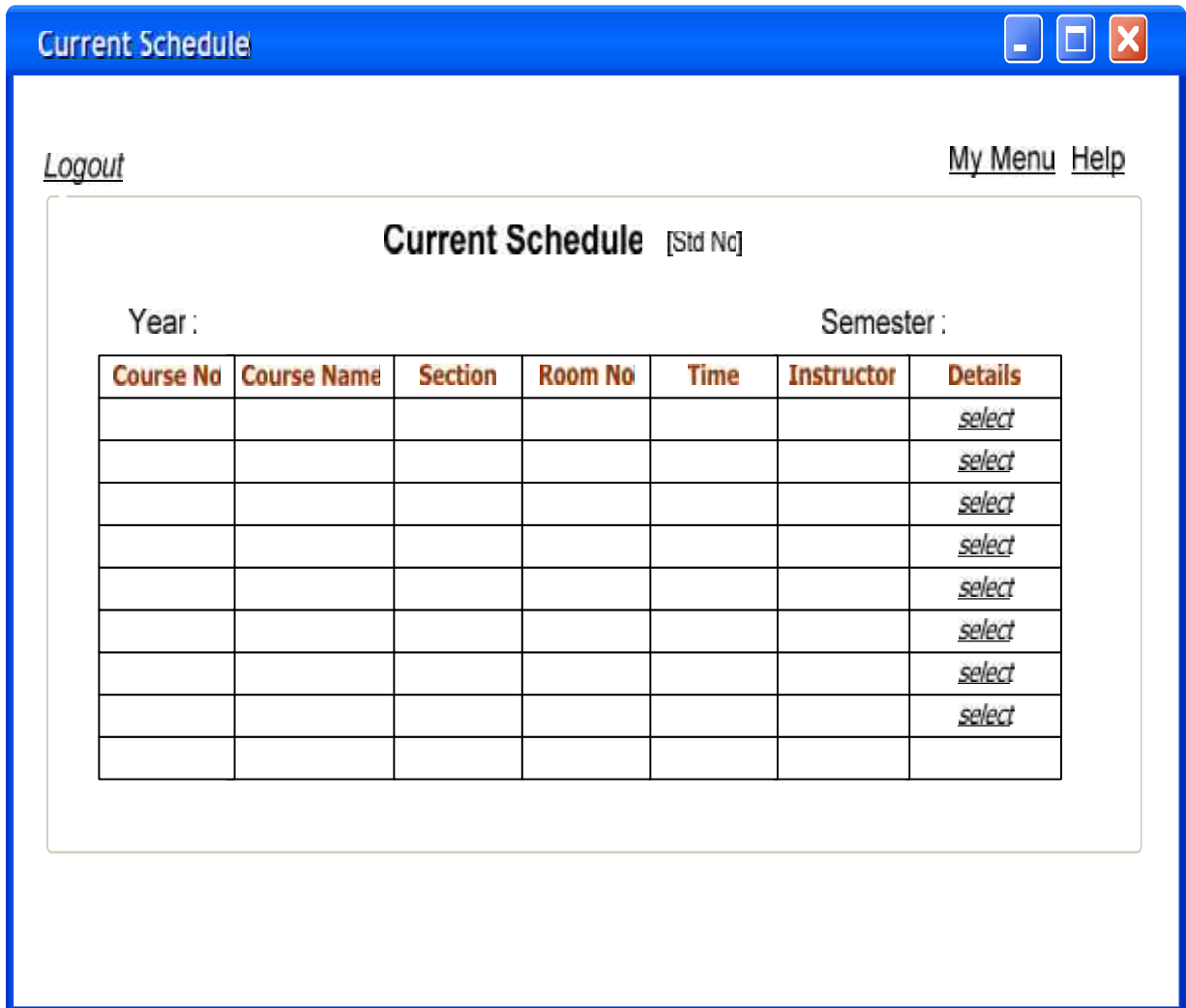


Figure (3.36) student current schedule..



➤ Financial Balance :

this screen shows the financial balance after calculate all financial aids, debits, credits for the logged student.

Financial balance

[Logout](#) [My Menu](#) [Help](#)

Financial balance [Std.No]

Year : Semester :

Financial Aids	Balance	Semester Balance	Total Balance
assistance Scholarship Other	Credit Debit	New	Deficit / surplus
<input type="text"/> <input type="text"/> <input type="text"/>	<input type="text"/> <input type="text"/>	<input type="text"/>	<input type="text"/>

Figure (3.37) student financial balance..



➤ Academic Status :

This screen displays the student academic status information such as accumulative average, major average, passed coursesetc.

Academic status

[Logout](#) [My Menu](#) [Help](#)

Academic Status

Year : _____ Semester : _____

First Name _____ Birth Place _____

Student No _____ Degree _____

Major _____

Department Name : _____ College Name : _____

Tawjibi

Tawjibi Average : _____ Tawjibi Branch : _____

Community Work Hours

Passed : _____ Remains : _____

Average

Accumulate _____ Major _____

Academic Alert: Dismiss :

_____ _____

Delay : Level : Regular : Graduated :

_____ _____ _____ _____

Figure (3.38) student academic status.



➤ Semester Status :

This screen shows only the academic status for the current semester.

Logout My Menu Help

Semester Status [stdno]

Year : Semester :

Course No	Course Name	Mark

Credit Hours :
Registered : Passed :

Honored : Semester Avg:

Figure (3.39) student semester status.



➤ Passed Courses :

From this screen the student can see all his passed courses and can filter them by choose from three available filtering methods.

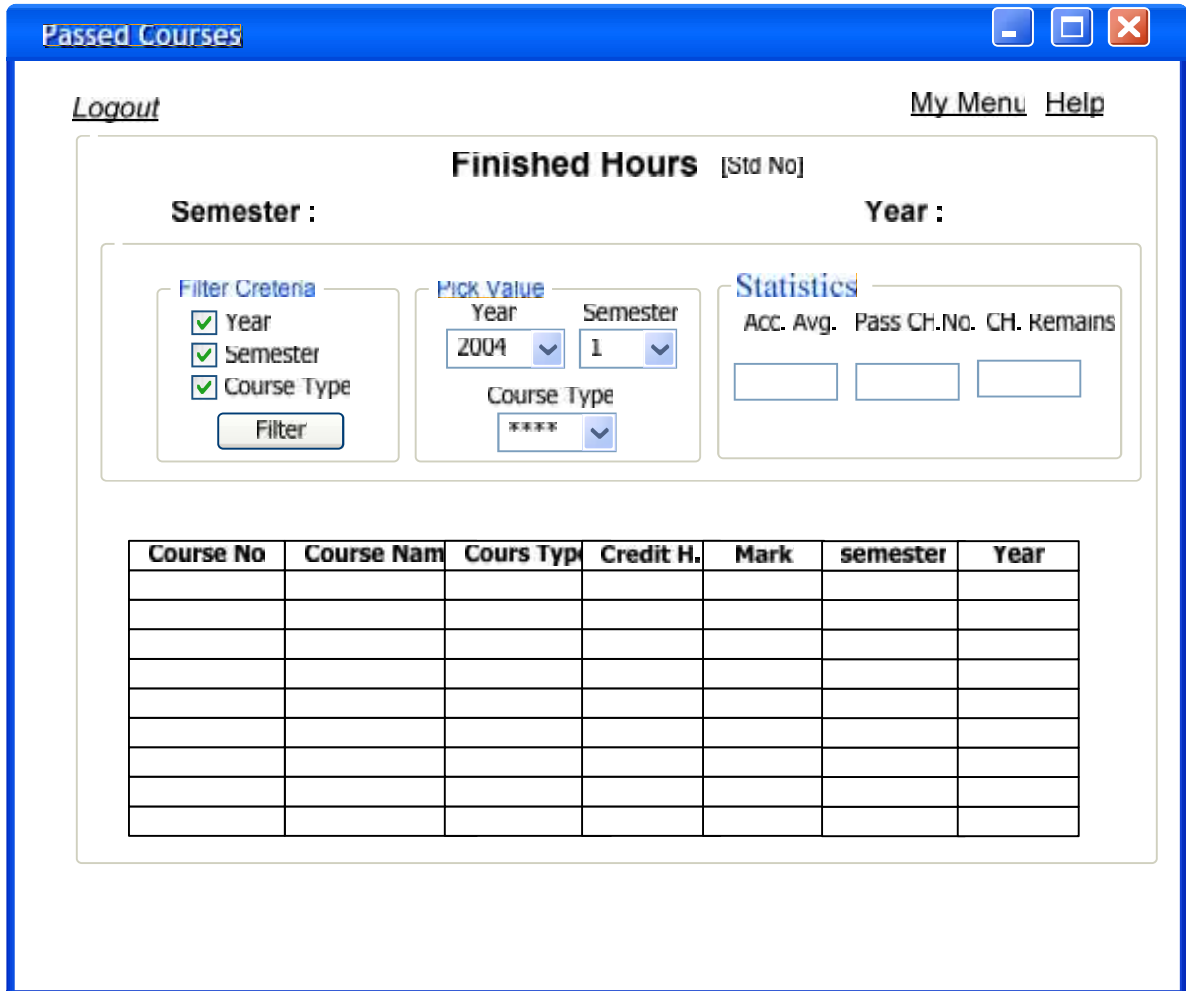


Figure (3.40) student passed courses.



➤ My Boards:

This screen shows the student all boards sent to him only.

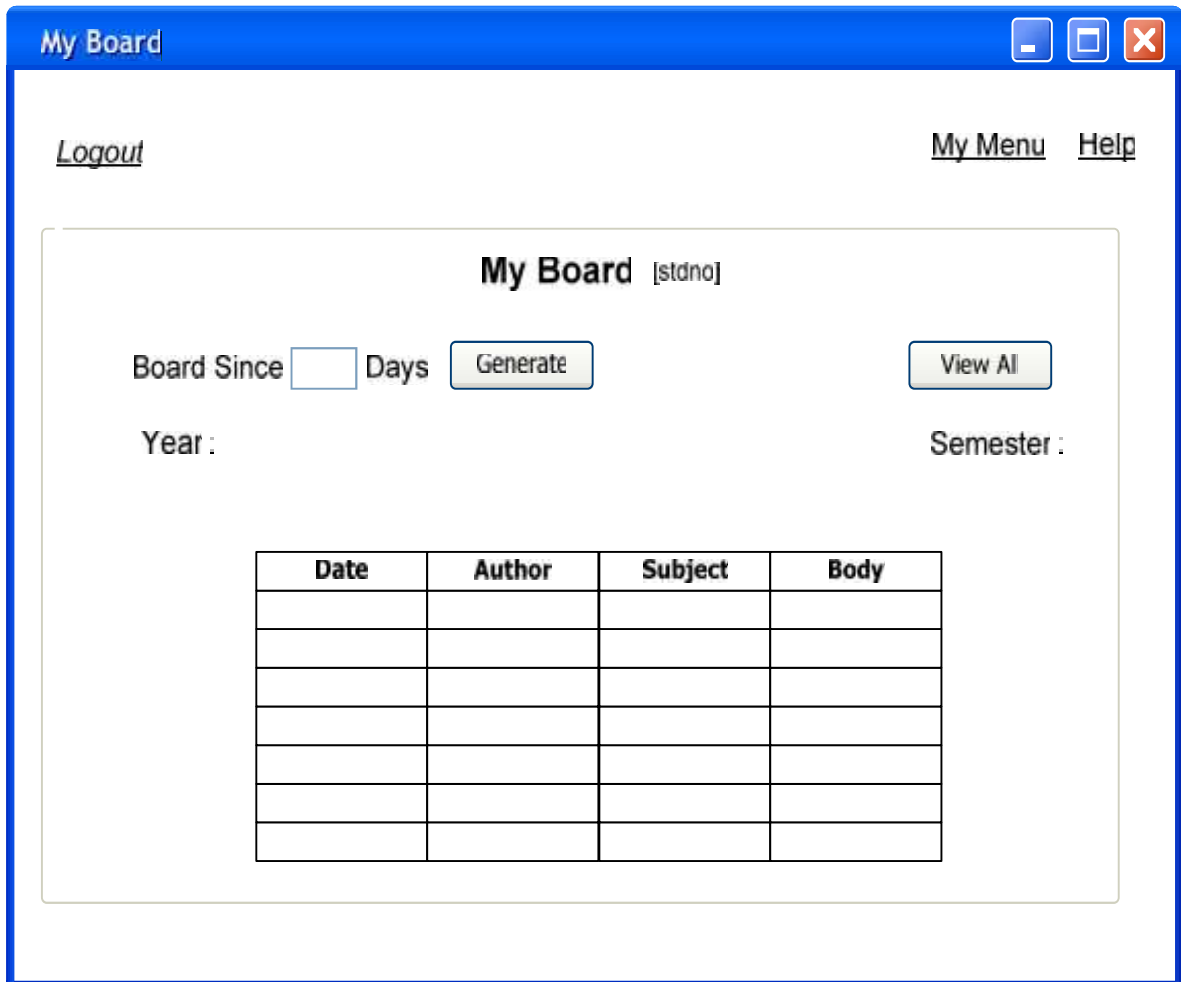


Figure (3.41) student boards.



➤ Course Boards

This screen shows boards for each course, so every student study this course can read it.

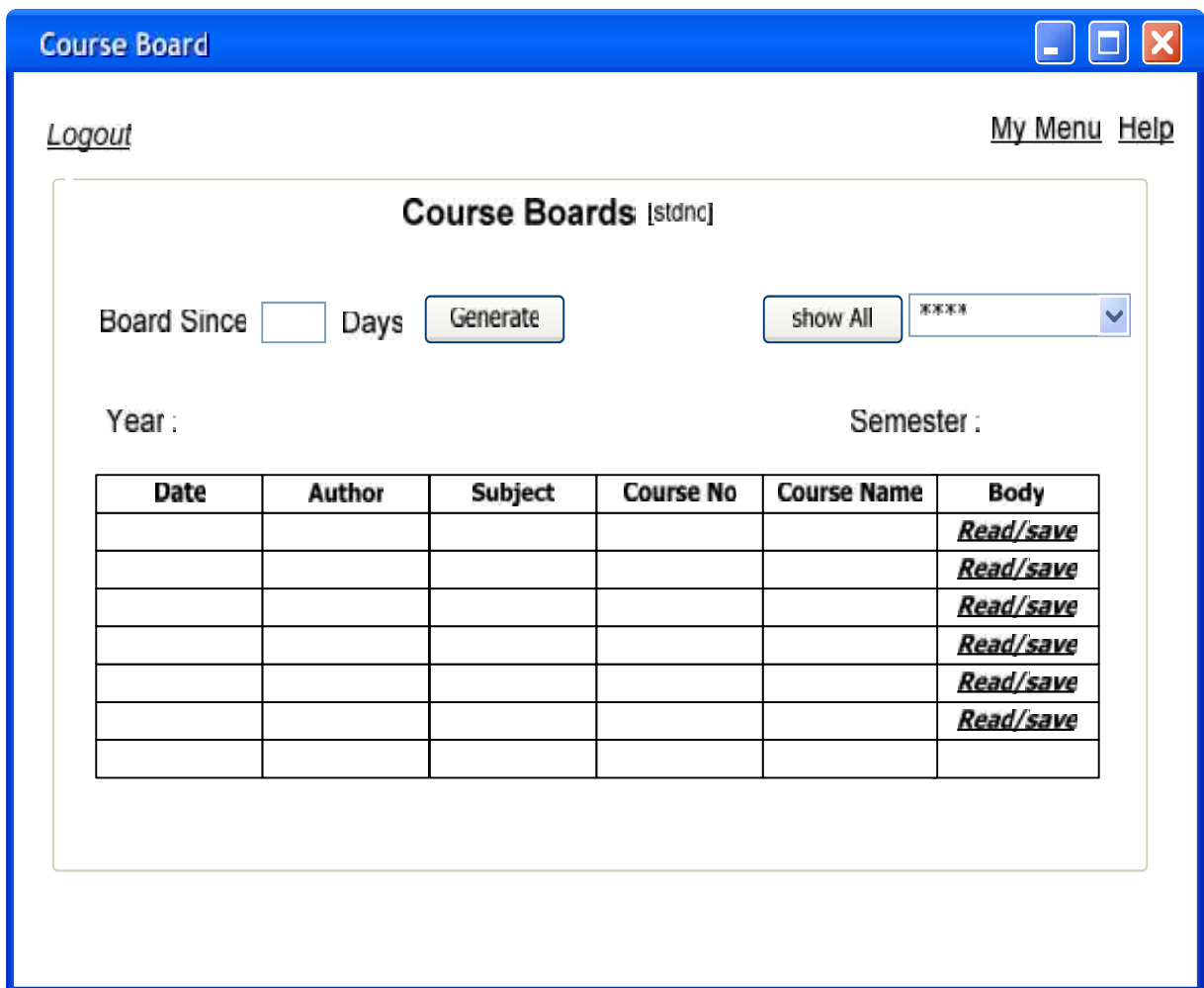


Figure (3.42) course boards.



- Department Boards :
here the student can read boards specified for his department, all students belong to this department can read this type of boards.

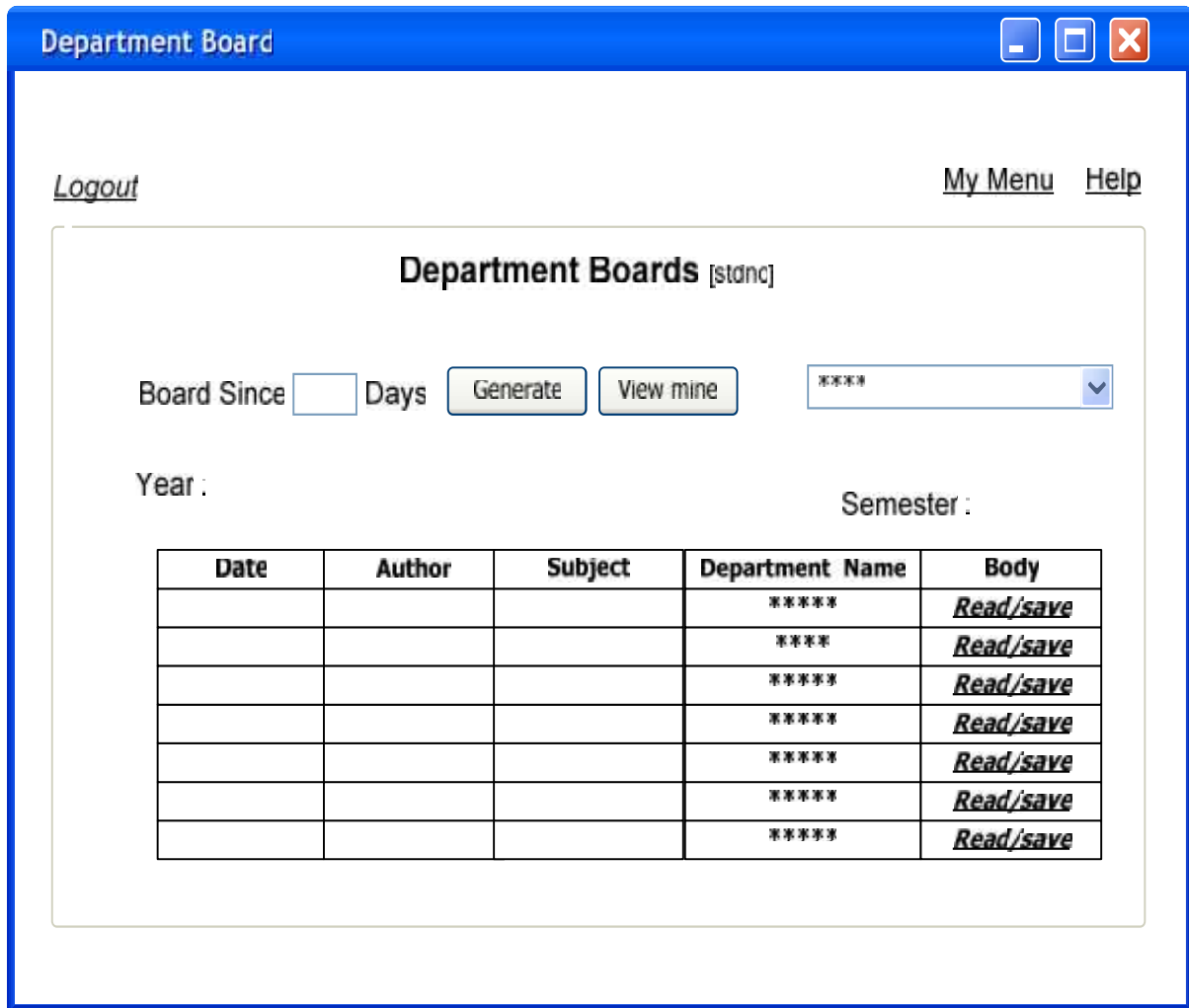


Figure (3.43) Department boards.



➤ College Boards :

This screen displays all boards specified for each college, so only students who belong to that college can see it.

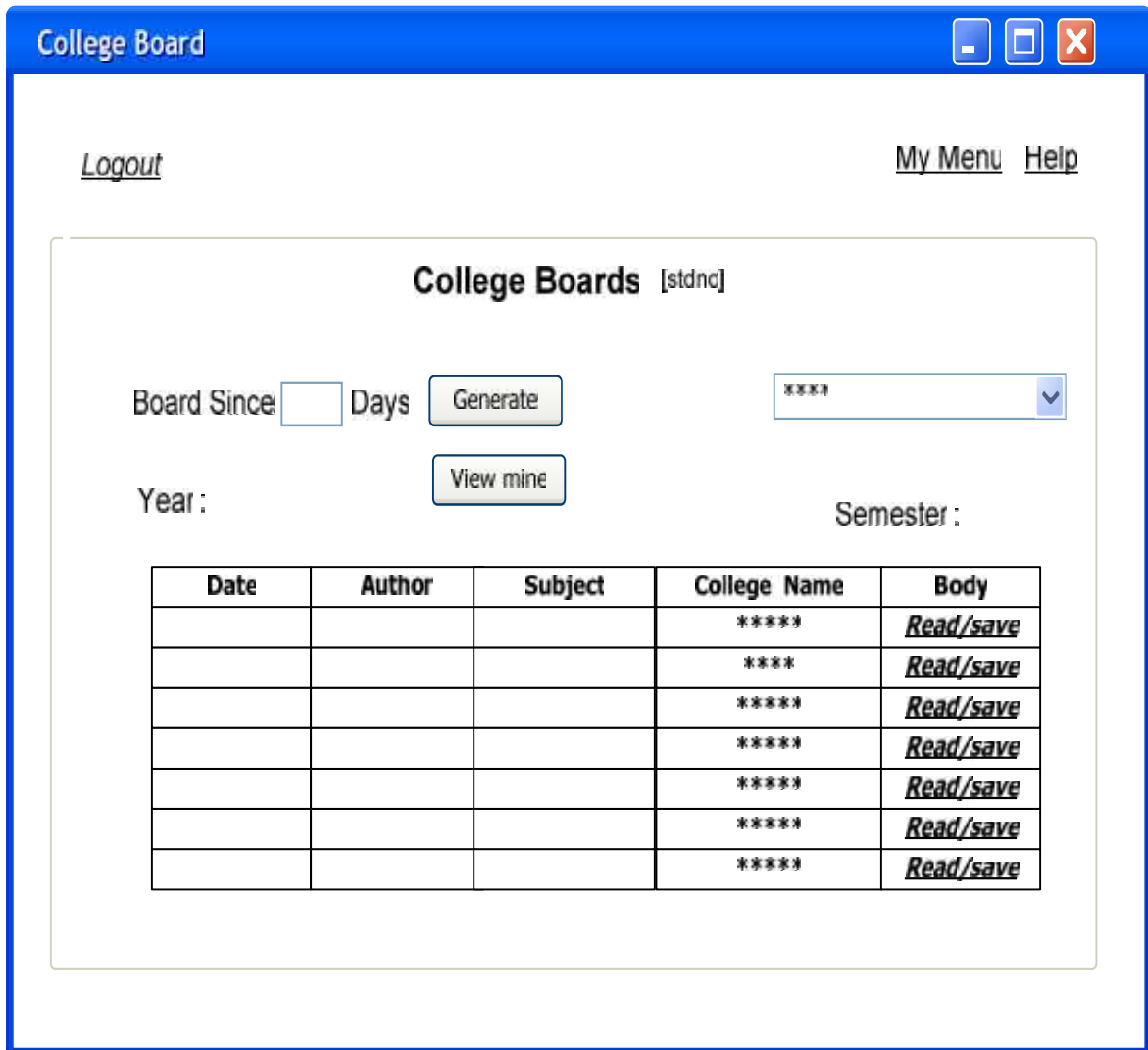


Figure (3.44) College boards.



B. Administrator interface design :

a.Administrator Input / output design:

➤ Login administrator :

this screen enables the administrator to enter to his account by insert his username and password.

Login

Please Insert Your ID And Password

Admin. ID

Password

Login

[Forgot My Password?](#)

Figure (3.45) Administrator login screen.



➤ Insert boards :

this screen enables the administrator to insert boards to the database and can determine the board destination such as college boards, course boards.... Etc.

The screenshot shows a web application window titled "Insert Boards". Inside the window, there is a form titled "Insert New Boards Form". The form is divided into several sections:

- Semester :** and **Year :** labels for identifying the board's time period.
- Board Data:** A section containing input fields for **Date:**, **Due To:**, **Title:**, **Board no:**, and **Author:**.
- Board body:** A section with two checked checkboxes: **Exit** and **New**.
- Select board type:** A list of radio buttons for selecting the board type: **Bulletin board**, **College board**, **Department board**, **Course board**, and **Student board**.
- Board text:** A large text area for entering the board's content, with a **Browse** button below it for file selection.
- Upload the board:** A button at the bottom right of the form to submit the data.

Figure (3.46) Insert new boards.



- Delete boards :
Here the administrator can delete boards depending on some criteria such as expired boards.

The screenshot shows a web application window titled "Delete boards". The main content area is titled "Delete boards Form". It contains a section "Delete boards According" with three radio buttons: "Expired boards", "Board name", and "Board type". Below these is a dropdown menu. To the right is a "Get Boards" button. Further right is a "Target Board" dropdown menu, a "Boards to delete" dropdown menu, and four navigation buttons: ">", "<", ">>", and "<<". Below these is an "Empty" button and a "Delete" button at the bottom right.

Figure (3.47) Delete boards.



➤ Update boards :

This screen enable the administrator to edit and update the board body or deadline or both.

The screenshot shows a web application window titled "Update boards". The main content area is titled "Update boards Form". It contains several sections:

- Available Boards Titles:** A dropdown menu.
- Update:** Two radio buttons: "Board & deadline" and "Deadline only".
- Board Type:** Two radio buttons: "Exit" and "New".
- Board Deadline:** A text input field and a "Change" button.
- Text Area:** A large text area with a "Browse" button.
- Update:** A button at the bottom right.

Figure (3.48) Update boards.



- insert mark :
the administrator can insert student's marks from this screen by select the course number and insert marks for all students studying this course.

Student no	Course no	Course name	Mark	Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit

Figure (3.49) Insert Marks.



➤ Insert college :

The administrator can insert new college to the portal database from here, and he can insert some information about the college.

Insert College

Insert college form

Description type

Exit

New one

College id :

College Name :

College description

Figure (3.50) Insert new college.



➤ Insert Major :

The administrator can use this screen to insert new major to some department in the university and he can insert major no, major description ...etc.

The screenshot shows a web browser window titled "Insert major". The main content area is titled "Insert major form" and contains the following elements:

- Description type:** Two radio buttons labeled "Exit" and "New one".
- Major no.:** A text input field.
- Major Name:** A text input field.
- Credit H.:** A text input field.
- Outline year:** A text input field.
- Major description/outline:** A large text area with a vertical scrollbar, a horizontal scrollbar, and a "Browse" button below it.
- Department name:** A dropdown menu.
- Upload major:** A button located at the bottom right of the form.

Figure (3.51) Insert new major.



➤ Questioner analysis

This screen used by administrator to display questionnaire analysis and he can upload new question for voting.

The screenshot shows a web application window titled "Financial balance". Inside the window, there is a section titled "Questionnaire Form". This section contains three summary boxes:

- Result static:** A box with three rows: "Yes" followed by four dashes, "No" followed by three dashes, and "Maybe" followed by four dashes.
- Result percentage:** A box with three rows: "Yes" followed by four dashes and a percent sign, "No" followed by three dashes and a percent sign, and "Maybe" followed by four dashes and a percent sign.
- Total Balance:** A box with three rows: "Yes" followed by three dashes, "No" followed by three dashes, and "Maybe" followed by three dashes.

Below these boxes, there is a "Questioner text" label, a "Total Sharing" input field, and two buttons labeled "Upload" and "New". At the bottom of the form is a large text area with a vertical scrollbar on the right side.

Figure (3.52) Questionnaire analysis.



- control panel :
this screen enable the administrator to gain control over the portal by activate or deactivate some controls in student or instructor accounts.

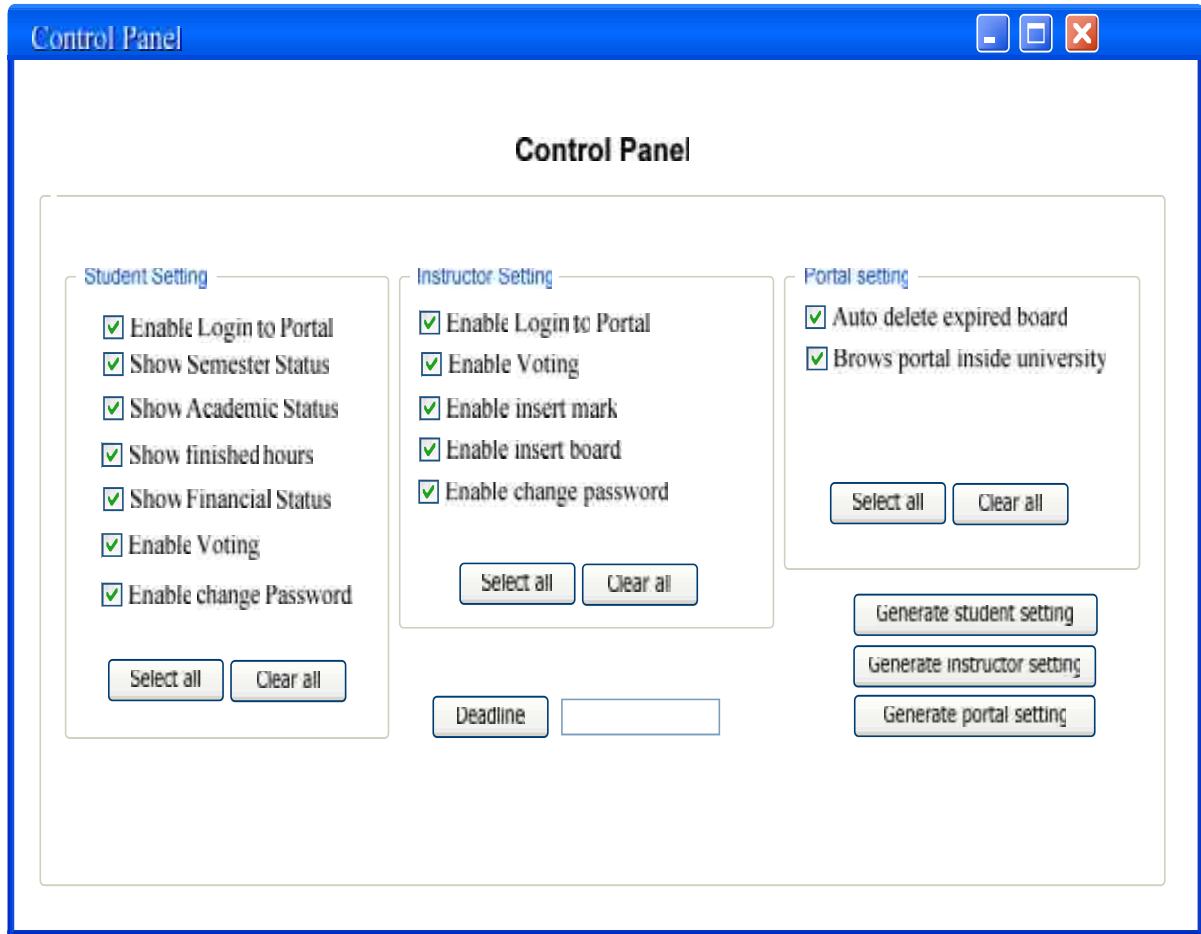


Figure (3.53) Control panel.



➤ User accounts :

Here the administrator displays all students accounts and so for instructors.

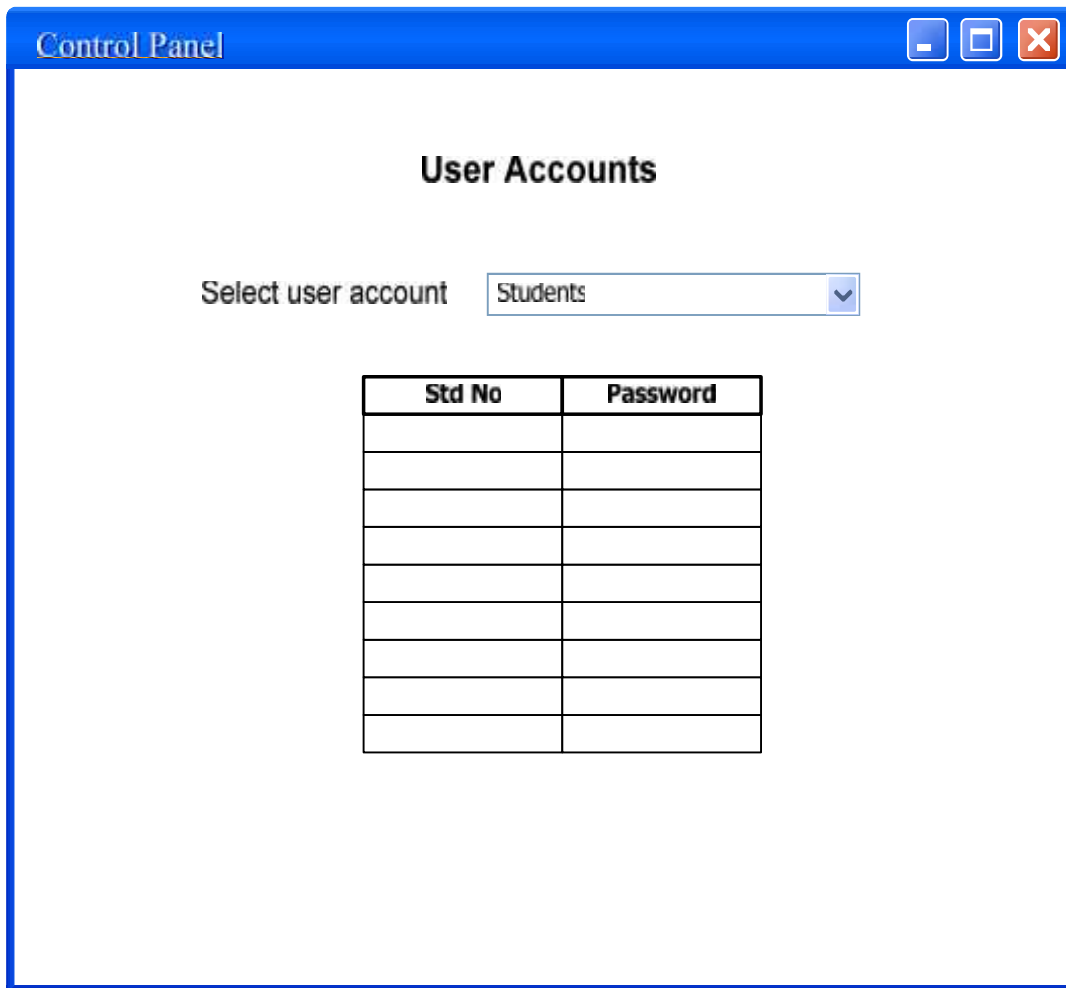


Figure (3.54) User accounts.



- change password :
this screen enables administrator to change his password when he want.

Change Password

Change Password

Old Password

New Password

Confirm it

Update password

Figure (3.55) Change password.



C. Instructor interface design :

a. Instructor Input / output design:

➤ Insert board :

this screen enable the instructor to insert boards only for his students and courses.

The screenshot shows a window titled "Insert Boards" with a blue title bar. Inside the window, the form is titled "Insert New Boards Form". At the top, there are labels for "Semester :" and "Year :". Below this, the form is divided into several sections:

- Board Data:** A section containing input fields for "Date.", "Due Id.", "Title.", and "Board no.", followed by an "Author." field.
- Board body:** A section containing two checked checkboxes: "Exit" and "New". Below these is a "Select board type" section with two radio buttons: "Course board" and "Student board".
- Board text:** A section containing a large text area for input, a "Browse" button, and an "Upload the board" button.

Figure (3.56) Insert new board by instructor.



- insert mark :
this screen enables the instructor to add marks for courses taught by him.

Student no	Course no	Course name	Mark	Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit

Figure (3.57) Insert marks by instructor.



➤ My regular student :

This screen used by instructor to list all his regular students.

Course no	Section	Student no	Student name

Figure (3.58) Instructor regular students.



D. Public interface design :

a. Public Input / output design:

➤ Search :

this screen contains sequential search engine restricted on the Portal database, this search applied for courses, colleges, departments ... etc.

The screenshot shows a web-based search interface. At the top, there is a blue header bar with the word "Search" in white. Below the header, the main content area is enclosed in a light gray border. On the left side of this area, the text "Search For:" is displayed. To its right is a dropdown menu with "Course" selected and a small downward arrow. Below the dropdown is a rectangular text input field. At the bottom center of the input area is a button labeled "Search".

Figure (3.59) Search engine.



3.4 Database design

3.4.1 Database Model



3.4.2 Database tables:

1. Students:

Field	Data Type	Required	Key	References	Description
Std_no	Numeric(9)	Yes	PK		Student Number
Std_fname	Varchar(50)	Yes			Student first name
Std_sname	Varchar(50)	Yes			Student second name
Std_tname	Varchar(50)	Yes			Student third name
Std_lname	Varchar(50)	Yes			Student family name
Birth_d	Nvarchar(50)	No			Student birth date
Birth_p	Varchar(50)	No			Student birth place
nationality	Varchar(50)	No			Student nationality
Social_s	Varchar(50)	No			Student social status
Std_id	Numeric(9)	Yes			Student identity number
Tawj_avg	Float(8)	Yes			Student taw. Avg.
Sex	Varchar(50)	No			Student gender
Address	Varchar(50)	Yes			Student address
Tel_no	Numeric(9)	Yes			Student tel. no.
Mjr_no	Numeric(9)	Yes	FK	Majors(mjr_no)	Student major number
Ac_no	Numeric(9)	Yes	FK	Academic_status(ac_no)	Student Academic status number
Ta_no	Numeric(9)	Yes	FK	Taw_branch(ta_no)	Student tawjihi number
pic	Nvarchar(50)	Yes			Student picture

Table (3.1) Students Table.

2. *Academic status:*

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Ac_no	Numeric (9)	Yes			Academic number
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Warning	Varchar (50)	No			Number of warnings
dismiss	Varchar (50)	No			Dismissed or not
Deelay	Varchar (50)	No			Delayed or not
Std_level	Numeric(9)	Yes			Student level
Regular	Varchar (50)	No			Regular or not
Graduated	Varchar (50)	No			Graduated or not
Hours_reg	Numeric (9)	Yes			Number of registered hours
Cwh_p	Numeric (9)	No			Community work hours
Sem_no	Nvarchar (50)	Yes			Semester number
Hours_p	Numeric (9)	No			Number of passed hours
Ac_year	Numeric (9)	Yes			Academic year
St_no	Numeric (9)	Yes	FK	Study_system (st_no)	
Std_avg	Float (8)	No			
Hours_p	Numeric (9)	Yes			

Table (3.2) Academic status Table.

3. *Boards:*

Field	Data Type	Required	key	References	Description
brd_no	Numeric (9)	Yes	PK		Board number
Brd_date	Datetime(9)	Yes			Board issue date
Due_to	Datetime(9)	Yes			Board due to date
Bodyt	Varchar (50)	No			Text body
Bodyb	Varchar (50)	No			Attachment body
Auther	Varchar (50)	No			Auther name
Subject	Varchar (50)	No			Title of the board

Table (3.3) Boards Table.

4. *Colleges Boards:*

Field	Data Type	Required	Key	References	Description
Col_no	Numeric (9)	Yes	PK		College number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.4) College boards Table.

5. *Departments boards:*

Field	Data type	Required	Key	References	Description
Dpt_no	Numeric (9)	Yes	PK		Department number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.5) Departments boards Table.

6. *Courses Boards:*

Field	Data type	Required	Key	References	Description
crs_no	Numeric (9)	Yes	PK		Course number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.6) Courses boards Table.

7. *Students Boards:*

Field	Data type	Required	Key	References	Description
std_no	Numeric (9)	Yes	PK		Student number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.7) Students boards Table.

8. *Colleges:*

Field	Data type	Required	Key	References	Description
Col_no	Numeric (9)	Yes	PK		College number
Coll_name	Varchar (50)	Yes			College name
descr	Nvarchar (500)	No			College description

Table (3.8) Colleges boards Table.

9. *Courses:*

Field	Data type	Required	Key	References	Description
crs_no	Numeric (9)	Yes	PK		Course number
Crs_name	Varchar (50)	Yes			Course name
Crd_hours	Numeric (9)	Yes			Course credit hours
descr	Nvarchar (500)	No			Course description

Table (3.9) Courses Table.

10. *Courses_type:*

Field	Data type	Required	Key	References	Description
Co_no	Numeric (9)	Yes	PK		Course number
Co_t	Varchar (50)	Yes			Course type

Table (3.10) Course_Type Table.

*11. Passed courses:*

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_Courses(crs_no)	Course number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes	PK		Academic year
Class_no	Numeric (9)	Yes			Section number
Mark	Float (8)	No			Course mark
Co_no	Numeric (9)	Yes	FK	Courses_type(co_no)	Course number

Table (3.11) Passed courses Table.

12. Offered_courses:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	courses(crs_no)	Course number
Class_no	Numeric (9)	Yes	PK		Section number
Time	Nvarchar (50)	Yes			Lecture time
Room_no	Numeric (9)	No			Room number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes			Academic year

Table (3.12) Offered courses Table.

*13. Major courses:*

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	Courses(crs_no)	Course number
Mjr_no	Numeric (9)	Yes	PK,FK	Majors(mjr_no)	Major number

Table (3.13) Major courses Table.

14. Departments:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Dpt_no	Numeric (9)	Yes	PK		Department number
Dpt_name	Varchar (50)	Yes			Department name
Descr	Nvarchar (500)	No			Department description
Col_no	Numeric (9)	Yes	FK	Colleges(col_no)	College number
Cwh_r	Numeric (9)	No	PK		Number of required community work hours
Ac_no	Numeric (9)	Yes	FK	Academic status(ac_no)	Academic year

Table (3.14) Departments Table.

15. Employee classes:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK,FK	Employee(emp_no)	Employee number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_courses(crs_no)	Course number
Sem_no	Nvarchar (50)	No			Semester number
Class_no	Numeric (9)	Yes	Pk		Section number

Table (3.15) Employee classes Table.

*16. Employee:*

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK		Employee number
Emp_name	Varchar (50)	Yes			Employee name
Dpt_no	Numeric (9)	No	FK	Departments(dpt_no)	Department number
St_no	Numeric (9)		FK	Study_system(st_no)	Study system (scientific degree)
Id_no	Numeric (9)	Yes			Identity number

Table (3.16) Employee Table.

*17. Financial status:*

Field	Data type	Required	Key	References	Description
Fin_no	Numeric (9)	Yes			Financial account number
assistant	Float (8)	No			Assistant value
scholarship	Float (8)	No			Scholarship value
others	Float (8)	No			Other financial aids
credit	Float (8)	Yes			Student surplus balance
debit	Float (8)	Yes			Student deficit balance
Std_no	Numeric (9)	Yes	PK		Academic year

Table (3.17) Financial status Table.

18. Administrator login:

Field	Data type	Required	Key	References	Description
username	Nvarchar (50)	Yes	PK		Administrator username (id)
pwd	Nvarchar (50)	Yes			Administrator password

Table (3.17) Administrator login Table.

19. Students login:

Field	Data type	Required	Key	References	Description
Std_no	Numeric (9)	Yes	PK		Student number (id)
pwd	Varchar (50)	Yes			Student password

Table (3.18) Students login Table.



20. Instructor login:

Field	Data type	Required	Key	References	Description
Emp_no	Numeric (9)	Yes	PK		Employee number
pwd	Nvarchar (50)	Yes			Employee password

Table (3.19) Instructor login Table.

21. Majors:

Field	Data type	Required	Key	References	Description
Mjr_no	Numeric (9)	Yes	PK		Major number
Mjr_name	Varchar (50)	Yes			Major name
Hours	Numeric (9)	Yes			Major credit hours number
Descr	Nvarchar (500)	No			Major description
Dpt_no	Numeric (9)	Yes	FK	Departments(dpt_no)	Department number
outline	Numeric (9)	No			Major outline

Table (3.20) Majors Table.

22. Projects:

Field	Data type	Required	Key	References	Description
Std_no	Numeric (9)	Yes	PK		student number
Proj_no	Int (4)	Yes			Project number
Proj_name	Varchar (50)	Yes			Project name
Proj_body	Nvarchar (500)	Yes			Project body
Emp_no	Numeric (9)	Yes	FK	employee(emp_no)	Employee number

Table (3.21) Projects Table.

23. *Setting:*

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Set_no	Int (4)	Yes	PK		Setting account number
Set_s	Int (4)	Yes			Setting status value
Set_date	Datetime (8)	No			Setting deadline date

Table (3.22) Setting Table.

24. *Student style:*

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		student number
Main_header	Varchar (50)	Yes			Main header image path
Name_header	Varchar (50)	Yes			Name header image path
Title_header	Varchar (50)	Yes			Title header image path
Box	Varchar (50)	Yes			Box image path
Center_box	Varchar (50)	Yes			Center box image path
Box_header	Varchar (50)	Yes			Box header image path
Centerbox_header	Varchar (50)	Yes			Center box header image path
logout	Varchar (50)	Yes			Logout image path

Table (3.23) Student style Table.



25. Study system:

Field	Data type	Required	Key	References	Description
St_no	Numeric (9)	Yes	PK		Study system number
St_name	Varchar (50)	Yes			Study system name

Table (3.24) Study system Table.

26. tawjihi branch:

Field	Data type	Required	Key	References	Description
Ta_no	Numeric (9)	Yes	PK		Taw. number
Ta_name	Varchar (50)	Yes			Taw. branch

Table (3.25) Tawjihi branch Table.

27. Vote:

Field	Data type	Required	Key	References	Description
Vote_no	Numeric (9)	Yes	PK		Vote number
Vote_body	Varchar (500)	Yes			Vote body text

Table (3.26) Vote Table.

28. Vote result:

Field	Data type	Required	Key	References	Description
Vote_no	Numeric (9)	Yes	PK,FK	Vote(vote_no)	Vote number
V_yes	Int (4)	No			Vote yes value (1)
V_no	Int (4)	No			Vote no value (1)
V_maybe	Int (4)	No			Vote maybe value (1)
who	Numeric (9)	Yes	PK	employee(emp_no)	Userid for who voted

Table (3.27) Vote result Table.



3.5 Test Plan:

Here we describe briefly the methodology that we have adapted to test the system, steps that will be followed in the system testing are described bellow:

Testing steps:

1- Unit and Module testing:

We will use the white and black test Box testing to ensure that each function or module will operate as expected, by inserting valid and invalid inputs to show how the system will handle it.

2- Sub-system testing:

In this stage we will test each sub-system individually to ensure that each sub-system is operates as expected and meet its requirements.

3- Integration testing:

The integration of all sub-systems will be tested so that to ensure that the subsystems work together properly as expected, and meets its requirements.

4- System testing:

The system with all subsystems and functions will be tested to ensure that it functions properly, it meets its specification, and show if there defects throw system running.



3.6 Programming Language and Coding

There are many languages that can be used to develop a system such ours, but the most two effective languages are JAVA language and ASP.NET using Microsoft Visual Studio .NET as the development tool, here we describe the advantages of each, and why our selection was on the ASP.NET (we describe the ASP.NET technology in details in chapter 1) :

1- Device Independent:

Both Java and ASP.Net are a device independent languages, this means that any user can open the page that written using theses two languages from any explorer without any additional components or drivers, because he will just receive an HTML code.

2- Security

Java and ASP.Net have a high level of security during transmitting data; they provide us with many algorithms and techniques.

In the ASP.NET there are a build in valuator that ensure the user's input before any generation on the server, so if there any unusual input the webpage it self will not return to the server.

In Java the programmer must do all algorithms and validations manually.

3- Server side

One of the most powerful advantages of the ASP.NET technology is that it do not need to make any efforts on the client side, all operations and functions will work on the server.



4.1 Introduction

In this chapter we will describe the process of coding and implementation of the system and describe the environment that we use it in these processes.

The development of our software system to work properly to meet its predetermined requirements on the internet architecture needs a set of certain software and hardware products found in a platform configured to be suitable for the deployment process.

This system is an internet application that depends on a number of technologies that need to be installed, maintained, and updated, continuously. There exist a large number of software development packages that belong to different companies such as Microsoft.

This system is built basically on a group of Microsoft technologies such as the MS SQL server 2000 and ASP.NET 2003, IIS, Microsoft internet explorer 6.0, and Microsoft windows XP.

We also included some other applications that serve for the user interface design such as Macromedia Flash MX, Adobe Photoshop 8.0, and PhotoImpact XL.

This chapter describes the packages of software and hardware used in the system for the operation phase and how they are installed and prepared for work, the database and web interfacing, and the user interface implementation.



4.2 Establishment of development environment

▪ *Hardware Environment*

For the system development we use the following hardware:

1. Three PC Pentium 4:
 - a. 2.40 GHz speed.
 - b. 256 MB RAM.
 - c. 40 GHz H.D.D
 - d. Monitor, mouse, and keyboard.
 - e. 52x MAX CD-ROM.
2. One printer.
3. Scanner.
4. Tow flash memory with 256 MB.

▪ *Software Environment*

For the system development environment we use the following software:

1. Microsoft Windows XP Professional edition with IIS web server and server extensions.
we use this platform or operating system because our system is built on Microsoft visual studio.NET technology , and this type of operating system is the best platform for this technology we can use.
2. Microsoft visual studio.NET 2003 with ADO.NET.



From this package we use ASP.NET technology which is a new powerful technology designed to create web applications, making complete error handling, and provide data access tools.

- Why we use ASP.NET technology?

We use it because of:

- Ease of use.
- Reduce development time.
- Multiple languages within a project.
- Integrated browser.
- Debugging support.
- Customizable interface.
- High compatibility with windows environment.
- High compatibility with database.

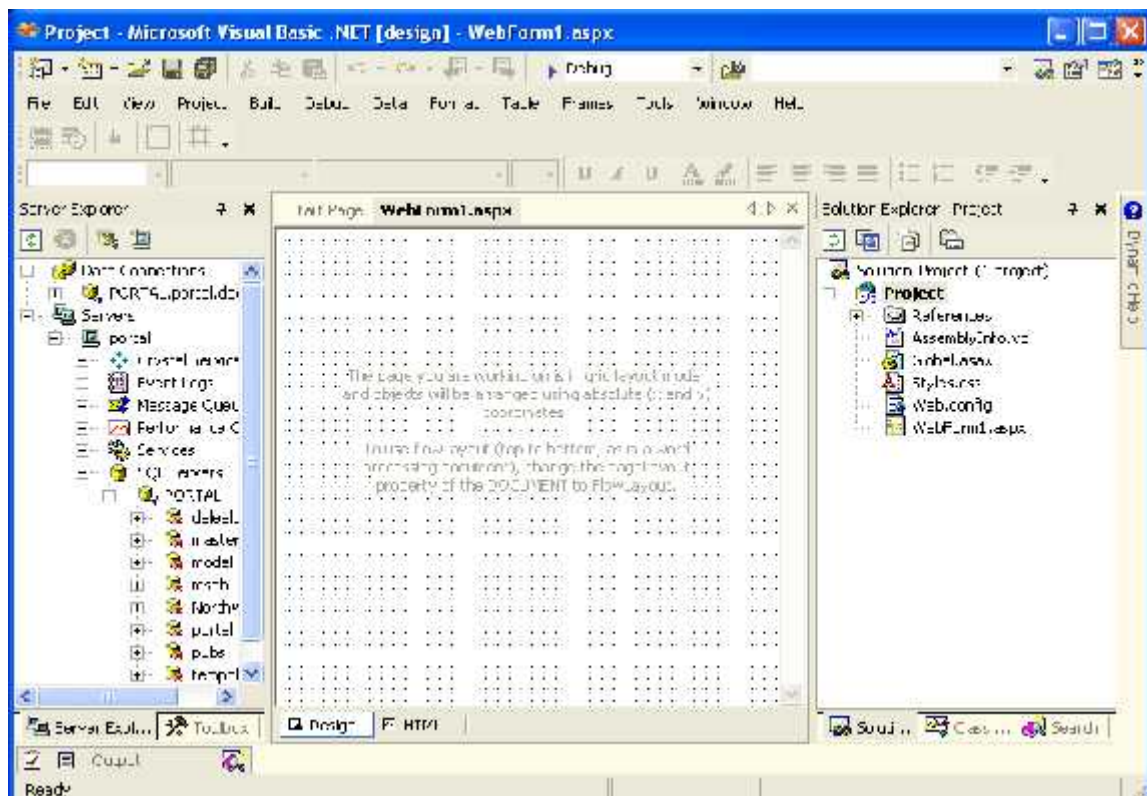


Figure (4.1) ASP.NET New Application Window.



To install visual studio.NET frame work, we need firstly to install a windows component which named IIS (Internet Information Services) on the windows by following these steps:

- ✓ Open control panel.
- ✓ Double click on add/remove programs icon.
- ✓ Click on add/remove windows components icon.
- ✓ Check the internet information system (IIS).
- ✓ Click on details button.
- ✓ Check options.
- ✓ Insert CD that labeled Windows XP Professional.
- ✓ Click next.
 - Setup will started to copy the required files to the hard disk.
- ✓ Click finish to close the windows components wizard.
- ✓ Close the add/remove programs window.

After IIS installation you can be able to install visual studio.NET program by using flexible integrated wizard.

3. Microsoft SQL server 2000.

An enterprise edition of the Microsoft SQL server 2000 is a good system to creating, accessing, and managing the database system of our application.

- Why we use SQL server 2000?
 - Ease of use.
 - High compatibility with ASP.NET.
 - High flexibility.
 - High security.
 - Ease of make a connection, query, and retrieve data.
 -

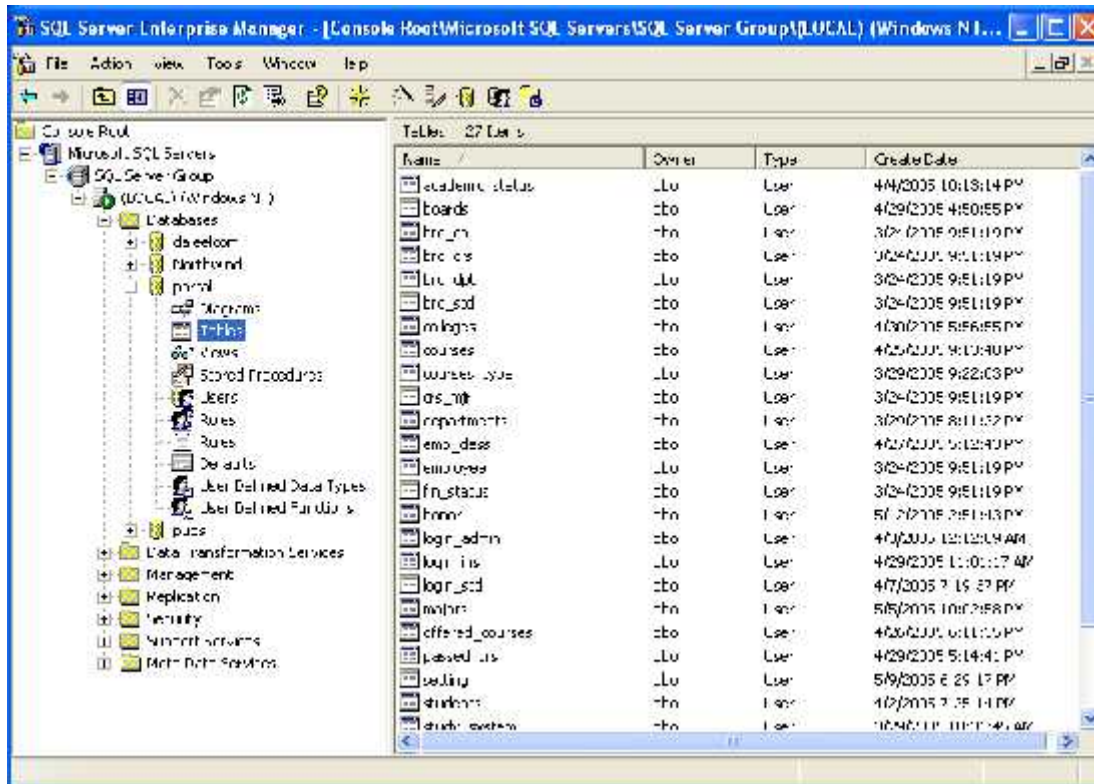


Figure (4.2) Enterprise Manager Window.

- ❖ In SQL server 2000 we can create new user or administrator and give him a specific privileges to access the database, in order to control the database resources.
- ❖ Also in SQL server 2000 we can build a stored procedures that we can use it from ASP.NET application by calling it, which provide a high security , ease of use, and high speed accessing data with database.
- ✓ *Configuration of SQL server 2000:*

When using .NET technology ADO.NET is used for database connections, ADO.net which is a new technology that is based on the usefulness of Microsoft ActiveX Data Object (ADO), however. It is a new technology for manipulating data, it contains numerous improvements over the previous version of ADO, and it is greatly simplifies the process of connecting a web application to a database.



Unlike ADO, ADO .NET is specifically designed for data connections located in a disconnected environment, so it the best choice when developing and implementing internet based applications.

The major point here is that why SQL server 2000? And how to configure it after installation?

As we described in this section, that the connectivity and manipulation of the database in the .Net is provided by the ADO.Net technology, now because SQL server 2000 is integrated with the .NET technology tools, it certainly should be compatible for the access by ADO.NET. However, this integration appears to be more efficient and secure especially in such systems (web applications).

By configuring the SQL server 2000 to the windows only authentication mode, which is the preferred method to use when connecting a web application to SQL server 2000 DBMS, this method does not need any user name or password to be transferred back and forth between servers, only the confirmation that the user has been authenticated by trusted sources is required to process the database request.

The last configuration of the SQL server 2000 is to work effectively and ensuring that the integration more comfortable is to add a new account in the login group of the SQL server. This account (ASPNET) is created by the .NET framework and it should be added to the login group of the SQL server. Figure (4.3) shows how to add this account to the logins group of the SQL Server.

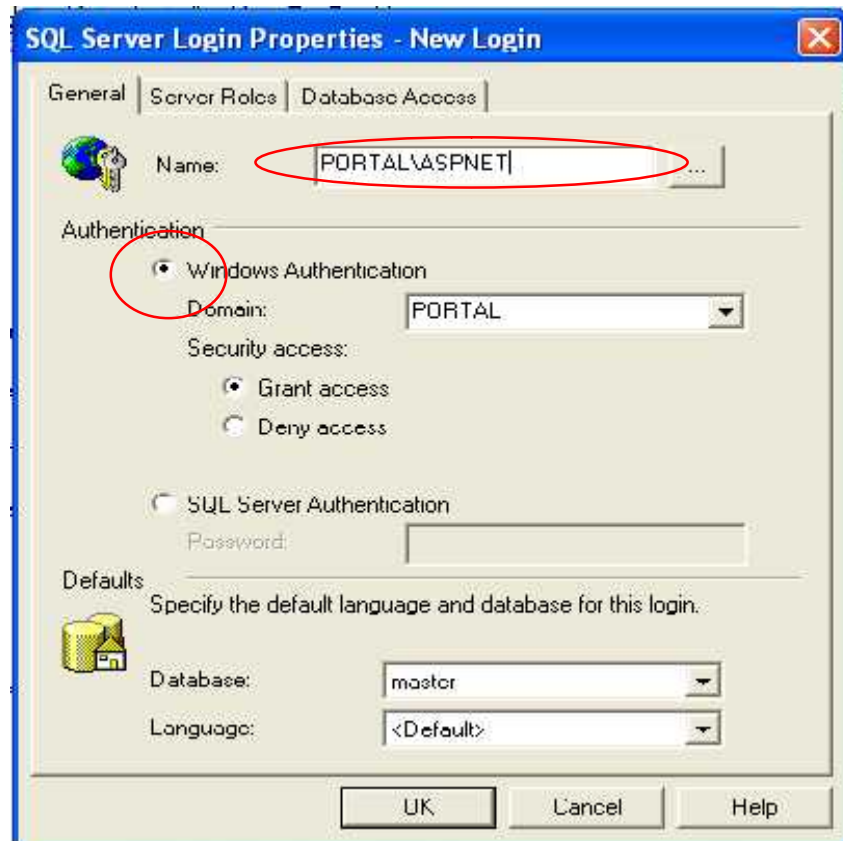


Figure (4.3) Adding ASPNET account on SQL Server logins group

4.3 Database implementation

The system database is implemented using the SQL server 2000 enterprise manager with the following properties:

- A. SQL server 2000 database name: portal.
- B. Database normalization, it normalized to insure that all tables and relations are not contained redundancy data and other database problems , so that it make it correct.
- C. Database creation : all keys which include, primary and foreign are created to ensure database consistency and correct relations.
- D. Database connection

At this stage, the path to the data stored in the database could be opened and used as a two channel path.



In fact, to configure a connection, it is better to do it programmatically i.e. by coding, but using wizards that create connections is preferred at the starting point of the database connection establishment process.

The core here is whether a connection secure or not, but in configuring the SQL server 2000 we have chosen the windows only authentication, now in this stage we have to integrate the connection security with the SQL server 2000 security mode, thereby making the connection secure, this could done by wizard, as it appears in figure (4.4) shown bellow, we select the integrated security option to be used when creating a connection to the database using Visual Studio.NET

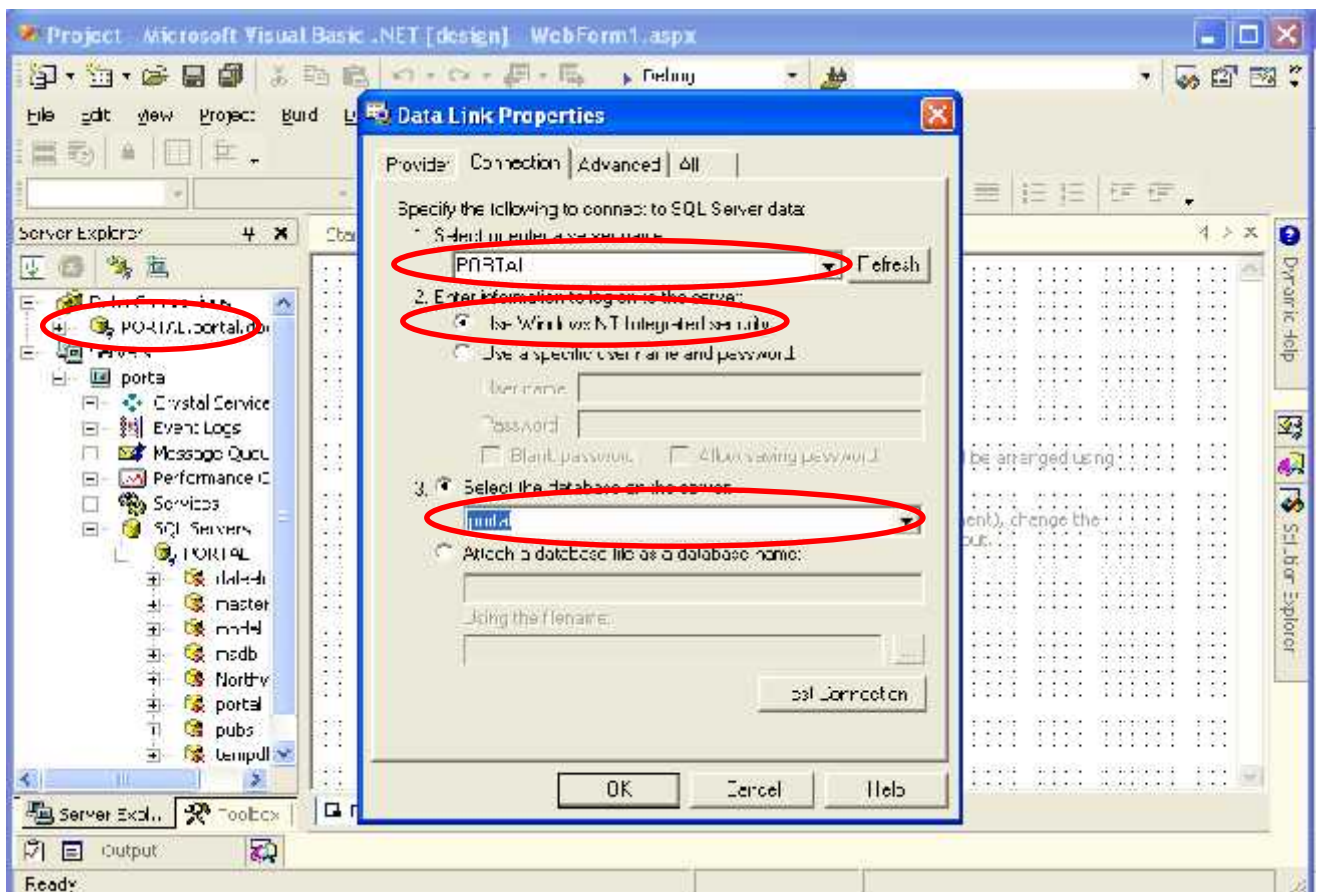


Figure (4.4) Creating Integrated windows security data link

After connect with database we can add, delete, update, and select data from database.



Particular example after making connection with database:

This example to write data to database:

❖ Code writes at ASP.NET code behind page:

- Create SQL connection string:

```
Me.SqlConnection1.ConnectionString "workstation
id=PORTAL;packet size=4096;integrated
security=true;data source=PORTA" & "L;persist
security info=False;initial catalog=portal"
```

- Definition of SQL command:

```
Dim cmd1 As New
SqlCommand("insertcolb",SqlConnection1)
cmd1.CommandType = CommandType.StoredProcedure
```

- Definition parameters:

```
Dim p1 As New SqlParameter("@col_no",
SqlDbType.NVarChar)
Dim p2 As New SqlParameter("@brd_no",
SqlDbType.NVarChar)
```

```
p1.Direction = ParameterDirection.Input
p2.Direction = ParameterDirection.Input
```

```
cmd1.Parameters.Add(p1)
cmd1.Parameters.Add(p2)
```

```
cmd1.Parameters("@col_no").Value =
Session("colno")
cmd1.Parameters("@brd_no").Value = brdno.Text
```

- Execute command:

```
SqlConnection1.Open()

cmd1.ExecuteNonQuery()

SqlConnection1.Close()
```



❖ Code write at SQL server 2000

```
CREATE PROCEDURE [dbo].[insertcolb]
@col_no numeric (9),@brd_no numeric (9)
AS
insert into brd_col values (@col_no,@brd_no)
GO
```

4.4 Supporting software :

Many other software tools where required to improve our system, these are used to support the appearance of GUI of our web application. So we used macromedia Flash 6.0, adobe Photoshop 8.0, and PhotoImpact 8.0. All of these software's are described precisely in chapter two (Software Requirements Specifications).

4.5 Operating the system:

To operate the system properly. Many steps should be executed before the system operates as expected in its environment:

- Configuring the network (physical connection).
- Assigning appropriate IP addresses to the computer in the network.
- Setting up the .NET framework.
- Building the system Database.
- Creating the Database connection with security options.
- Setting up the system on the server (publishing through IIS).



To operate the system from development environment follow the below steps:

- ❖ From start menu select Microsoft Visual Studio.NET 2003.
- ❖ Then will appear window that ask you either to open existing project or open new project, select open existing project.
- ❖ Double click on project that named " PAP ".
- ❖ Then you can make running and brows the web.

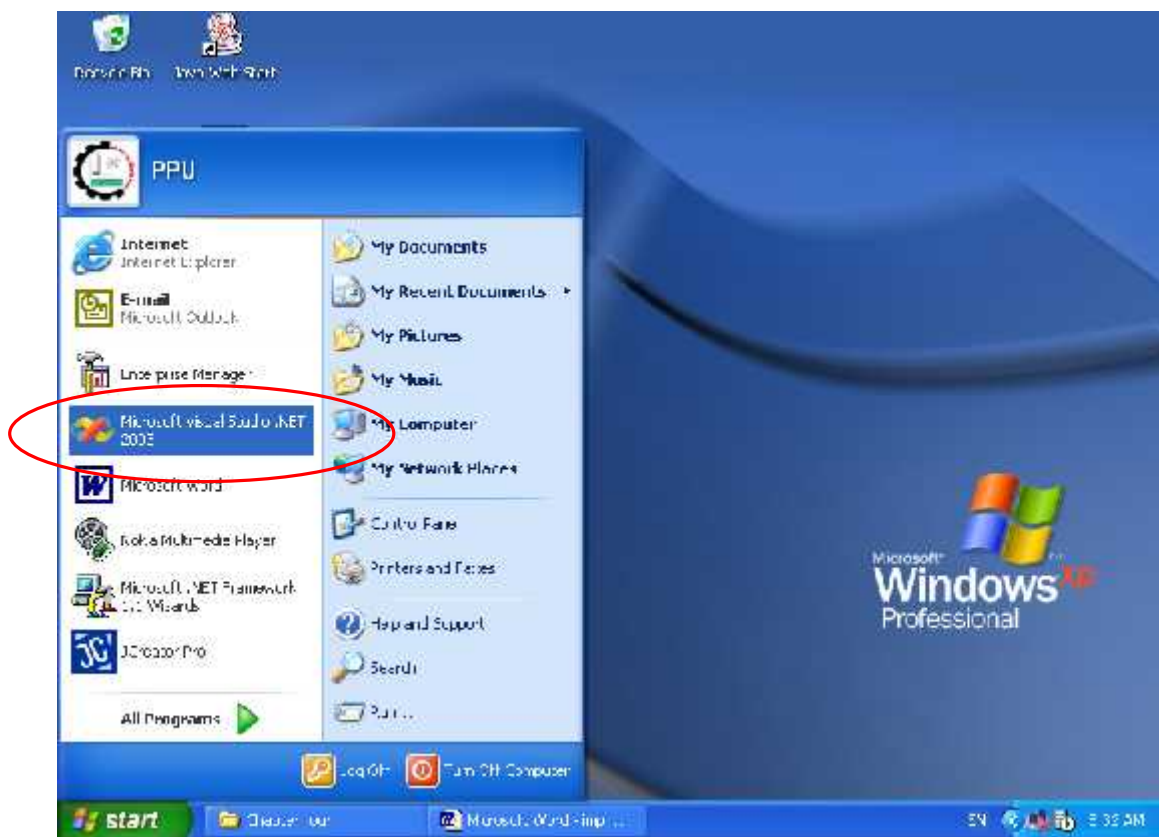


Figure (4.5) Select Microsoft Visual Studio.NET 2003

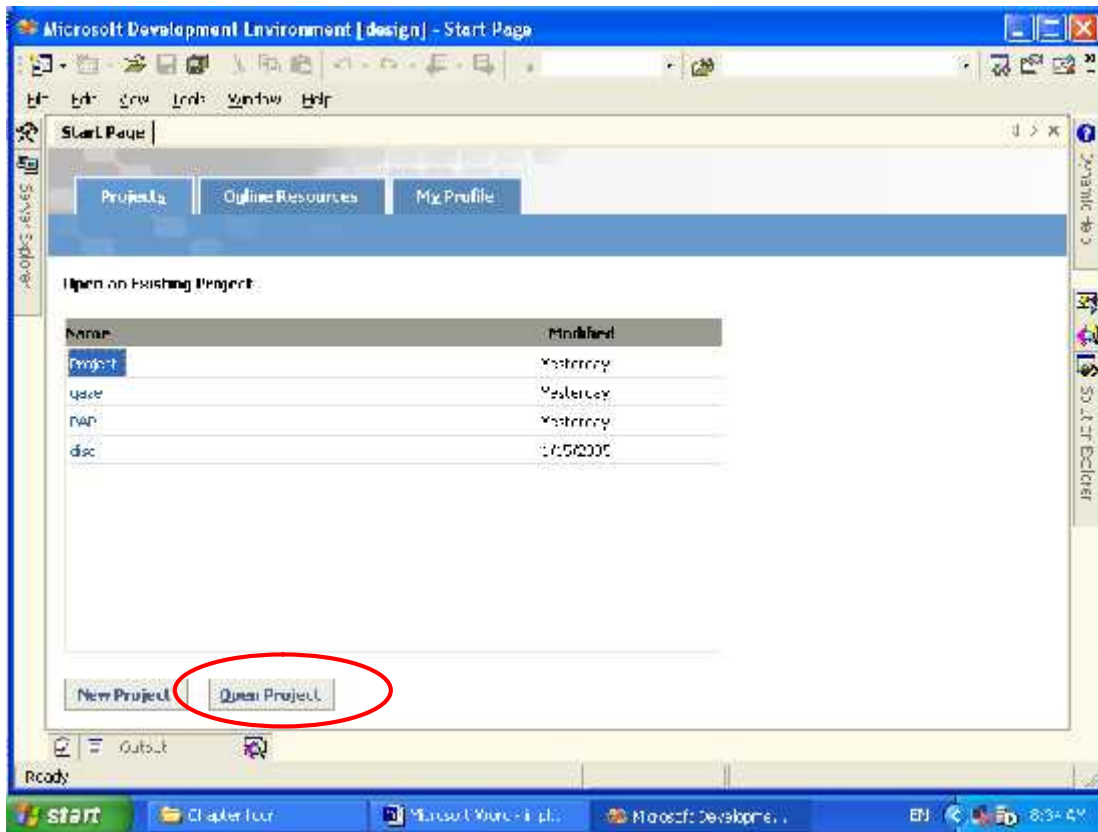


Figure (4.6) Select Open Project

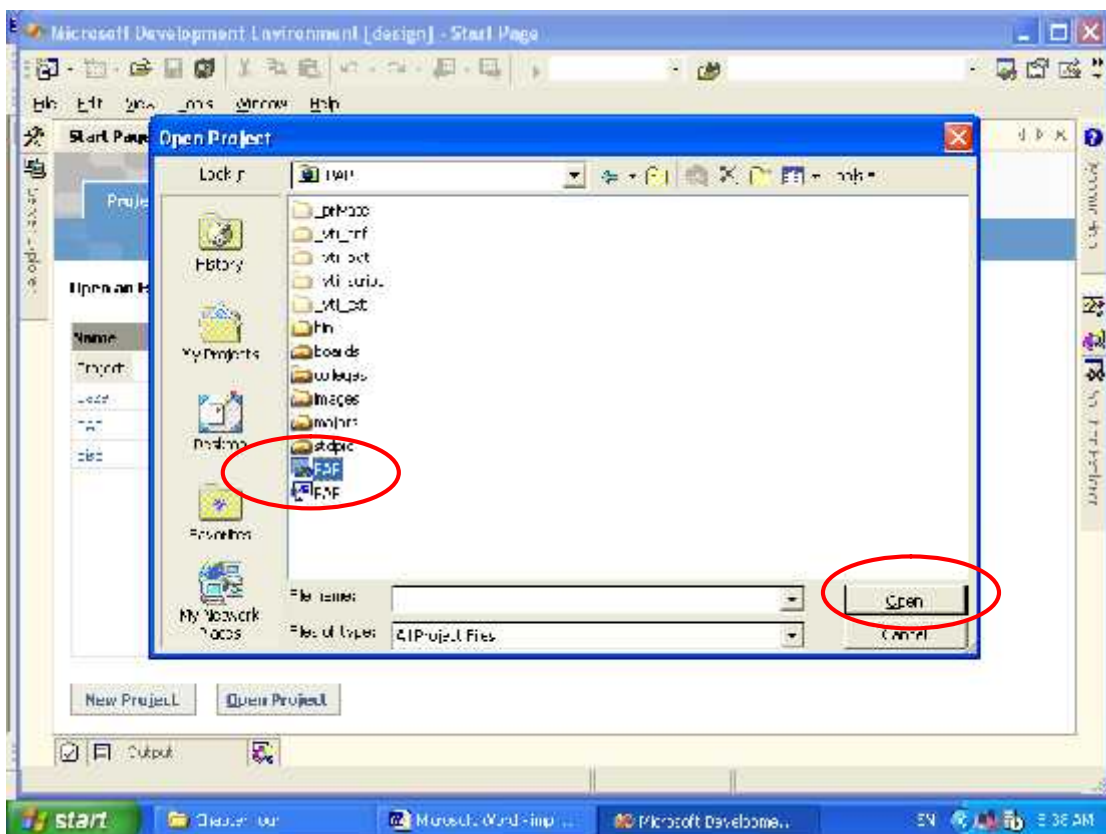


Figure (4.7) Select PAP project



- ❖ Or you can brows the web direct from Microsoft Internet Explorer 4.0 or later (6.0 is recommended).

Write the following URL on address text box in explorer then click enter:

`http://localhost/PAP/index.aspx`

4.6 Coding:

The principle improvements in programming are the reusability and code auto generation, this improvement and others reduce the time of programmer to write code. For development of a system, in our case a web-based application we used visual Studio.NET, which simplifies the development of powerful, reliable software application by providing familiar and shared development environment. It contain pre-built component, programming wizard, as well the ability to use component built using various language. In Visual Studio.Net there is a single integrated development environment (IDE), which provides a sense of what you see is what you get (the visual programming environment).

The usage of this tool for the purpose of programming and coding reduce the time and efforts and thereby increasing the performance.

When using Visual Studio.Net as a programming environment we gain the benefits of the separation between writing the logical code (the program functionality) from one side and the design of the appearance and graphical user interface (GUI) from the other side.

This application occurs by the new way that enables the programmer not to spend his time or determining programmatically where each control on a page should appears, however he could write the code that make the logic of each control operation in separate page called code behind page. We have appended the source code written manually for the main functionalities in our application.



4.7 Development process implementation.

The system was first created by going to Microsoft visual studio.NET development environment and named as PAP.

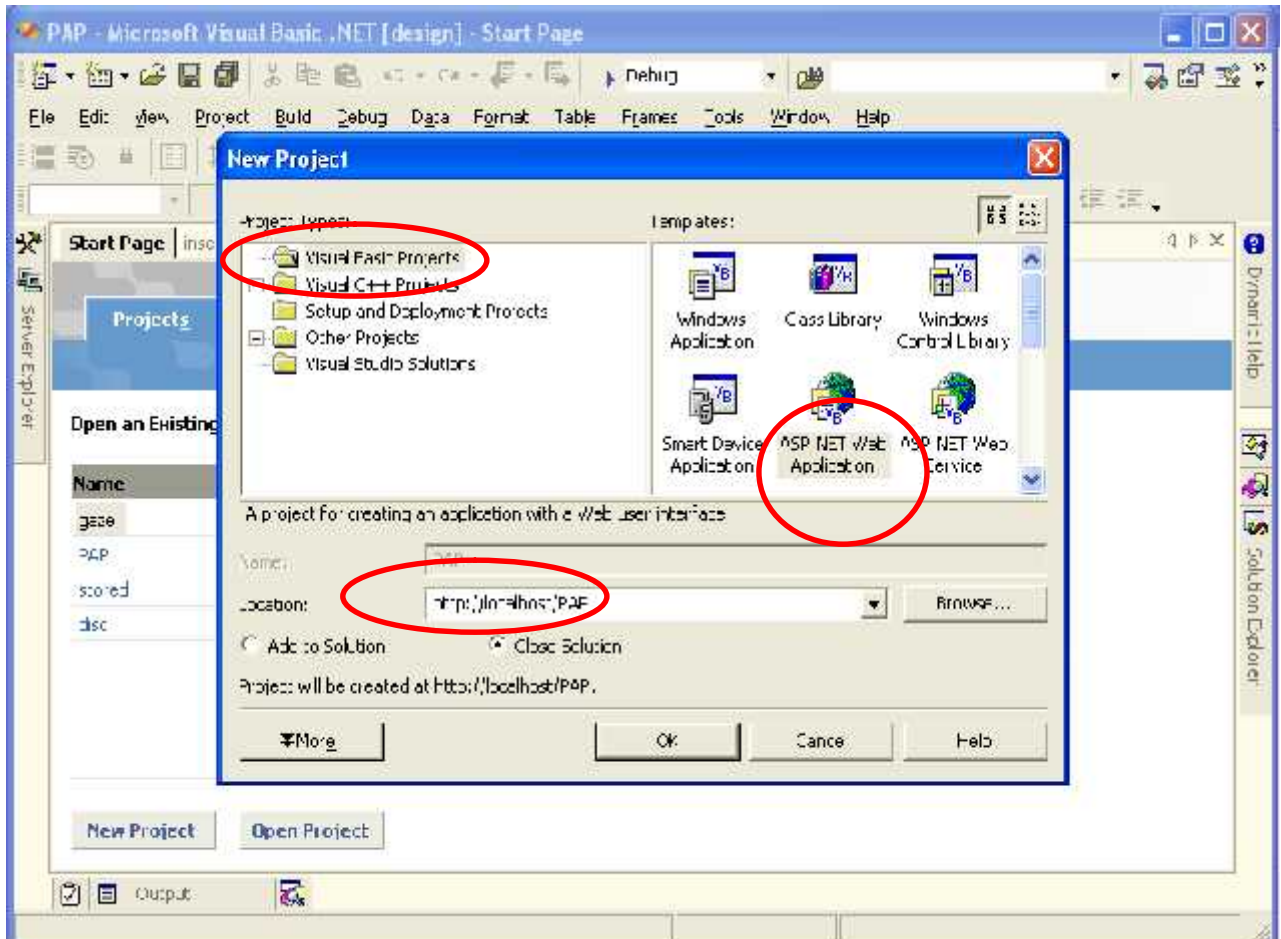


Figure (4.5) Development Process (create new project)



5.1 Introduction

Before we deliver the final system copy to the customer we must test the system to ensure that the system meet its specification and is worked properly as we expected.

Testing process is divided into number of successive stages or levels that will be implemented to ensure form system validation.

This phase from software development is a most important phase since it consumes a lot of time, and require a great effort to get system to be reliable, this doesn't mean that the system must be completely free of errors and faults, but it means that the system should be good enough for its intended use.

The testing process include four successive levels as below:

- System units testing.
- Module testing.
- Subsystems testing.
- Integration testing.

Testing process will take place in time space that was assigned for the testing process, table (5.1) illustrate testing process schedule:

Time in week	1st Week	2nd Week
Testing Process		
Unit testing	■	
Module testing		■
Subsystem testing	■	■
Integration testing		■

Table (5.1) Testing schedule.



5.2 System unit testing

at this stage of testing process we separate or divided the system into subsystems and components that will be tested separately to ensure that they operate correctly and meet its specification.

In our system the administrator operations is the most operations that affect the system and database , so we will test login, delete boards, and change password process, and for the students operations we will test change password, and voting process.

We will insert valid inputs one time, and invalid another time, and we will see the system responding result.

This table describes the unit testing process for administrator login:

Unit testing process case	Input data testing		Expected result	Actual result
	Username	Password		
Valid login username and password	111	456	Administrator page loaded	Administrator page loaded
Wrong username or password expression	Adk14	\$hy1/dj	Invalid inputs format	Invalid inputs format
Invalid login username or password	111	654	Error alert appear, return to login page	Error alert appear, return to login page

Table (5.2) Administrator login process testing



Now we will testing change password process, so the following table illustrate it:

Unit testing process case	Input data testing			Expected result	Actual result
	Username	New Password	Confirm it		
Valid login username and password and confirm it correctly	111	678952	678952	Password updated successfully	Password updated successfully
Wrong username or password expression	Gfr342	Efr&4\$	Efr&4\$	Invalid inputs format	Invalid inputs format
Invalid login username	Frd324	678952	678952	Error alert appear, return to change pwd page	Error alert appear, return to change pwd page
Wrong in password confirmation	111	678952	678925	Error alert, "Confirm password correctly"	Error alert, "Confirm password correctly"

Table (5.3) Change password process testing



Now we will testing delete boards process, the following table show the process:

Unit testing process case	Input data testing		Expected result	Actual result
	Select type of deleted board	Select board you want to delete		
Select deleted board type, then select the board from dropdown list to delete it, then click delete button	Expired boards	Board title	The board was deleted successfully	The board was deleted successfully
Doesn't select board type to be deleted, and click on delete button	Not selected	Empty list	Error alert, "Please select the board type first"	Error alert, "Please select the board type first"
Select board type, doesn't select the board to delete, and click on delete button	Board Name	Not selected	Error alert , "Please select the board you want to delete from list"	Error alert , "Please select the board you want to delete from list"

Table (5.4) Delete boards process testing



The table below show the testing of change password for students process:

Unit testing process case	Input data testing			Expected result	Actual result
	Username	New Password	Confirm it		
Valid login username and password and confirm it correctly	Session("b")	111222	111222	Password updated successfully	Password updated successfully
Wrong password expression	Session("b")	Efr&4\$	Efr&4\$	Invalid inputs format	Invalid inputs format
Invalid login password	Session("b")	\$j\7@	\$j\7@	Error alert appear, return to change pwd page	Error alert appear, return to change pwd page
Wrong in password confirmation	Session("b")	111222	222111	Error alert, "Confirm password correctly"	Error alert, "Confirm password correctly"

Table (5.5) Change student password process testing.



The following table illustrate the student voting process testing:

Unit testing process case	Input data testing			Expected result	Actual result
	Yes	No	Maybe		
Doesn't select any choice, then click submit button	Not selected	Not selected	Not selected	Error alert, "select at least one choice from list"	Error alert, "select at least one choice from list"
Select yes, no, or maybe " first time"	✓	Or ✓	Or ✓	Thanks, process done successfully.	Thanks, process done successfully.
Select yes, no, or maybe at another time for the same question	✓	Or ✓	Or ✓	Error alert , " Sorry, you were voting"	Error alert , " Sorry, you were voting"

Table (5.6) Answer question (voting) process testing.



5.3 System module testing

At this level we test the modules which encapsulated related components, so it can be tested without other system modules. Procedures, functions, and system interface are tested.

5.4 Sub-system testing

Our system include mainly three subsystems or clusters there are:

1. *Administrator subsystem.*

We tested all functions and procedures that operate through administrator pages to ensure that they are work correctly and meet its specification.

And we tested the data flow and transfer between procedures and functions, and interfaces and how data and information layout on pages and how they are appearance.

And we tested security side, to ensure that the system is secure and is safe from hacking.

2. *Instructor subsystem.*

We tested instructor subsystem as we do at administrator subsystem, we tested all functions and procedures to ensure that they work correctly and meet its specification and its behave as we expected.

And we tested the interface , and how the information are layout and appearance on screen, in addition we tested security to prevent any attempt to harmful or destroy the system, and we tested data transfer between functions and procedures to ensure that all data are correctly flowed and used.



3. *Student subsystem.*

As the previous subsystems we tested all functions and procedures, and in every page we tested the data flow between the user and database to ensure that they are transfer correctly and in correct way.

And we tested user interface, and how information layout on pages, and we take in our consideration the interface colors that affect user, so we tested it and we use suitable color for user, and we tested security to ensure that the system is secure and safely.

5.5 Integration testing

After we tested the subsystems and then we tested modules as encapsulated components, at this level we integrated all modules together and tested them as a whole system to ensure that the system works properly and meet its specifications.

The testing of the system integration indicates that the system works properly and meet its specification.



5.7 Snapshots

in this section we will select some snapshots from the web site to display they here to show how will the real system behave under certain situations and conditions.

5.7.1 Some of the administrator pages.

- 1) Not allowing invalid inputs formats.

This snapshot for login page display the system handle invalid input formats:



Figure(5.7) Invalid input formats testing snapshot fro login as administrator.



► This snapshot for login page with valid input formats:



Figure (5.8) Valid input format testing snapshot fro login as administrator.



► Administrator Home Page.

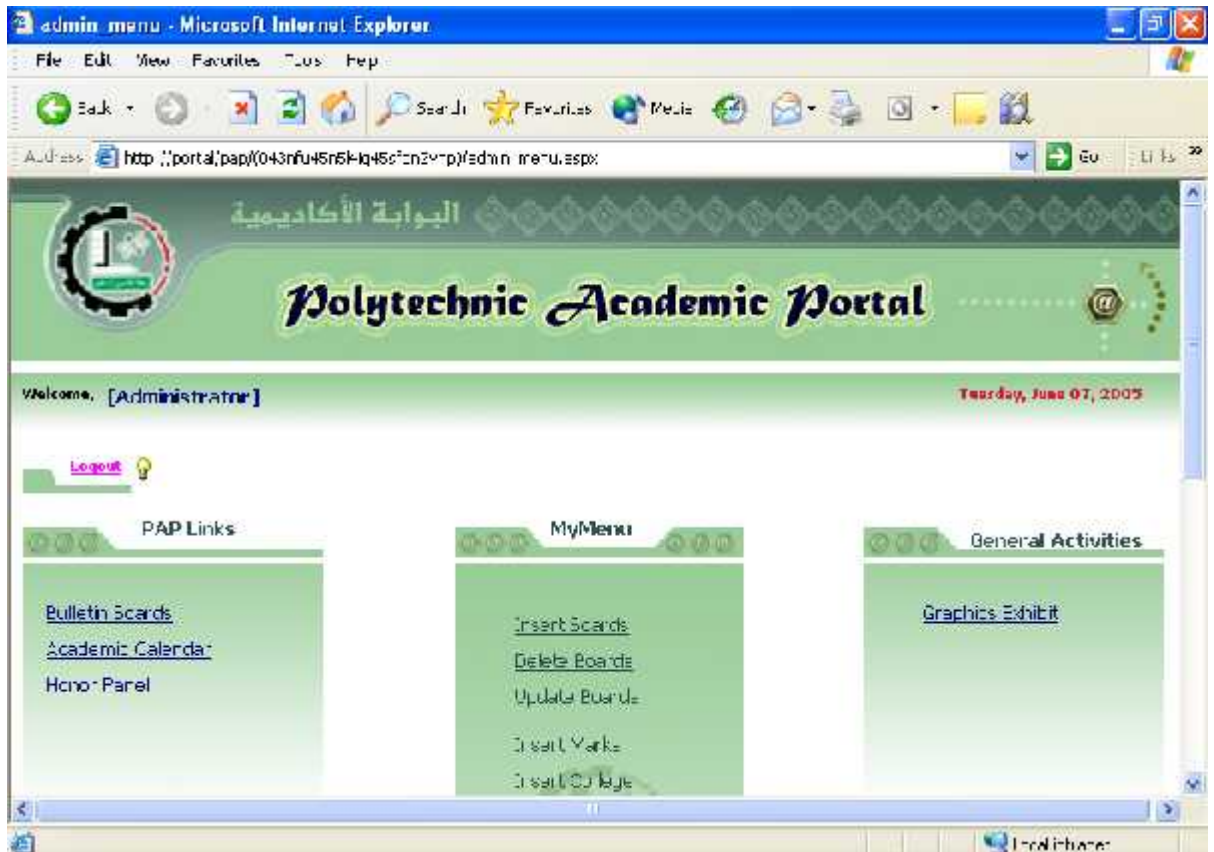


Figure (5.9) Administrator Home page.
(this page appeared after insert valid data in login page as administrator).



- 2) This snapshot for insert new boards page display how the system handle the invalid inputs:

The screenshot shows a web browser window titled 'insert_boards - Microsoft Internet Explorer'. The address bar shows the URL 'http://localhost:1024/insert_boards.aspx'. The page content includes a welcome message for an administrator, a date of Tuesday, June 07, 2005, and navigation links for 'Site Map' and 'Help'. The main heading is 'Insert New Boards Form'. Below this, there is a 'Boards Data' section with several input fields: 'Chg' (a dropdown menu), 'F/7/2005' (a date field), 'Due to' (a dropdown menu), '5/15/2005' (a date field), 'Title' (a text field containing '1234'), and 'Board No.' (a text field containing '123'). A red error message 'Character: [a-z] allow only' is displayed above the 'Title' field. Below the 'Boards Data' section, there is an 'Author' field containing 'Administrahtr' and a question 'Do you want to send this board more than one time, for multicestination?' with 'Yes' and 'No' radio buttons. At the bottom, there are sections for 'Board Body' and 'Board Text'.

Figure (5.10) Invalid input format testing snapshot for insert new board.



- ▶ This snapshot display how the system behavior when administrator or instructor leave some fields empty and then click upload new board:

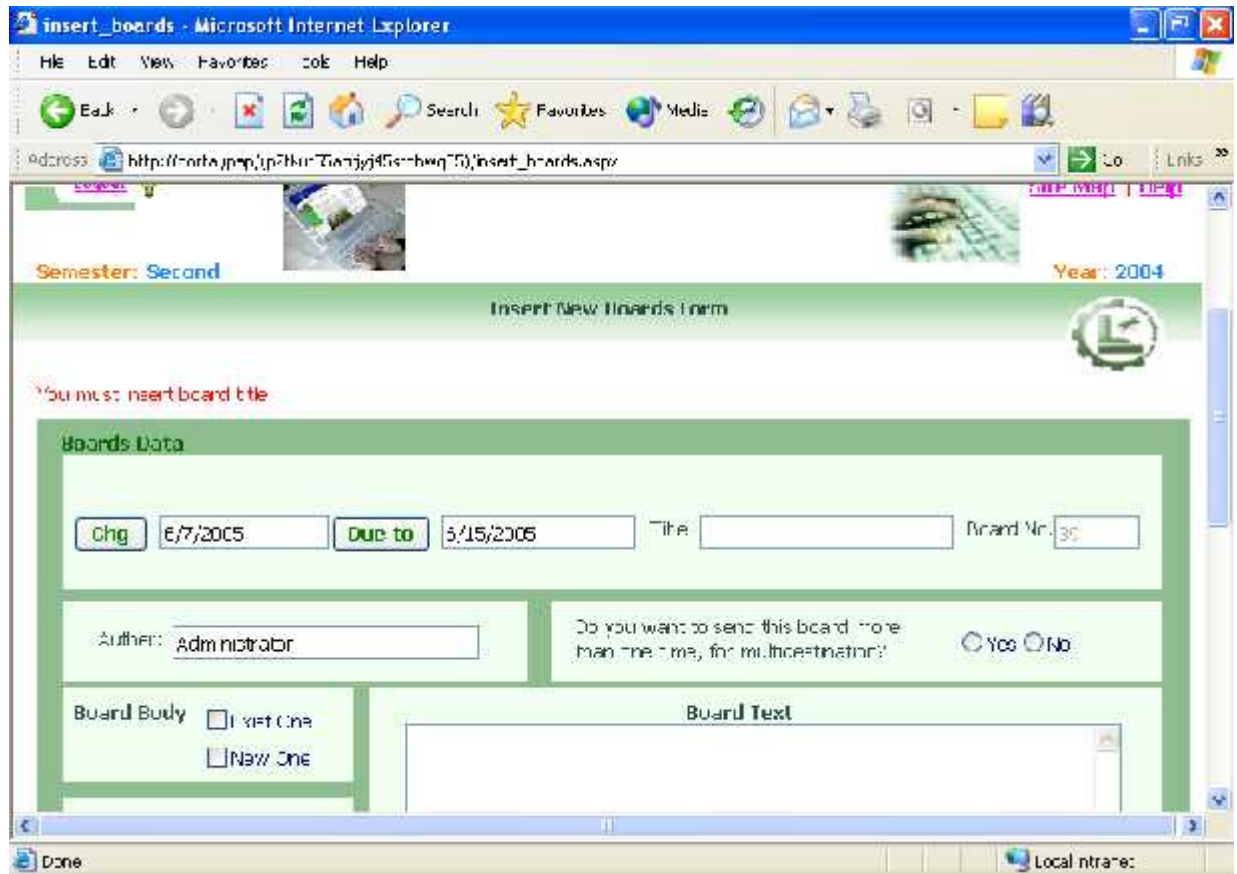


Figure (5.11) Leave title field empty and click upload new board



- ▶ This snapshot show the system behavior when the administrator or instructor insert all required fields and the data is valid:

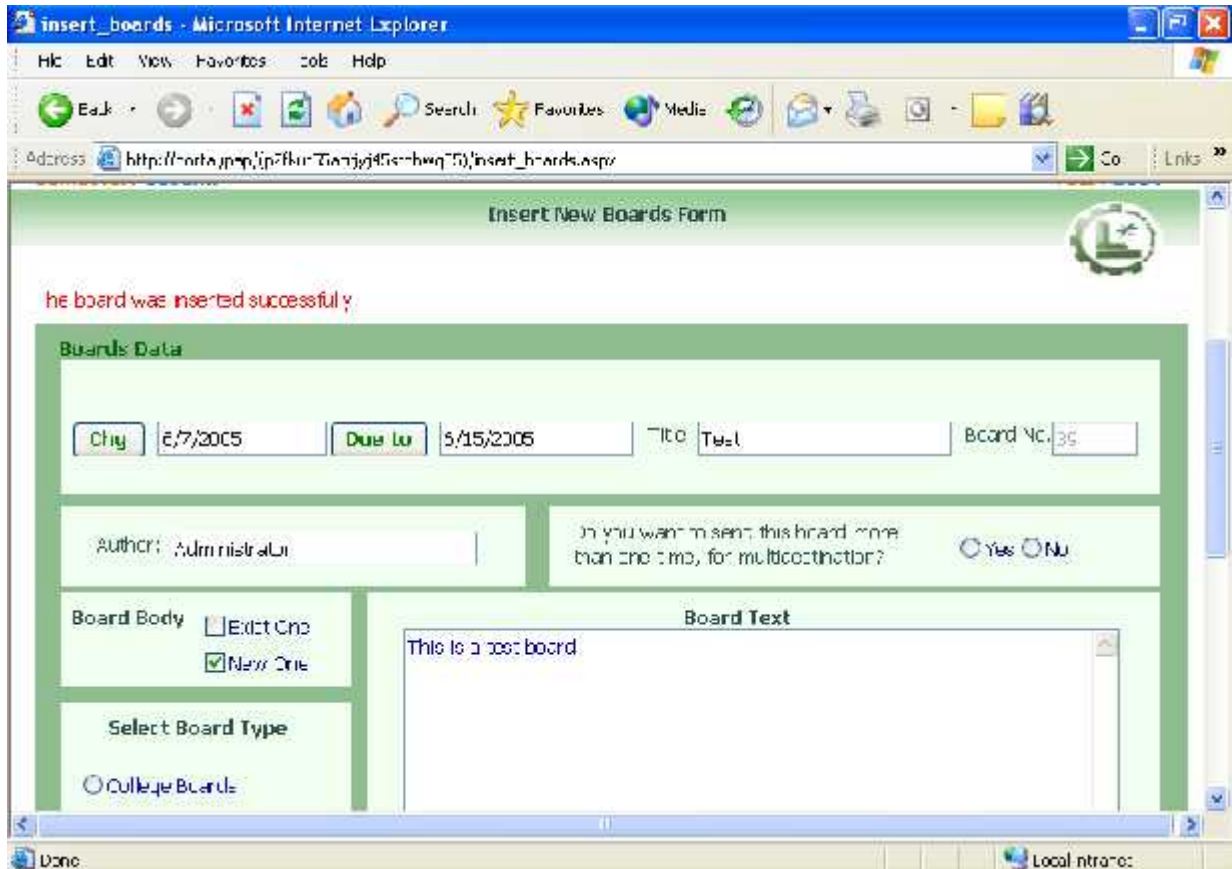


Figure (5.12) Insert all fields and valid inputs for insert new board



- ▶ This snapshot show the new board (Test) in marquee for students after it inserted by administrator:

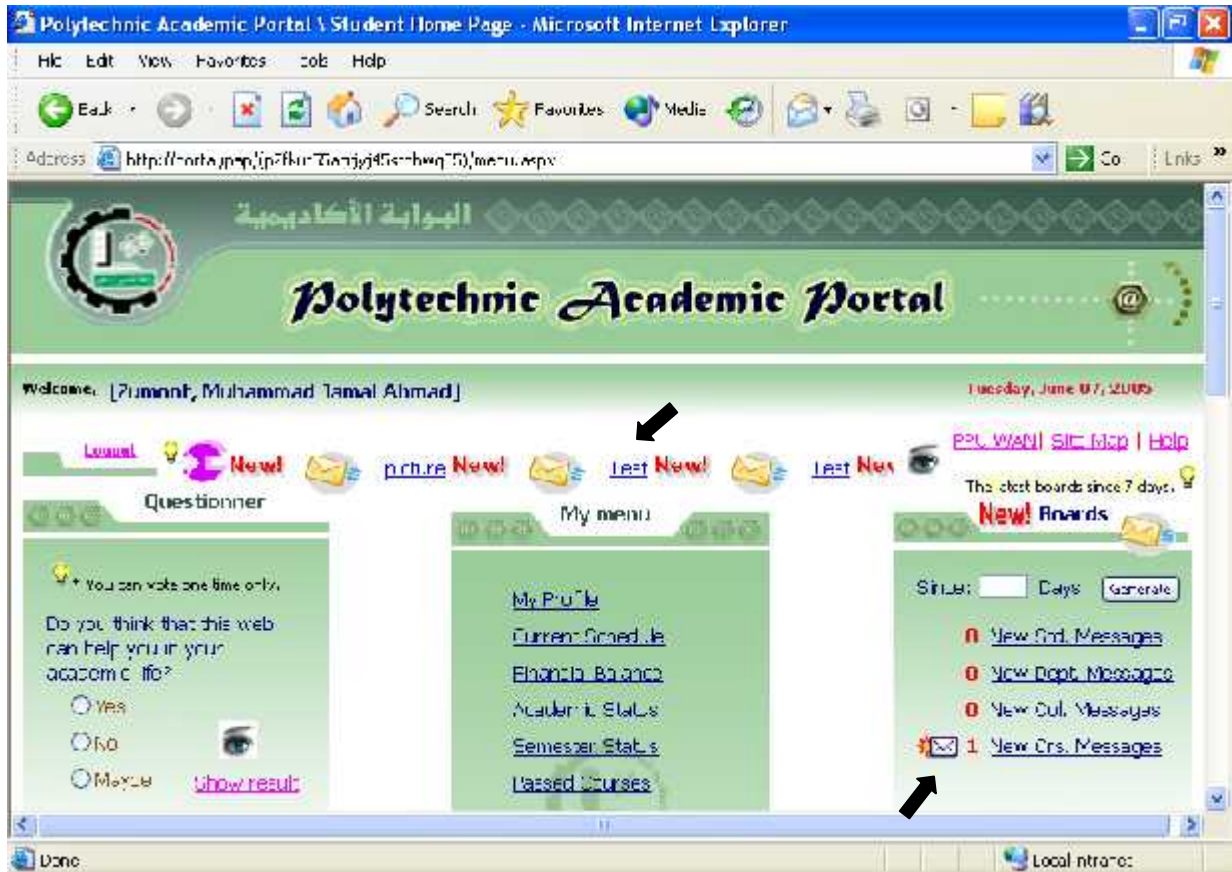


Figure (5.12) New course board in marquee and new boards area.



3) This snapshot for insert marks page display how the system handle the invalid Inputs.

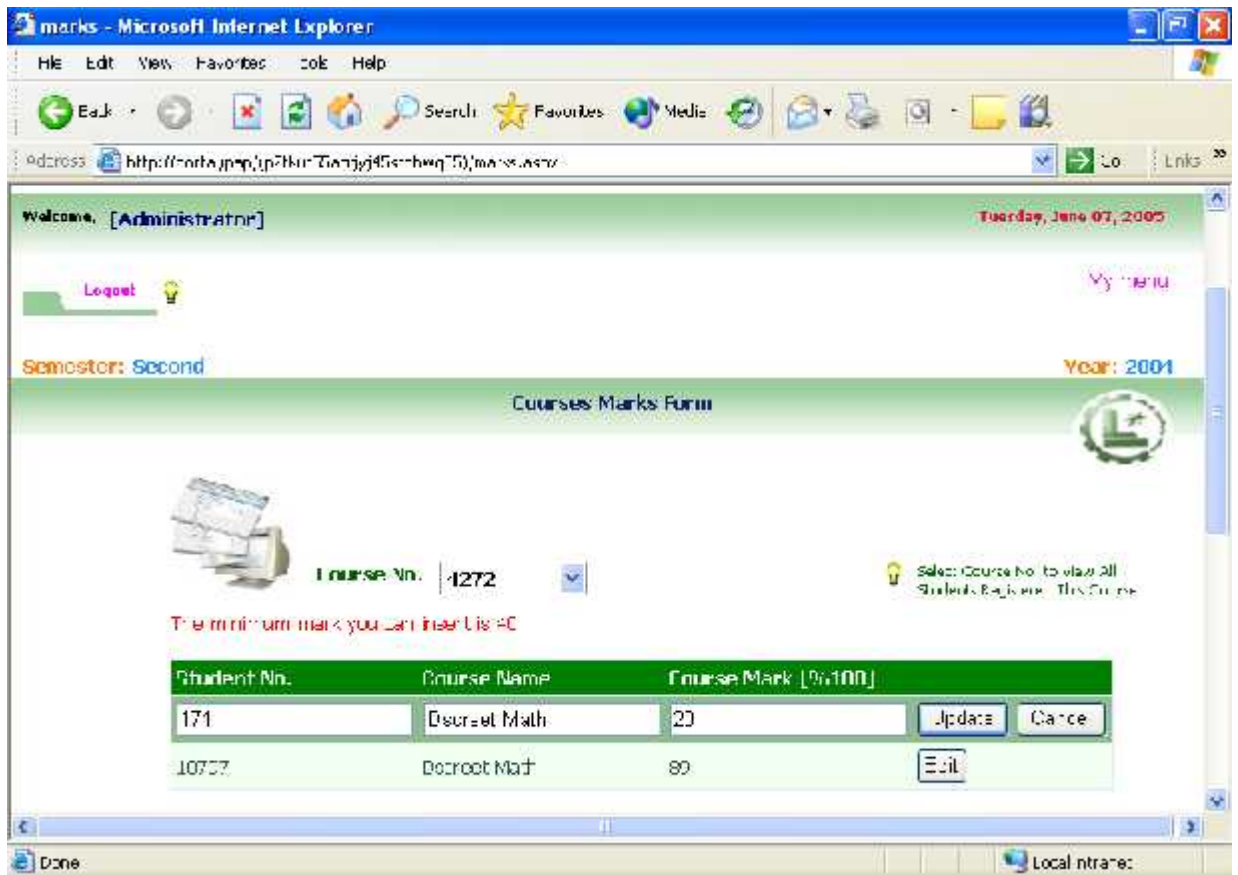


Figure (5.13) Invalid input testing snapshot for insert mark.



- 4) This snapshot for delete boards page, display how the system behavior when the administrator delete boards.

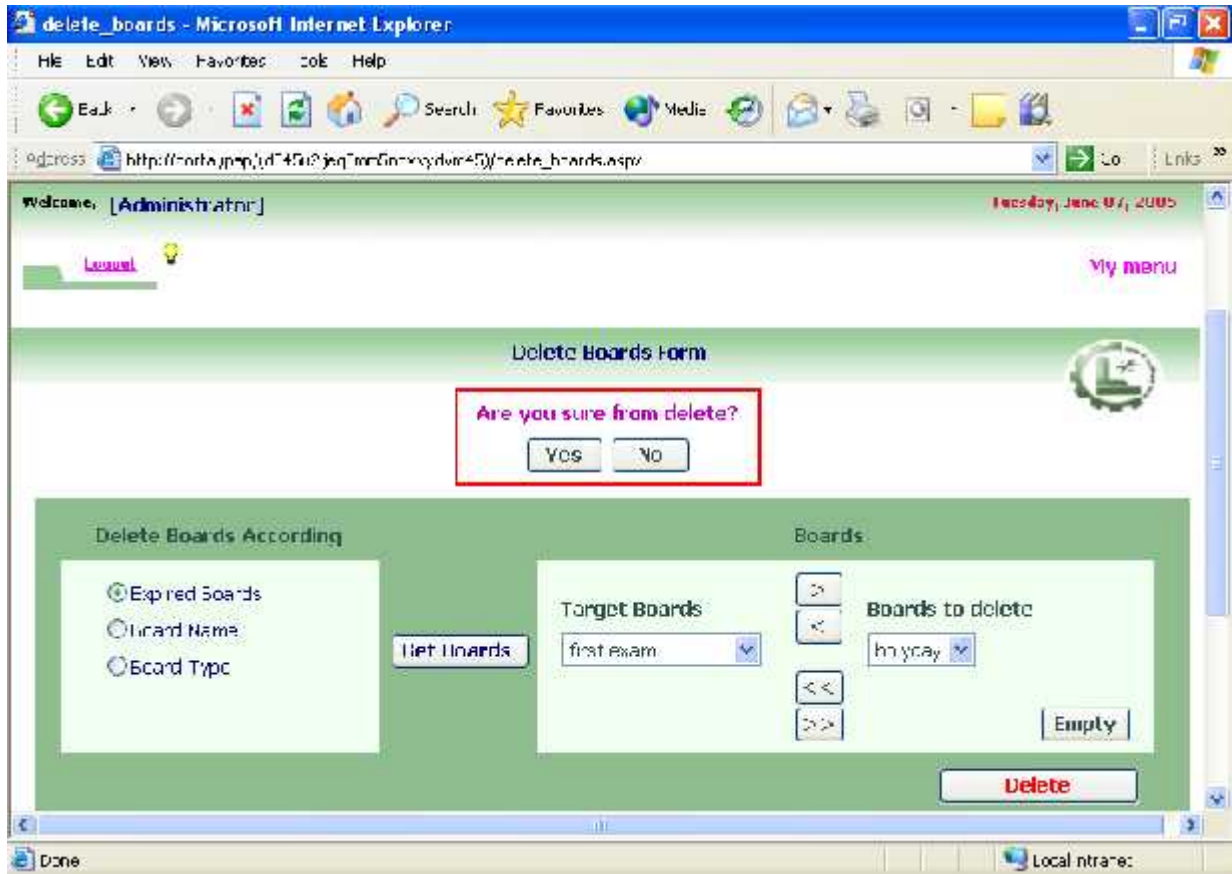


Figure (5.15) delete expired board from database.



- ▶ This snapshot show how the system behavior after select yes from confirm dialog box.

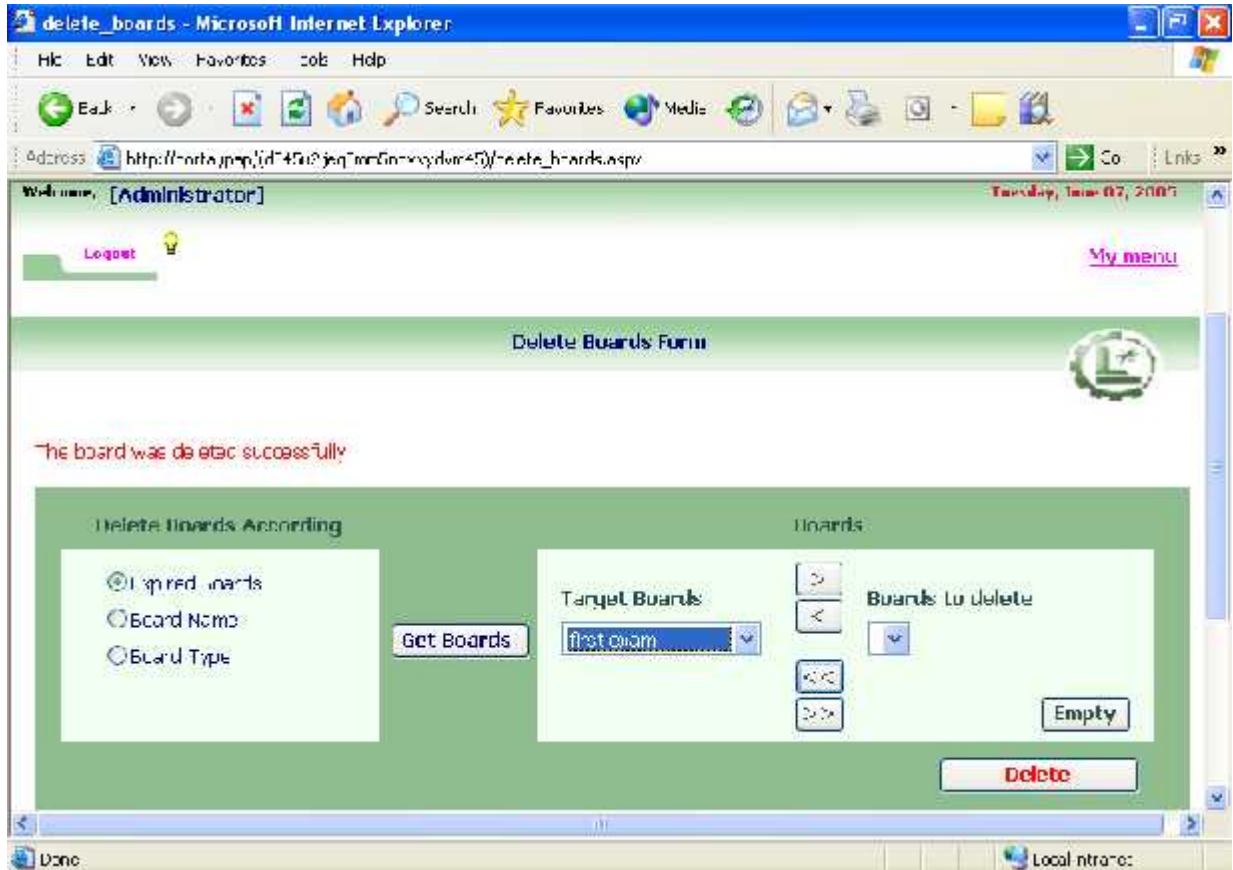


Figure (5.16) Snapshot after deleting the board successfully.



- 5) This snapshot for change password page display how system handle invalid input data format.



Figure (5.17) Invalid data format snapshot for change password.



- ▶ This snapshot display how the system handle invalid confirm for new password in change password page.



Figure (5.18) Invalid confirm new password snapshot.



- 6) This snapshot for questionnaire , display how the system behavior when the student voting for the first time.



Figure (5.19) Voting for the first time from student snapshot.



- ▶ This snapshot display how the system will behavior when the same student will try to vote another time.



Figure (5.20) Voting another time for the same student snapshot.



5.7.2 Some of student pages.

A. Current schedule page.

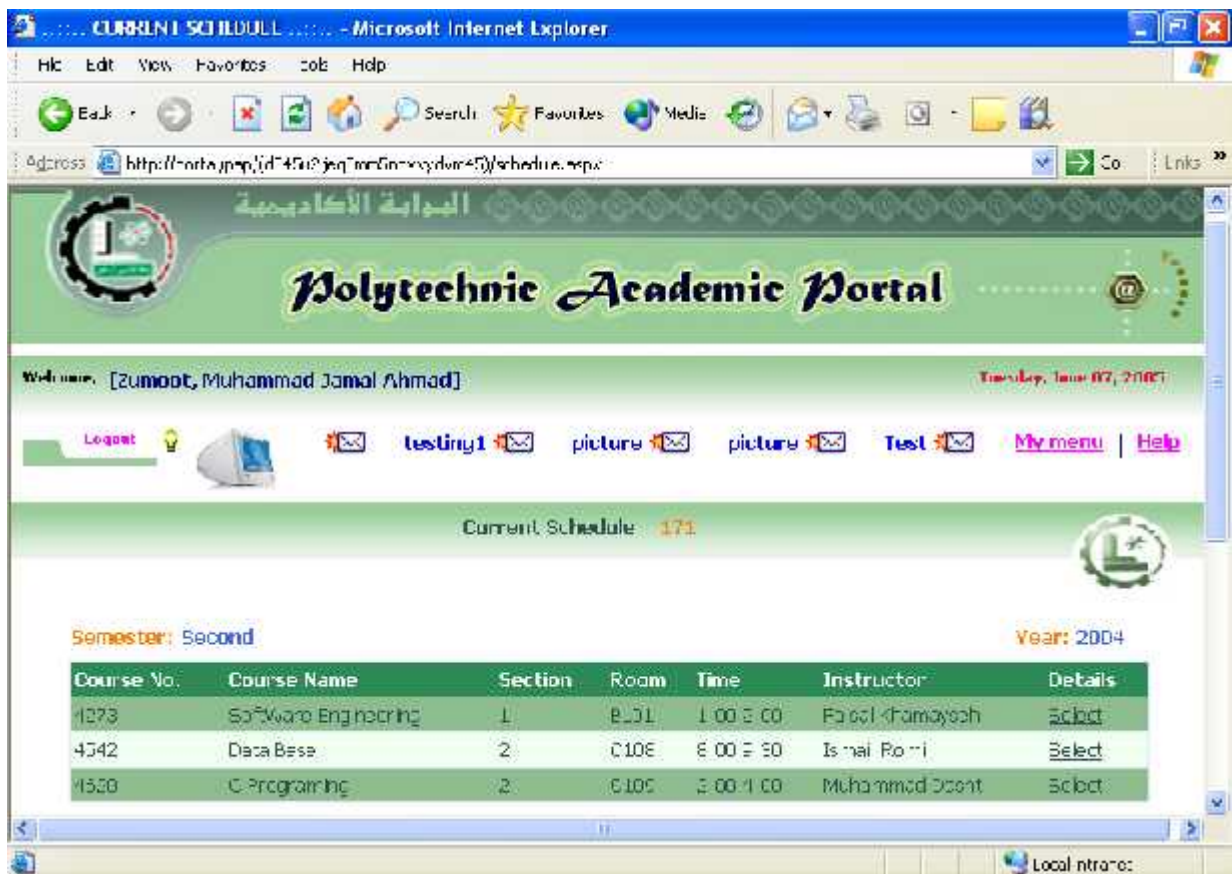


Figure (5.21) Current schedule page for student.



B. Financial balance page.



Figure (5.22) Financial balance page.



C. Finished hours and marks.

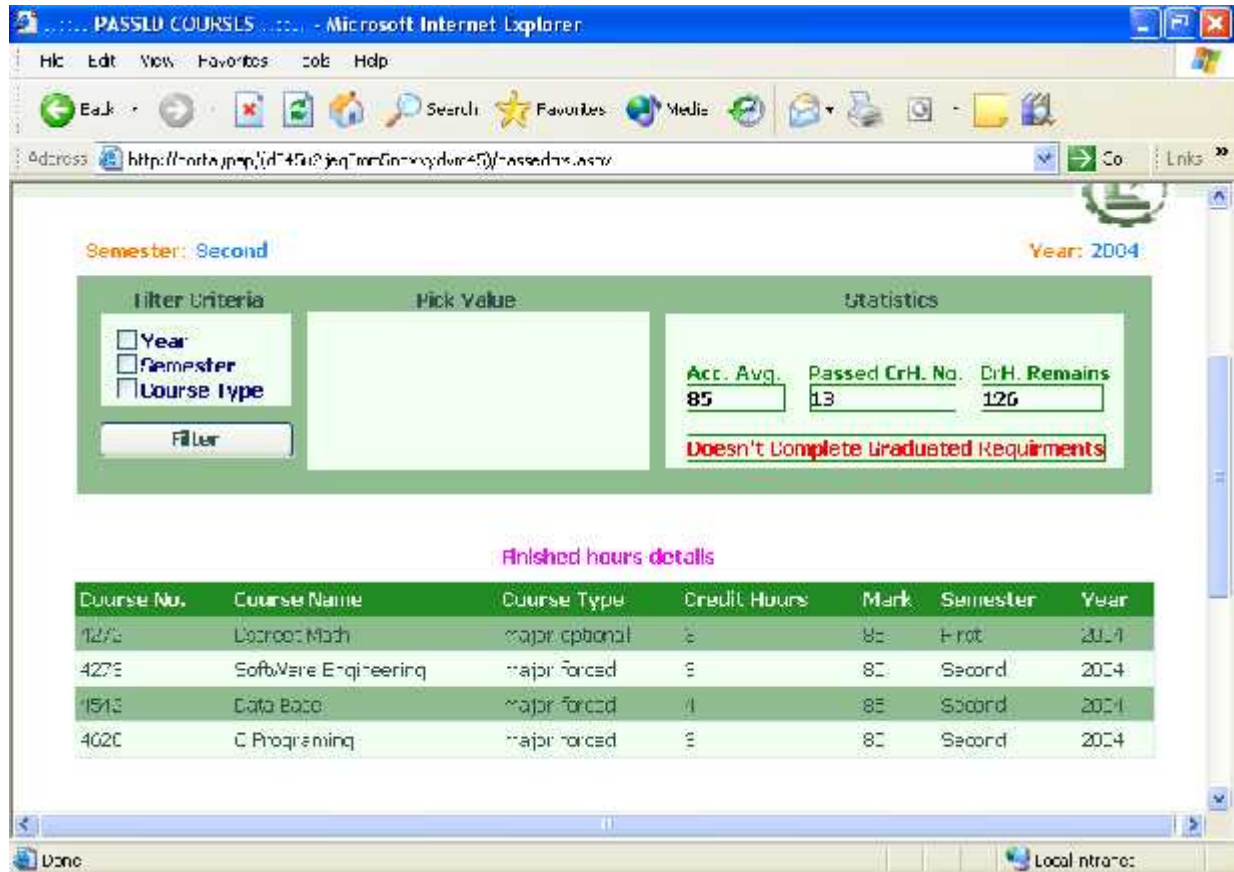


Figure (5.23) Finished hours and marks.



6.1 Introduction

At this chapter we will provide and explain the process and techniques that guidelines the system administrator to keep tracing and maintaining the system after running it.

At this stage, the system which had been developed and tested is needed to take place in the real working environment. However, it is a kind of joke to think about customers as if they were the developers of the system, they are just end users, so those users must be provided by sufficient information and guidance about system deployment and how it could be maintained. In this chapter, we describe the real working environment within which the system will operate, as well how it could be established, how it could be migrated, and how it is maintained.

In this chapter we will describe:

- Maintenance plan.
- Migration.

And we will talk about system maintenance which we divided to:

- IIS maintenance.
- SQL server 2000 maintenance.
- .NET framework maintenance.



6.2 Maintenance plan

In this section we will describe some procedures that should be taken in our consideration to handle any system failure, errors, and other types of problems and exceptions that maybe appear or occur during system life cycle.

1. Backup

Database is the nerve of the system, so it is the most important component of the system, because it contains all tables , views, and stored procedures that contain the data of the system, so we should keep this data from losing and harmful by specific means such as take a backup for this database periodically to enable maintain the system in the future.

This is an issue where the adaptation of the backup methodology (type and time interval) is determined and implemented by the company working on the system itself. The backup on the system database could be configured by means that are provided by the producer company of the DBMS that we have used in our software system which is Microsoft© on its DBMS product (SQL Server 2000).

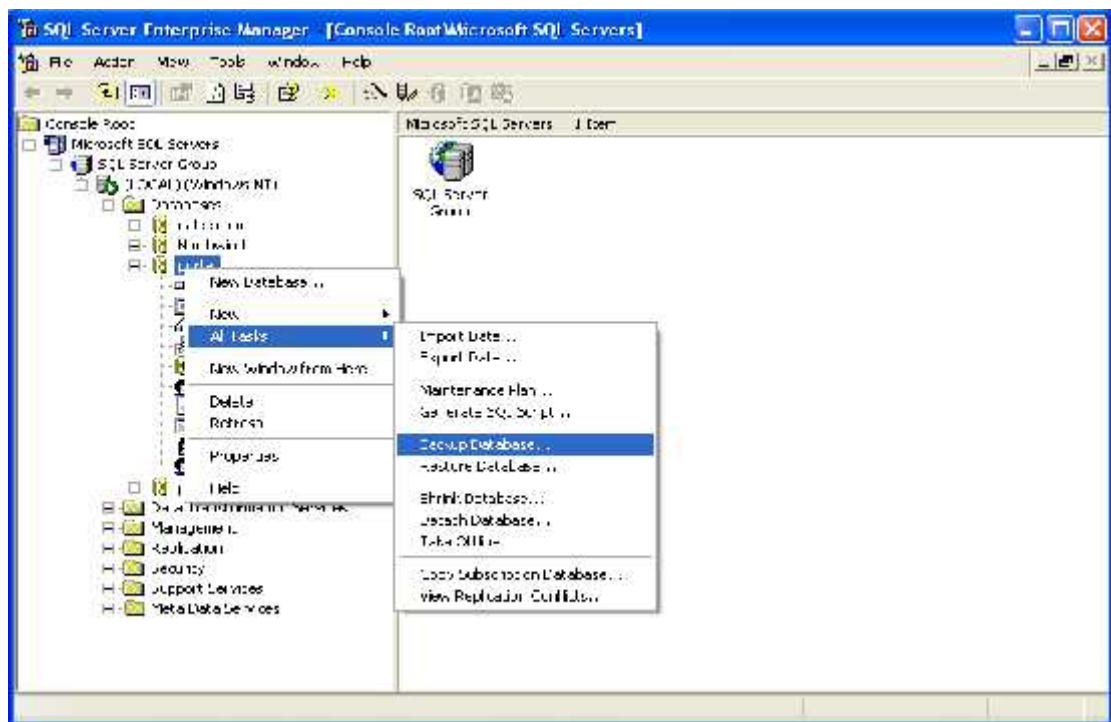


Figure (6.1) Database backup.

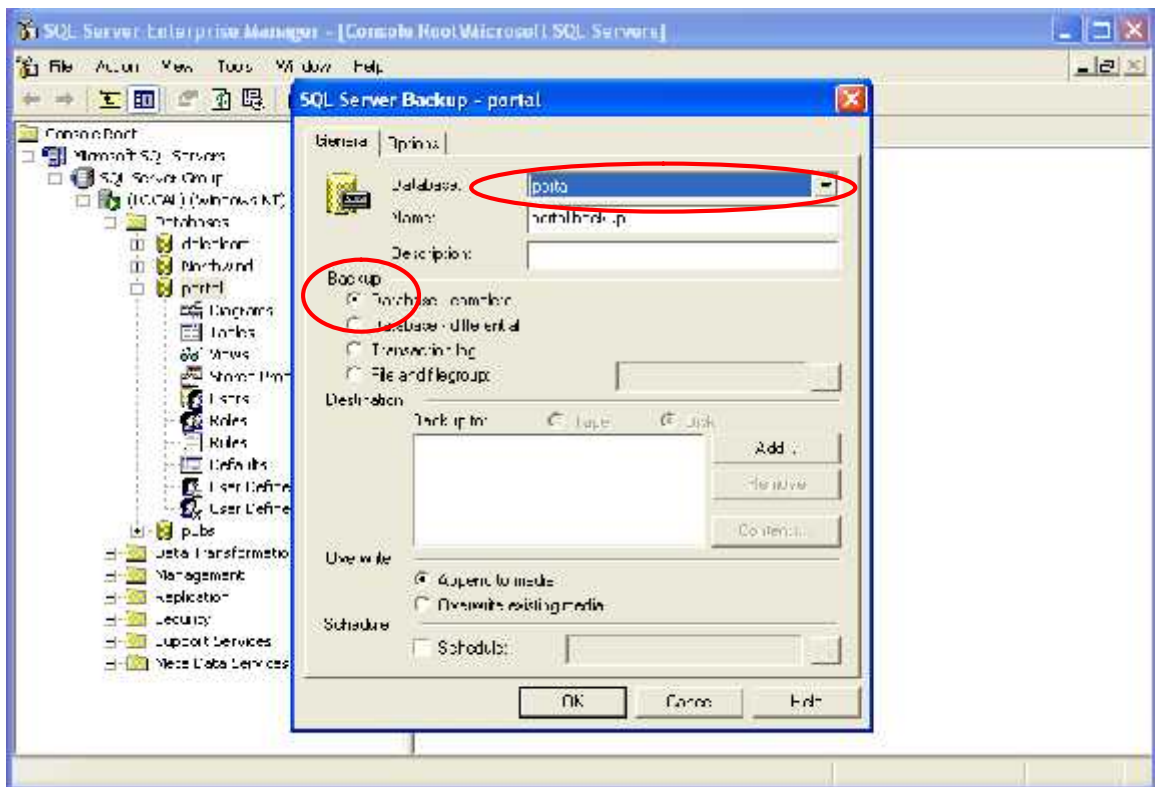


Figure (6.2) Backup options

2. Error reporting

When errors occur, certain actions are to be taken as the contract agreement describes the conditions and situations of the maintenance and the responsibility allocation on the contract sides, such as System Service Request (SSR) which take the following form in many cases:



System Service Request

REQUESTED BY ----- DATE -----

DEPARTMENT -----

LOCATION-----

CONTACT-----

TYPE OF REQUEST

URGENCY

- New system
- System enhancement.
- System error correction.

- Immediate, operations are impaired or opportunity lost.
- Problems exist, but can be worked around.
- Business losses can be tolerated until new system is installed.

PROBLEM STATEMENT

SERVICE REQUEST

IS LIAISON -----

SPONSER -----

----- TO BE COMPLETED BY SYSTEMS PRIORITY

BOARD -----

- Request approved
- Recommend revision
- Suggest user development.
- Reject for reason -----

Assigned to -----
 Start date -----



6.3 Migration

The deployment of the system must be preceded by certain steps so that to work properly within its environment; the production environment has to be established, configured, and a decision of operating on the new system must be taken considering all constraints and risks of the process of migration to the new system. Toward deploying and migrating to the new system we describe here the steps that must be done:

1. Establishment of the production environment:

The minimal requirements of deploying the system are described in chapter one (system specification), and the needed configurations for the machine running the system are described in chapter four (Coding and implementation). For example, we say that our software system will not operate on a machine that doesn't have the .NET Framework, so that the company that decides to work on our system must have all of the production environment elements available.

2. Deciding to deploy the new system :

Here we say that our system was tested and we found that it works well as it should be, the system can operate immediately whenever a suitable production environment is created, but we say that our software is an integrated system that must run with its all parts integrated, nevertheless the system may not work properly. Now, the way of migration to it depends on the technicians and managers recommendations in the company that wishes to deploy this new system.

3. Running the system

After the system being complete it can be running.



6.4 Internet Information Services maintenance.

IIS is the core of web server components , since without it the server cannot be consider as web server ,so it must be maintained to make system operate correctly, because IIS represent a gateway for the application to be published over the internet.

Without IIS we can't publish the web over the internet , so success publish process is dependent on the consistency, security , and make a correct configuration for IIS.

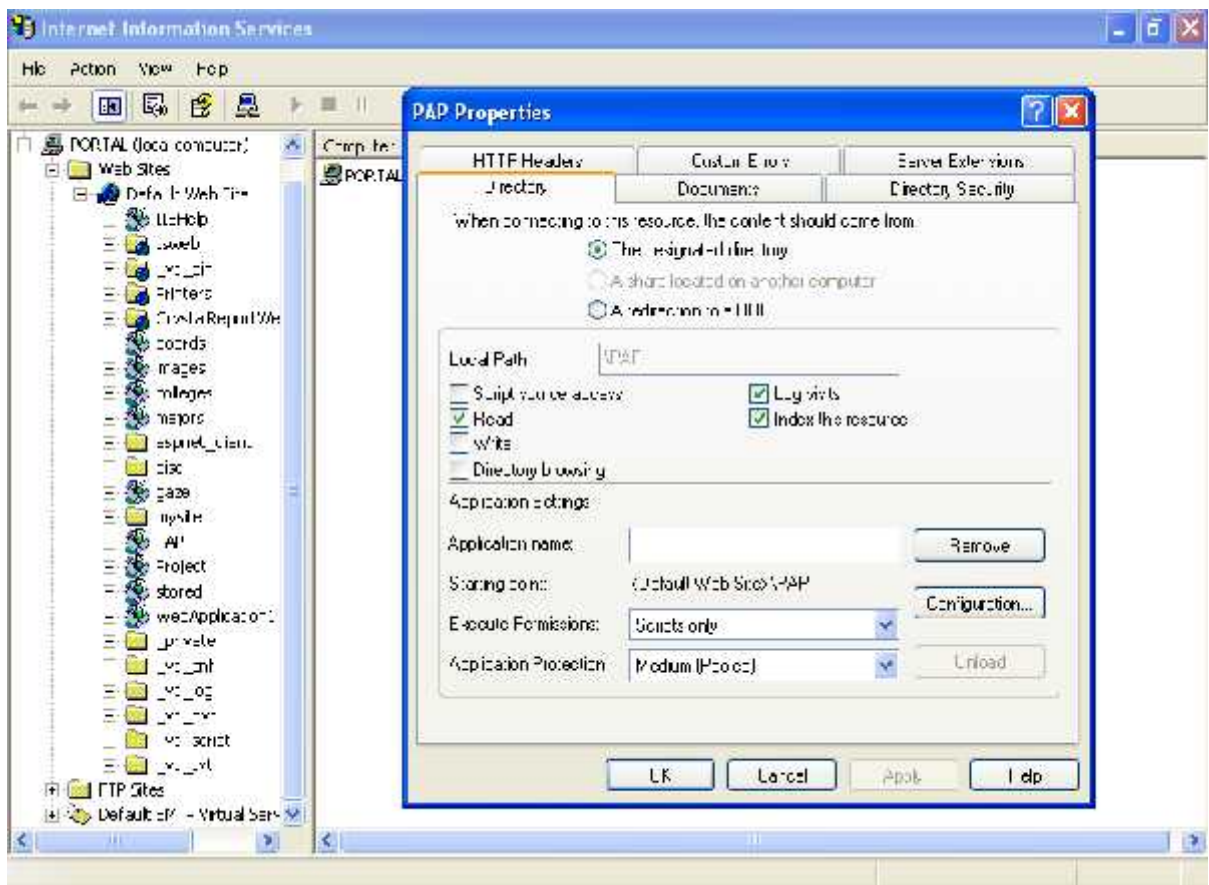


Figure (6.3) IIS Window



6.5 SQL server 2000 maintenance

The most important part of the system is database, since it contain the data used in the system, so it must be very secure much as better, so we must keep a track with it by maintaining database by using specific techniques as we mention before like backup .

Main components in SQL server 2000 according our system

a) The database of " portal"

Which include all the system tables, views, and stored procedures.

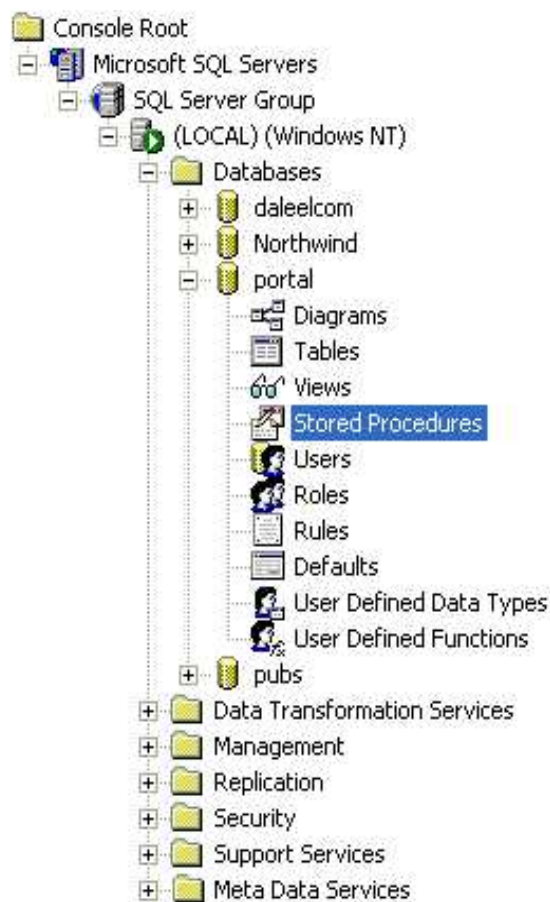


Figure (6.4) SQL server database



b) Security

This section include all roles that implemented on database, and users with their privileges allowed for them, and you can personalize your system setting by creating new users and roles and give each user specific privileges to allow access on database .

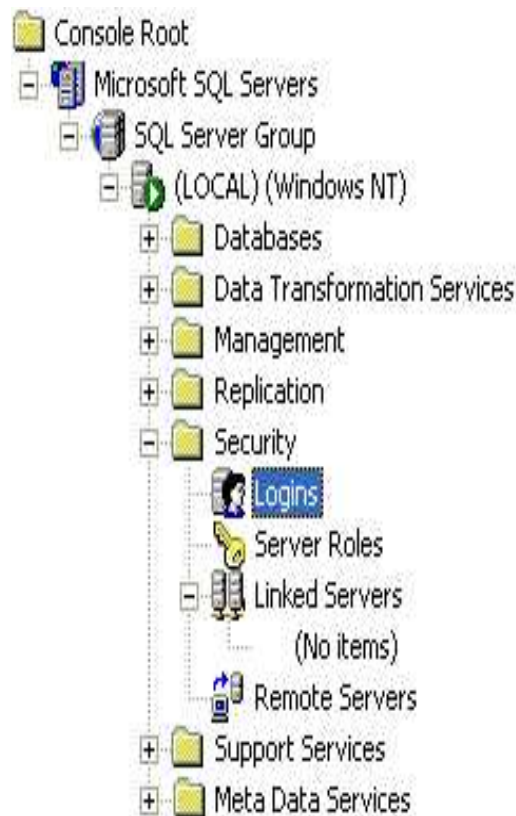


Figure (6.5) SQL server 2000 security section



6.6 The .NET Framework maintenance

.NET framework is an infrastructure of the Microsoft .NET technology, which we use to development our system one of it's components (ASP.NET).

Visual studio.NET collect all project files in one consol window known as solution explorer, through this window administrator can pick any file and make any justification on it or updated it.

If any error or problem was occur and cause system failure or it damage the visual studio.NET program, we can repair this problem by inserting the .NET framework CD and pick a repair choice , and the system will be repaired, or we can install new refresh copy from .NET framework.



7.1 Conclusions and recommendations:

7.1.1 Conclusions:

- The Portal system is subject to enhance and upgrading according to students and instructors opinions and feedback .
- Systems such as Portals must provide suitable and strong interface to attract attention and minimize user potential errors by providing appropriate validation messages to keep track of user actions.
- Portal system must be highly secure to reject any evil and hacking attempts.
- The Portal system is subject to the expansion of new ideas and enrichment features such as new services.
- We consider our system as a dynamic system, so it should response as a real time system.

7.1.2 Skills:

- We trained how to work as a team.
- We have take a good chance to deal with several developing programs such as:
 - ASP.NET Technology including Visual Studio.Net using Visual Basic.Net.
 - SQL Server 2000, that include creating tables, establish their relationships, apply constraints and connect it with web based applications.
 - Multimedia programs including adobe Photoshop, photoimpact and flash .
 - Microsoft office programs including Visio and ms-project.
- We have a good knowledge with how to engineer the software system in its different stages.



7.2 Recommendations and future work

7.2.1 Recommendations :

- ❖ We recommend to connect our Portal System with registration database, thus we work on a virtual database to some extent similar to real registration database, because we can't obtain the real skeleton of the registration database.
- ❖ We recommend to expand our system to include e-learning system and e-registration system to serve the students as possible.
- ❖ In addition to that, we recommend to publish our system on ppu.edu domain.
- ❖ We strongly recommend to build e-library database system and connect it with portal, in order to simplify student search transactions and reading.
- ❖ Build university forum that create virtual interactive area for only university students to exchange their opinions and ideas about given problem.
- ❖ Provide students with available accommodations .

7.2.2 Future Work:

- ❖ Multiple language interface supported by Portal (English, Arabic).
- ❖ Online help for potential problems that students might face.
- ❖ Strengthen the security during student Portal connections, by using more encryption algorithm techniques.
- ❖ Build e-magazine that receive participations only from university students.
- ❖ Build e-library for programs that serve different students in different majors.



7.3 References:

1. Sommerville, Ian, *Software Engineering*, 7th edition, Addison-Wesley, 2001.
2. Microsoft SQL server 2003 Help Topics.
3. Microsoft .NET Framework SDK v1.1.
4. www.asp.net/forums.

Conclusions:

- The Portal system is subject to enhance and upgrading according to students and instructors opinions and feedback .
- Systems such as Portals must provide suitable le and strong interface to attract attention and minimize user potential errors by providing appropriate validation messages to keep track of user actions.
- Portal system must be highly secure to reject any evil and hacking attempts.
- The Portal system is subject to the expansion of new ideas and enrichment features such as new services.
- We consider our system as a dynamic system, so it should response as a real time system.

Skills:

- We trained how to work as a team.
- We have take a good chance to deal with several developing programs such as:
 - ASP.NET Technology including Visual Studio.Net using Visual Basic.Net.
 - SQL Server 2000, that include creating tables, establish their relationships, apply constraints and connect it with web based applications.
 - Multimedia programs including adobe Photoshop, photoimpact and flash .
 - Microsoft office programs including Visio and ms-project.
- We have a good knowledge with how to engineer the software system in its different stages.



1. *Students:*

Field	Data Type	Required	Key	References	Description
Std_no	Numeric(9)	Yes	PK		Student Number
Std_fname	Varchar(50)	Yes			Student first name
Std_sname	Varchar(50)	Yes			Student second name
Std_tname	Varchar(50)	Yes			Student third name
Std_lname	Varchar(50)	Yes			Student family name
Birth_d	Nvarchar(50)	No			Student birth date
Birth_p	Varchar(50)	No			Student birth place
nationality	Varchar(50)	No			Student nationality
Social_s	Varchar(50)	No			Student social status
Std_id	Numeric(9)	Yes			Student identity number
Tawj_avg	Float(8)	Yes			Student taw. Avg.
Sex	Varchar(50)	No			Student gender
Address	Varchar(50)	Yes			Student address
Tel_no	Numeric(9)	Yes			Student tel. no.
Mjr_no	Numeric(9)	Yes	FK	Majors(mjr_no)	Student major number
Ac_no	Numeric(9)	Yes	FK	Academic_status(ac_no)	Student Academic status number
Ta_no	Numeric(9)	Yes	FK	Taw_branch(ta_no)	Student tawjihi number
pic	Nvarchar(50)	Yes			Student picture

Table (3.1) Students Table.

**2. Academic status:**

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Ac_no	Numeric (9)	Yes			Academic number
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Warning	Varchar (50)	No			Number of warnings
dismiss	Varchar (50)	No			Dismissed or not
Deelay	Varchar (50)	No			Delayed or not
Std_level	Numeric(9)	Yes			Student level
Regular	Varchar (50)	No			Regular or not
Graduated	Varchar (50)	No			Graduated or not
Hours_reg	Numeric (9)	Yes			Number of registered hours
Cwh_p	Numeric (9)	No			Community work hours
Sem_no	Nvarchar (50)	Yes			Semester number
Hours_p	Numeric (9)	No			Number of passed hours
Ac_year	Numeric (9)	Yes			Academic year
St_no	Numeric (9)	Yes	FK	Study_system (st_no)	
Std_avg	Float (8)	No			
Hours_p	Numeric (9)	Yes			

Table (3.2) Academic status Table.

3. Boards:

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>key</i>	<i>References</i>	<i>Description</i>
brd_no	Numeric (9)	Yes	PK		Board number
Brd_date	Datetime(9)	Yes			Board issue date
Due_to	Datetime(9)	Yes			Board due to date
Bodyt	Varchar (50)	No			Text body
Bodyb	Varchar (50)	No			Attachment body
Aauther	Varchar (50)	No			Aauther name
Subject	Varchar (50)	No			Title of the board

Table (3.3) Boards Table.



4. Colleges Boards:

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Col_no	Numeric (9)	Yes	PK		College number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.4) College boards Table.

5. Departments boards:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Dpt_no	Numeric (9)	Yes	PK		Department number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.5) Departments boards Table.

6. Courses Boards:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
crs_no	Numeric (9)	Yes	PK		Course number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.6) Courses boards Table.

7. Students Boards:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
std_no	Numeric (9)	Yes	PK		Student number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.7) Students boards Table.

8. Colleges:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Col_no	Numeric (9)	Yes	PK		College number
Coll_name	Varchar (50)	Yes			College name
descr	Nvarchar (500)	No			College description

Table (3.8) Colleges boards Table.

**9. Courses:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
crs_no	Numeric (9)	Yes	PK		Course number
Crs_name	Varchar (50)	Yes			Course name
Crđ_hours	Numeric (9)	Yes			Course credit hours
descr	Nvarchar (500)	No			Course description

Table (3.9) Courses Table.

10. Courses_type:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Co_no	Numeric (9)	Yes	PK		Course number
Co_t	Varchar (50)	Yes			Course type

Table (3.10) Course_Type Table.

11. Passed courses:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_Courses(crs_no)	Course number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes	PK		Academic year
Class_no	Numeric (9)	Yes			Section number
Mark	Float (8)	No			Course mark
Co_no	Numeric (9)	Yes	FK	Courses_type(co_no)	Course number

Table (3.11) Passed courses Table.

**12. Offered_courses:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	courses(crs_no)	Course number
Class_no	Numeric (9)	Yes	PK		Section number
Time	Nvarchar (50)	Yes			Lecture time
Room_no	Numeric (9)	No			Room number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes			Academic year

Table (3.12) Offered courses Table.

13. Major courses:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	Courses(crs_no)	Course number
Mjr_no	Numeric (9)	Yes	PK,FK	Majors(mjr_no)	Major number

Table (3.13) Major courses Table.

14. Departments:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Dpt_no	Numeric (9)	Yes	PK		Department number
Dpt_name	Varchar (50)	Yes			Department name
Descr	Nvarchar (500)	No			Department description
Col_no	Numeric (9)	Yes	FK	Colleges(col_no)	College number
Cwh_r	Numeric (9)	No	PK		Number of required community work hours
Ac_no	Numeric (9)	Yes	FK	Academic status(ac_no)	Academic year

Table (3.14) Departments Table.

**15. Employee classes:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK,FK	Employee(emp_no)	Employee number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_courses(crs_no)	Course number
Sem_no	Nvarchar (50)	No			Semester number
Class_no	Numeric (9)	Yes	Pk		Section number

Table (3.15) Employee classes Table.

16. Employee:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK		Employee number
Emp_name	Varchar (50)	Yes			Employee name
Dpt_no	Numeric (9)	No	FK	Departments(dpt_no)	Department number
St_no	Numeric (9)		FK	Study_system(st_no)	Study system (scientific degree)
Id_no	Numeric (9)	Yes			Identity number

Table (3.16) Employee Table.

**17. Financial status:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Fin_no	Numeric (9)	Yes			Financial account number
assistant	Float (8)	No			Assistant value
scholarship	Float (8)	No			Scholarship value
others	Float (8)	No			Other financial aids
credit	Float (8)	Yes			Student surplus balance
debit	Float (8)	Yes			Student deficit balance
Std_no	Numeric (9)	Yes	PK		Academic year

Table (3.16) Financial status Table.

18. Administrator login:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
username	Nvarchar (50)	Yes	PK		Administrator username (id)
pwd	Nvarchar (50)	Yes			Administrator password

Table (3.17) Administrator login Table.

19. Students login:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		Student number (id)
pwd	Varchar (50)	Yes			Student password

Table (3.18) Students login Table.

**20. Instructor login:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK		Employee number
pwd	Nvarchar (50)	Yes			Employee password

Table (3.19) Instructor login Table.

21. Majors:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Mjr_no	Numeric (9)	Yes	PK		Major number
Mjr_name	Varchar (50)	Yes			Major name
Hours	Numeric (9)	Yes			Major credit hours number
Descr	Nvarchar (500)	No			Major description
Dpt_no	Numeric (9)	Yes	FK	Departments(dpt_no)	Department number
outline	Numeric (9)	No			Major outline

Table (3.20) Majors Table.

22. Projects:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		student number
Proj_no	Int (4)	Yes			Project number
Proj_name	Varchar (50)	Yes			Project name
Proj_body	Nvarchar (500)	Yes			Project body
Emp_no	Numeric (9)	Yes	FK	employee(emp_no)	Employee number

Table (3.21) Projects Table.



23. Setting:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Set_no	Int (4)	Yes	PK		Setting account number
Set_s	Int (4)	Yes			Setting status value
Set_date	Datetime (8)	No			Setting deadline date

Table (3.22) Setting Table.

24. Student style:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		student number
Main_header	Varchar (50)	Yes			Main header image path
Name_header	Varchar (50)	Yes			Name header image path
Title_header	Varchar (50)	Yes			Title header image path
Box	Varchar (50)	Yes			Box image path
Center_box	Varchar (50)	Yes			Center box image path
Box_header	Varchar (50)	Yes			Box header image path
Centerbox_header	Varchar (50)	Yes			Center box header image path
logout	Varchar (50)	Yes			Logout image path

Table (3.23) Student style Table.

**25. Study system:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
St_no	Numeric (9)	Yes	PK		Study system number
St_name	Varchar (50)	Yes			Study system name

Table (3.24) Study system Table.

26. tawjihi branch:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Ta_no	Numeric (9)	Yes	PK		Taw. number
Ta_name	Varchar (50)	Yes			Taw. branch

Table (3.25) Tawjihi branch Table.

27. Vote:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Vote_no	Numeric (9)	Yes	PK		Vote number
Vote_body	Varchar (500)	Yes			Vote body text

Table (3.26) Vote Table.

28. Vote result:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Vote_no	Numeric (9)	Yes	PK,FK	Vote(vote_no)	Vote number
V_yes	Int (4)	No			Vote yes value (1)
V_no	Int (4)	No			Vote no value (1)
V_maybe	Int (4)	No			Vote maybe value (1)
who	Numeric (9)	Yes	PK	employee(emp_no)	Userid for who voted

Table (3.27) Vote result Table.



3.5 Test Plan:

Here we describe briefly the methodology that we have adapted to test the system, steps that will be followed in the system testing are described below:

Testing steps:

1- Unit and Module testing:

We will use the white and black test Box testing to ensure that each function or module will operate as expected, by inserting valid and invalid inputs to show how the system will handle it.

2- Sub-system testing:

In this stage we will test each sub-system individually to ensure that each sub-system is operates as expected and meet its requirements.

3- Integration testing:

The integration of all sub-systems will be tested so that to ensure that the subsystems work together properly as expected, and meets its requirements.

4- System testing:

The system with all subsystems and functions will be tested to ensure that it functions properly, it meets its specification, and show if there defects throw system running.



3.6 Programming Language and Coding

There are many languages that can be used to develop a system such ours, but the most two effective languages are JAVA language and ASP.NET using Microsoft Visual Studio .NET as the development tool, here we describe the advantages of each, and why our selection was on the ASP.NET (we describe the ASP.NET technology in details in chapter 1) :

1- *Device Independent:*

Both Java and ASP.Net are a device independent languages, this means that any user can open the page that written using theses two languages from any explorer without any additional components or drivers, because he will just receive an HTML code.

2- *Security*

Java and ASP.Net have a high level of security during transmitting data; they provide us with many algorithms and techniques.

In the ASP.NET there are a build in valuator that ensure the user's input before any generation on the server, so if there any unusual input the webpage it self will not return to the server.

In Java the programmer must do all algorithms and validations manually.

3- *Server side*

One of the most powerful advantages of the ASP.NET technology is that it do not need to make any efforts on the client side, all operations and functions will work on the server.



3.1 Introduction

This chapter describes the system design that have objects design for each module and the topics in this chapter is: (description, interface, flowchart, constraints or validation, and user interface design) , I/O design for designing all input /output screens , and database design show database model and all table constraints, test plan , and program languages and coding .

3.2 Functional design

functional design for each module should described in the software system, accordingly with the description of the interface, the constraints, and the user interface design in which we use means of diagramming to help us understand it:

These functions design divided to:

1. Student Functions design
2. Administrator Functions design
3. Instructor Functions design
4. Public Function design

A . Student Functions Design :



1- *login student:*

a. Description : This function that gives the students authentication or enabling them to login to them accounts .

b. Interface:

- ◆ Input : student ID (Student number), student Password
- ◆ Output : Student home page if your id and password is true , error message if id or password is invalid.

c. Constraints:

- ◆ Only students and administrator are authenticated to login to this page.
- ◆ Password must be checked. and it considered as a character case
- ◆ Password must be at least 6 characters.
- ◆ Username must be numbers only (student academic number).



d. Flowchart:

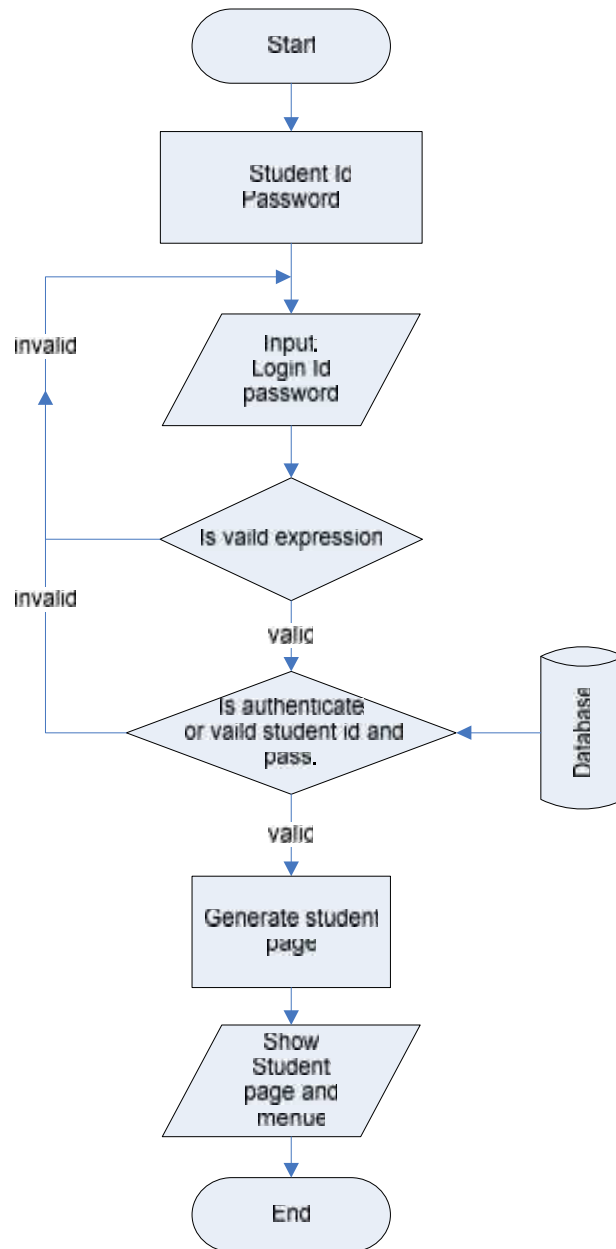


Figure (3.1) Student login Operation.



>> User interface design

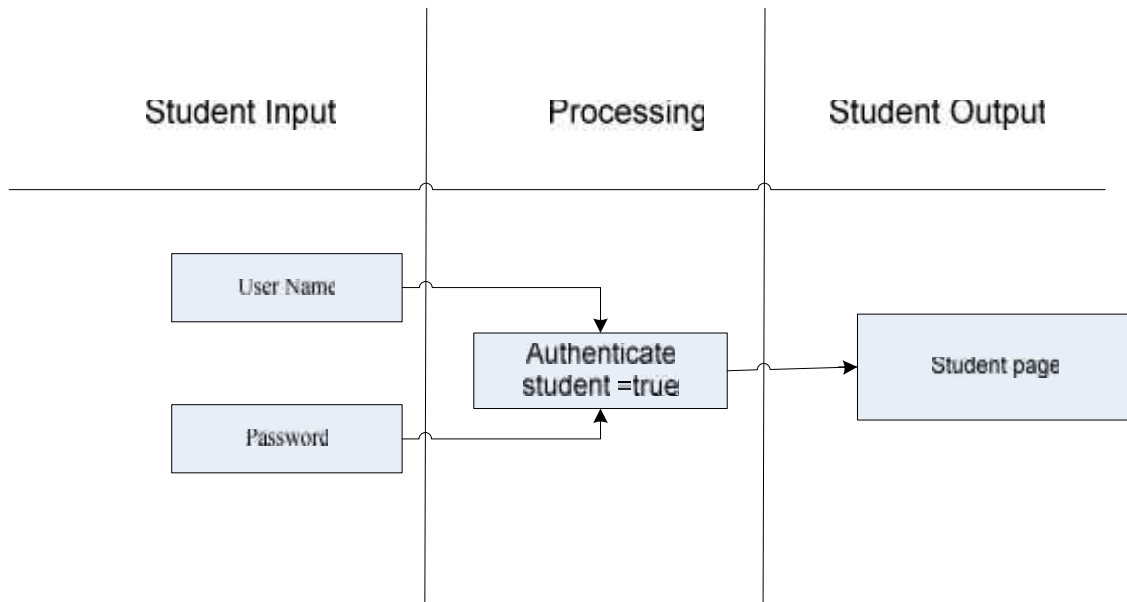


Figure (3.2) Student Login Interface Design.

2- *Forgot my Password:*

- a. Description :This function is important and allow the student to retrieve his password if he forgot it, and the student can get his account after the system checked his username and then asked he some questions that must be answered correctly.

b. Interface:

- ◆ Input: student username, to ensure that he has an account on PAP.
- ◆ Input : if the previous username is valid then he will be asked about : student id card Number , Tawjihi Average, Accumulative Average, Birth Date



- ◆ Output : if he answered all questions correctly the system will send the student account to his e-mail, but in our system he will show his account (username, password, group) because there is no connection yet with university mail server, else s\error message will be appear.

c. Constraints:

- ◆ Student id card number must be numbers and 9 digit.
- ◆ Accumulative average and tawjihi average must be numbers.
- ◆ Birh date format (month, day, year).



d. Flowchart:

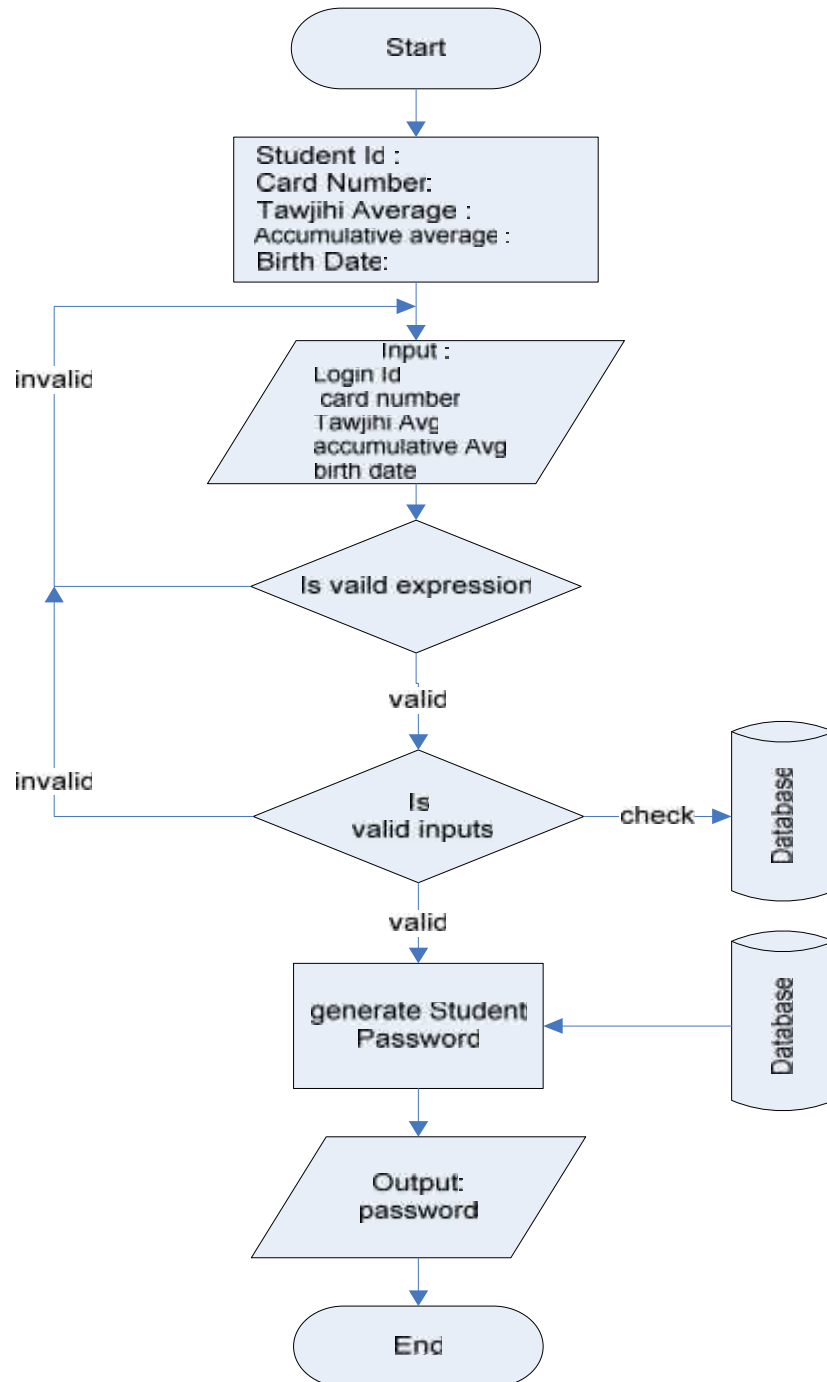


Figure (3.3) Forgot My password Operation.



>> User interface design

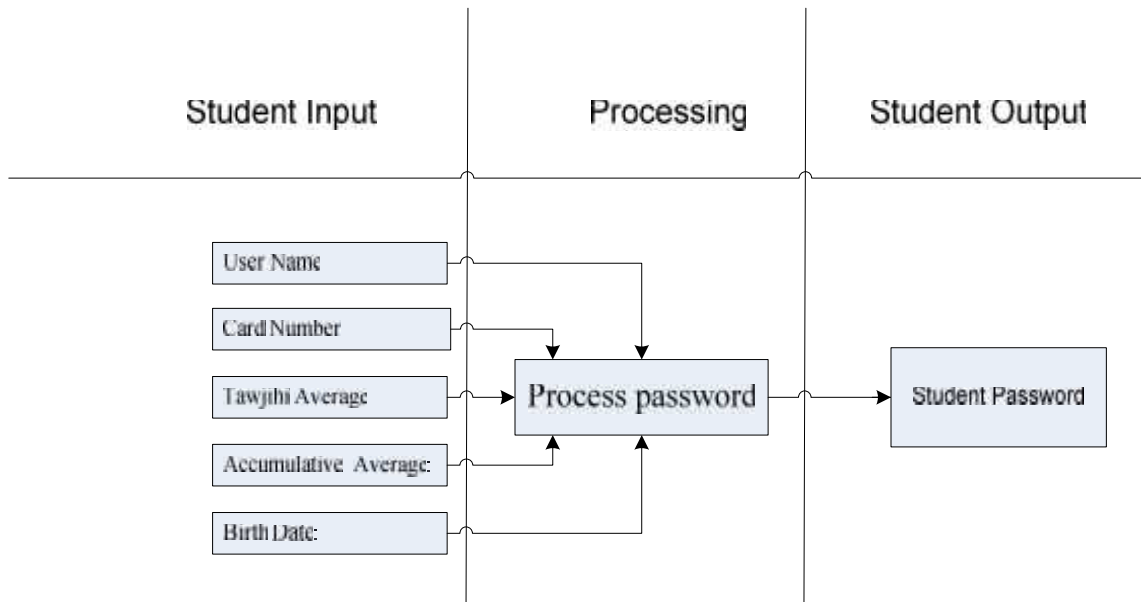


Figure (3.4) Forgot Password Interface Design.

3- Student Profile :

a. Description :This function show the student profile in the university that have name, address of student ,and academic information, and there is an area that allow to student to insert him picture if he want, and if he was graduated he will be able to insert his graduate project document to portal. This function dependent on the student id (username), the session take student Id (username) and show student profile according it .

b. Interface:

- ◆ Input : Session(student Id)
- ◆ Output : Full Student Name, Address, Tel. No., Academic No, Scientific Degree , Major, Outline.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information displayed is read only and the student can't edit it, just his picture and document can edit them.

d. Flowchart:

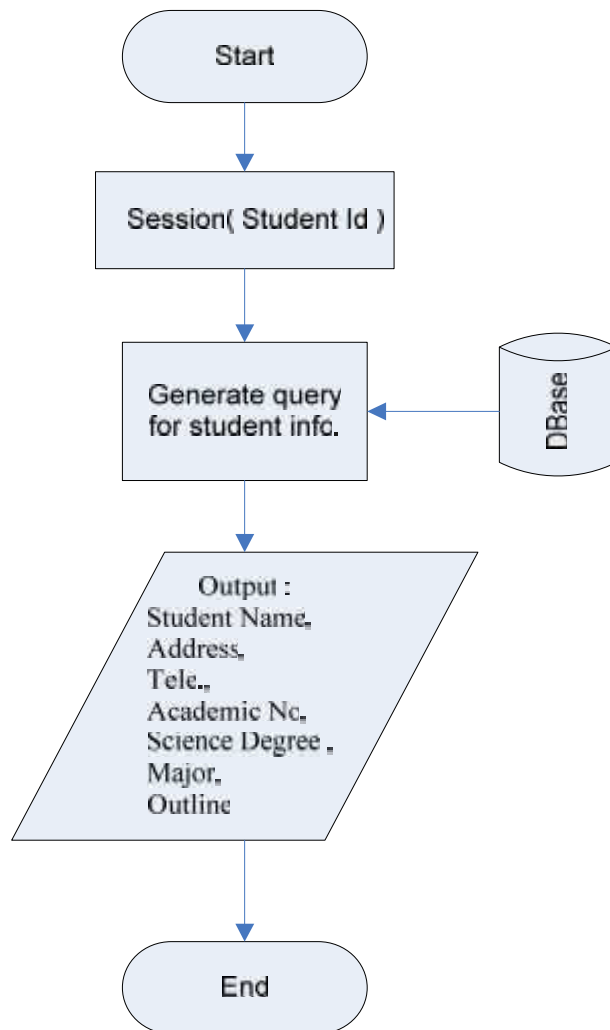


Figure (3.5) Student profile Operation.



>> User interface design

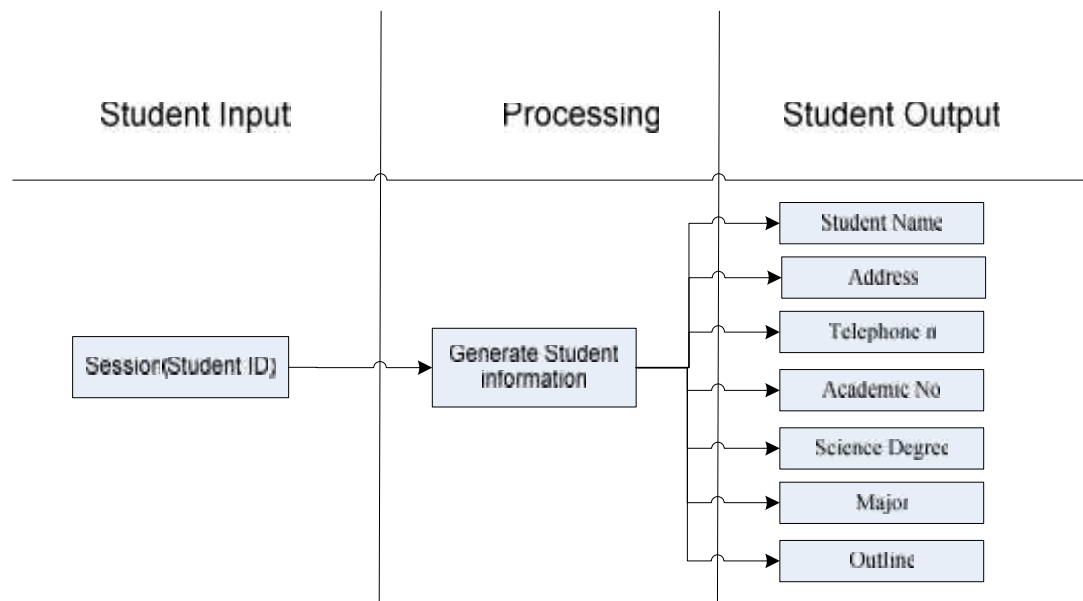


Figure (3.6) Student Profile Interface Design.

4- *Current Schedule* :

- a. **Description** :This function show the student Current Schedule in the university that have Course No., Semester, Course Name, Instructor, Section, Time, Room No. This function dependent on the student id (username), the session take student Id and show current schedule according it .

- b. **Interface**:

- ◆ **Input** : Session(student Id)
- ◆ **Output** : Course No., Semester, Course Name, Instructor, Section, Time, Room No.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ Student ser can't change any information (all information read only).

d. Flowchart:

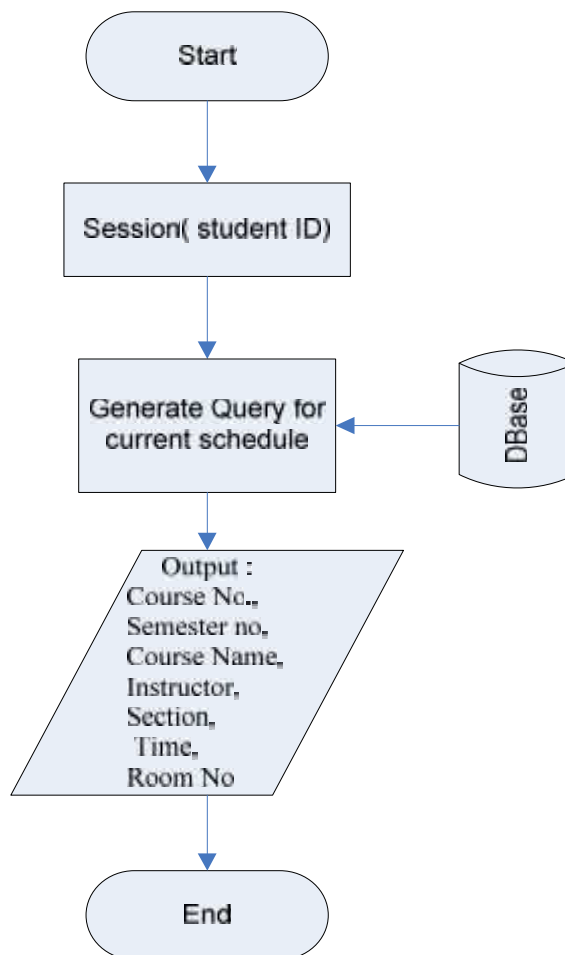


Figure (3.7) Student Current Schedule Operation.



>> User interface design

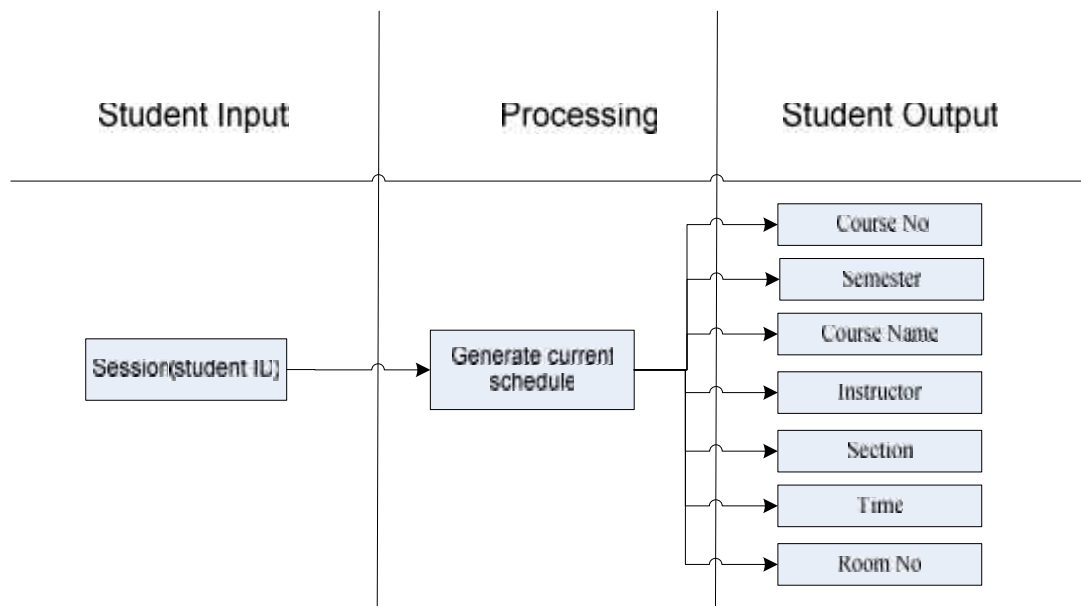


Figure (3.8) Student Current schedule Interface design.

5- Financial Balance:

a. Description :This function show the financial balance for student in the university that have Financial Aids (Assistance, Scholarship, Other) , Balance(Credit, Debit), Semester Balance and Total Balance.

b. Interface:

- ◆ Input : student Id in session ,semester number in session
- ◆ Output : Financial Aids (Assistance, Scholarship, Other), Balance(Credit, Debit), Semester Balance and Total Balance



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information here is read only.

d. Flowchart:

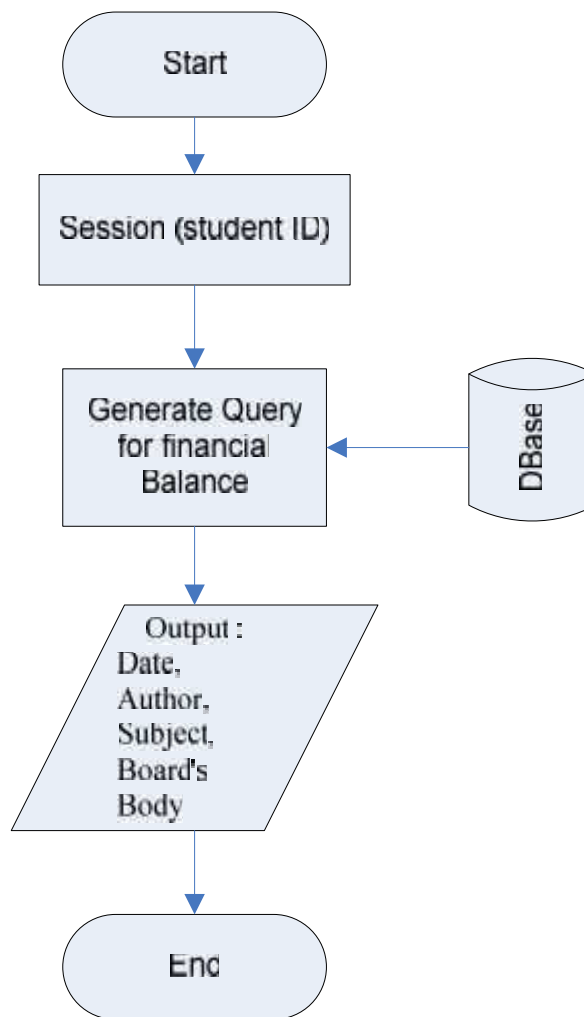


Figure (3.9) Student Financial Balance Operation.



>> User interface design

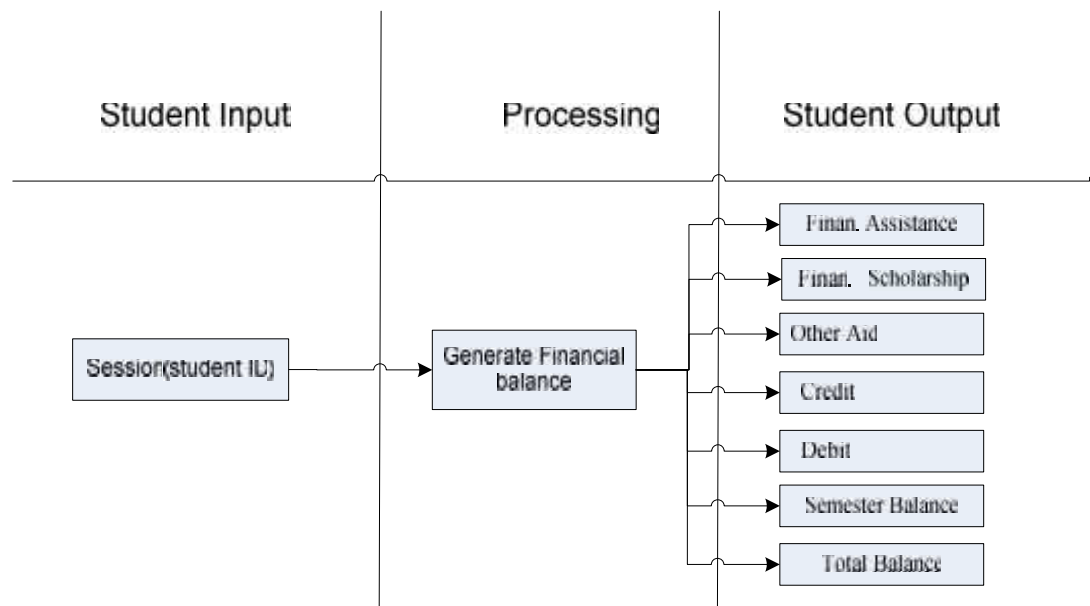


Figure (3.10) Student financial Balance Interface Design

6- Academic Status:

a. Description :This function show the academic status for each student in the university that have major degree, tawjihi branch and average, community work hours (passed and remains) department name and college name, average accumulate and major, academic alert,

b. Interface:

- ◆ Input : Session(student ID)
- ◆ Output : major degree, tawjihi branch, tawjihi average, community work hours (passed and remains), department name and college name, average accumulate and average major, academic alert, Dismiss, Delay, Level, Regular, Graduated.



c. Constraints:

- ◆ Authenticated student can only show this page.
- ◆ All information here is read only.

d. Flowchart:

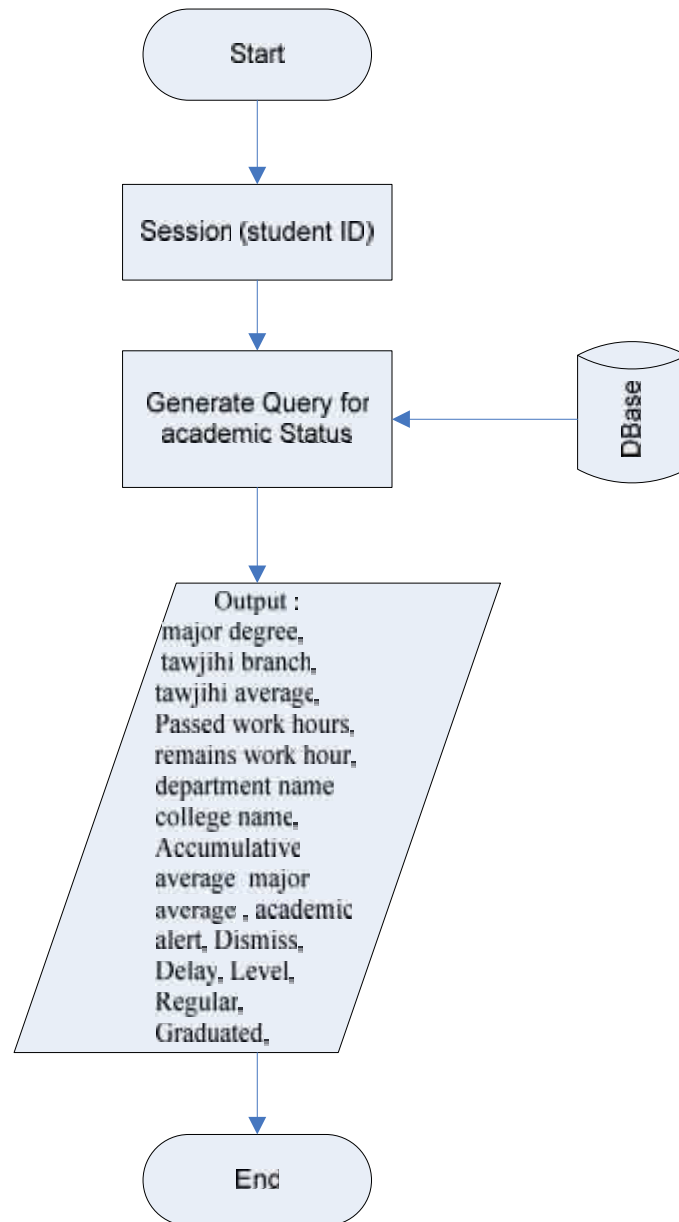


Figure (3.11) Student academic status



>> User interface design

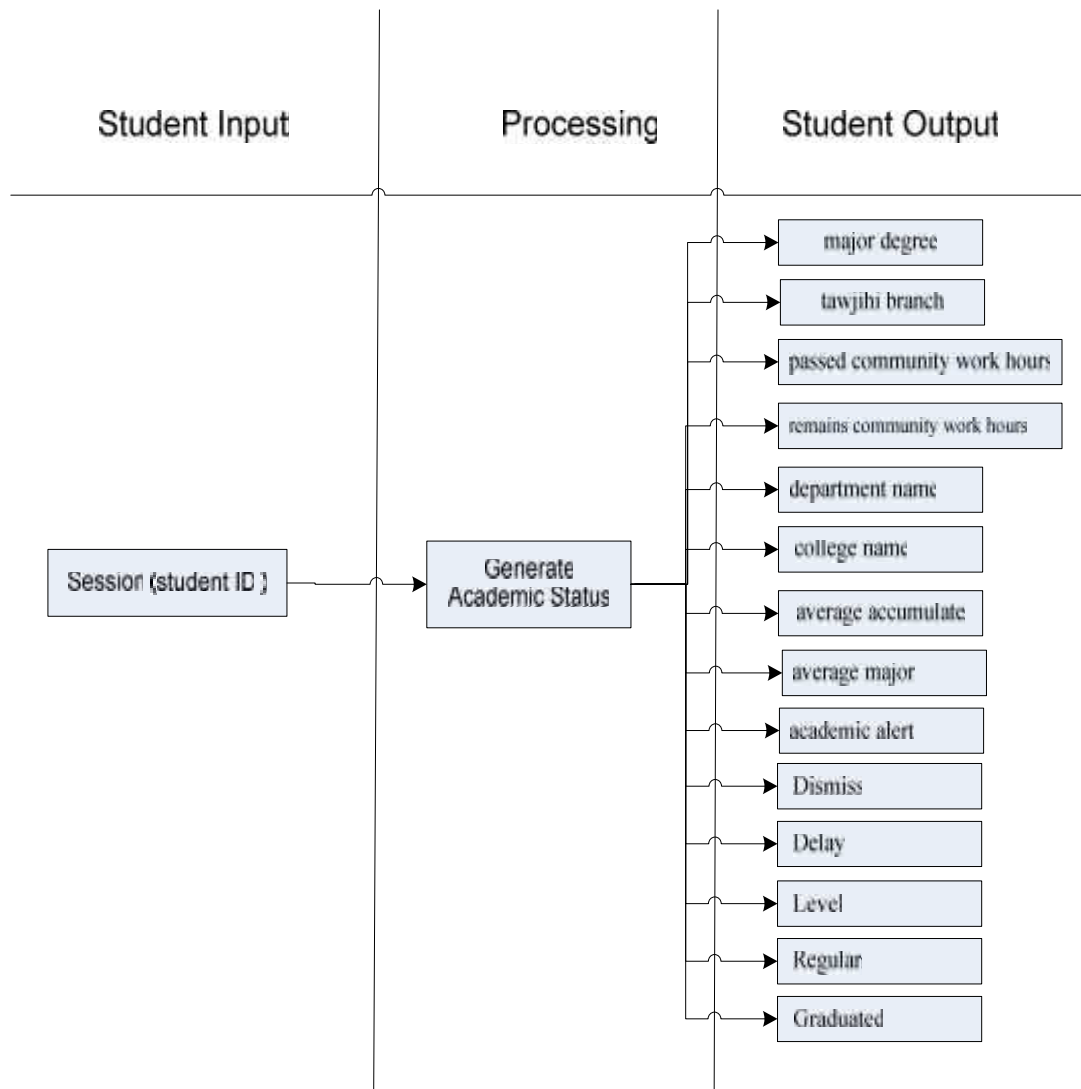


Figure (3.12) Student academic Status Interface Design



7- *Semester Status:*

a. **Description :** This function shows the semester status for a registered student, and this page shows the finished semester marks, the semester average, hours registered and hours passed, and if the student has an honor or not according to his average.

b. **Interface:**

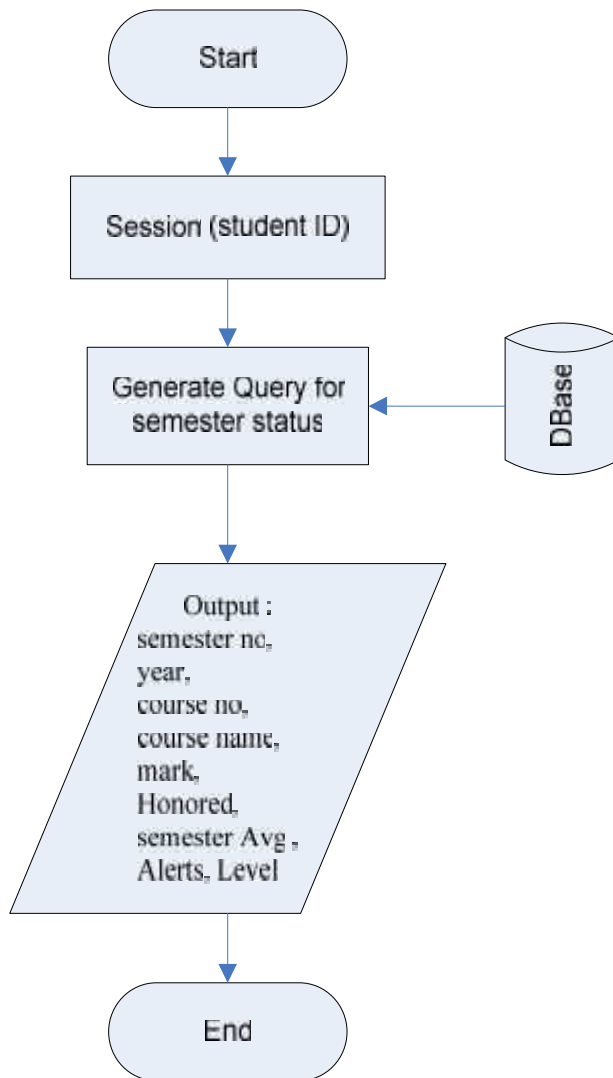
- ◆ **Input :** student Id in session, semester number in session.
- ◆ **Output :** semester number, year, course no, course name, mark, Honored, Semester average.

c. **Constraints:**

- ◆ Authenticated student can only show this page.
- ◆ All information is read only.



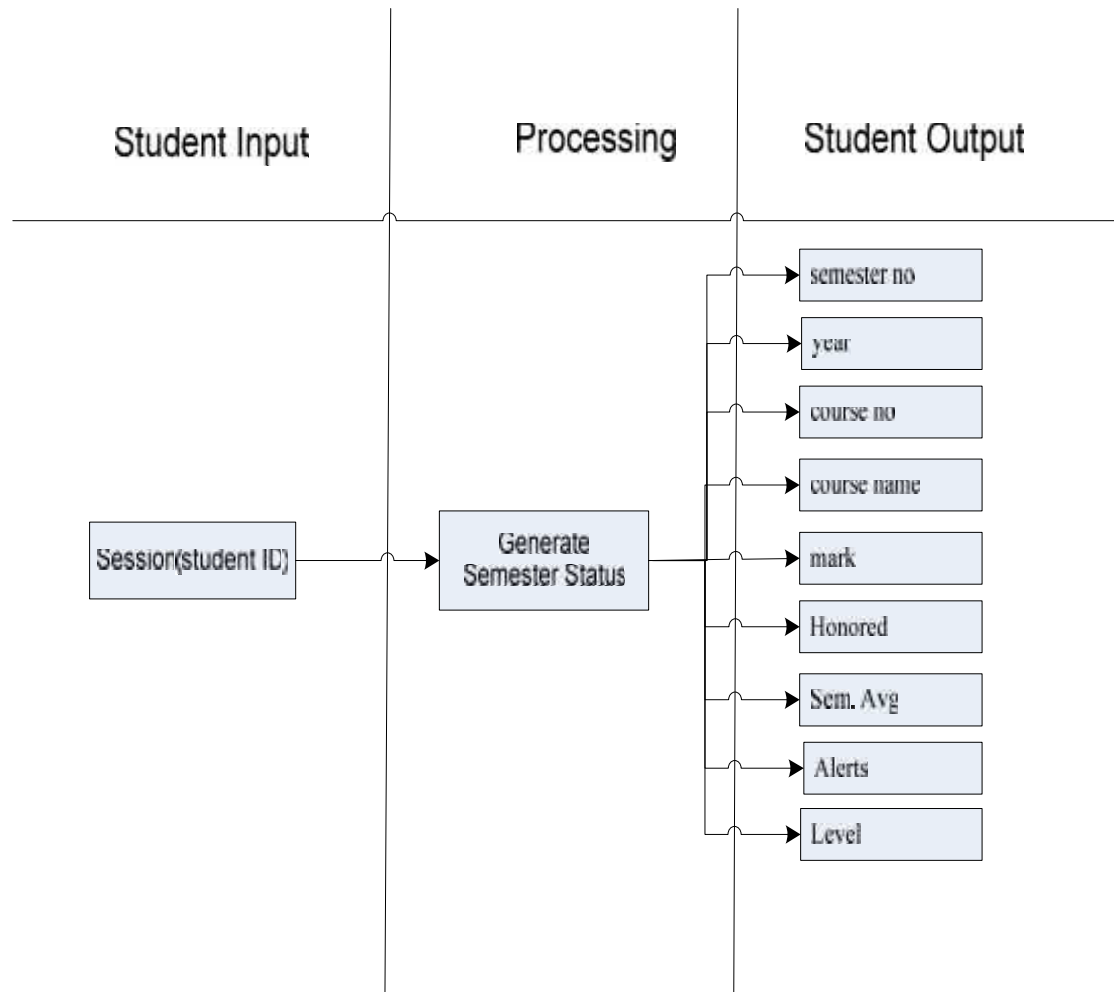
d. Flowchart:



Figure(3.13) student semester status operation.



>> User interface design



figure(3.14) student semester status interface design.

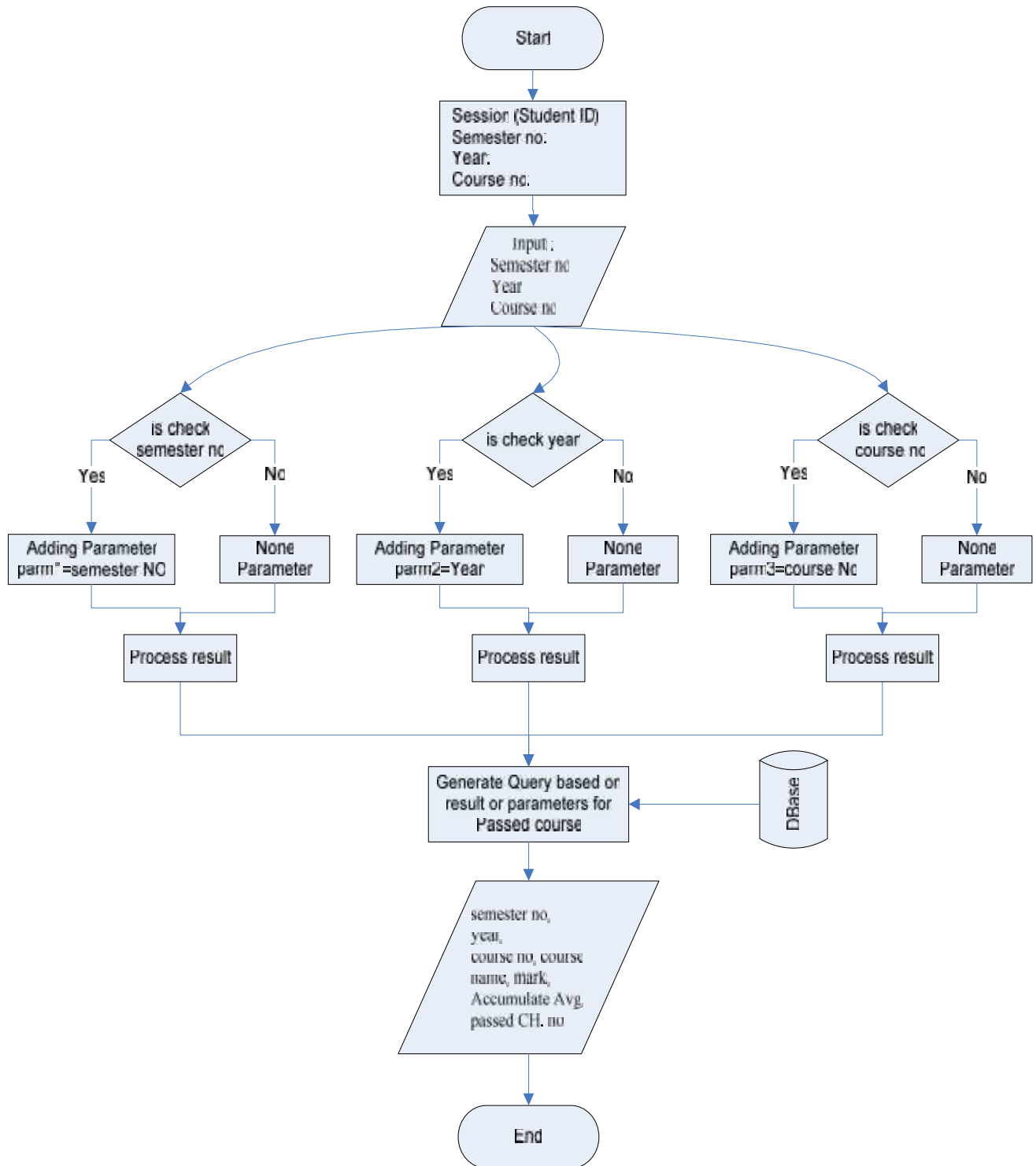


8- *Passed Courses:*

- a. **Description :** This function shows the finished hours and marks for the student, the student can filter this information according to semester, year, course type, and the student can show his accumulative average, number of passed hours, and number of remains, and he can show if he has completed the graduate requirements or not..
- b. **Interface:**
 - ◆ **Input :** Session(student Id) , semester number in session, year in session.
 - ◆ **Output :** semester number, year, course no, course name, course type, course mark, credit hours passed and remains number, accumulate avg..
- c. **Constraints:**
 - ◆ Authenticated student can only show this page..
 - ◆ All information is read only.



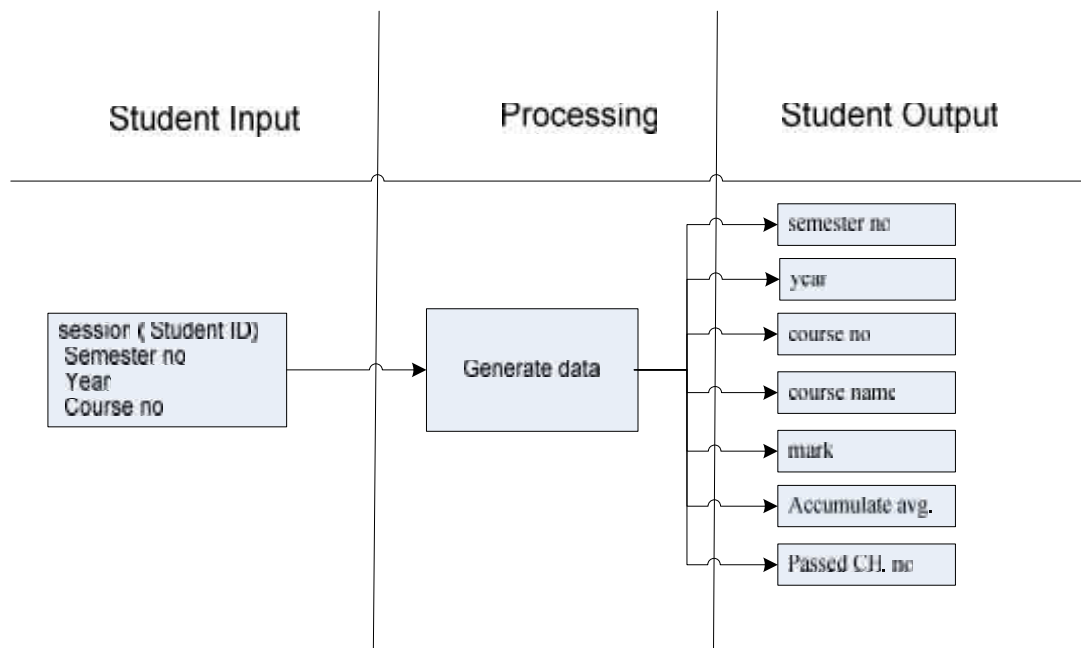
d. Flowchart:



Figure(3.15) passed courses operation.



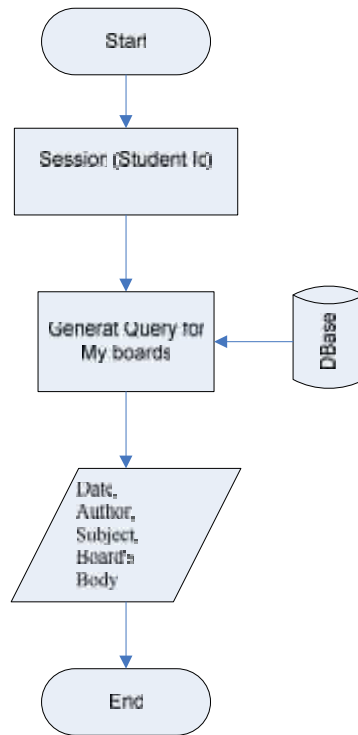
>> User interface design



Figure(3.16) passed courses interface design.

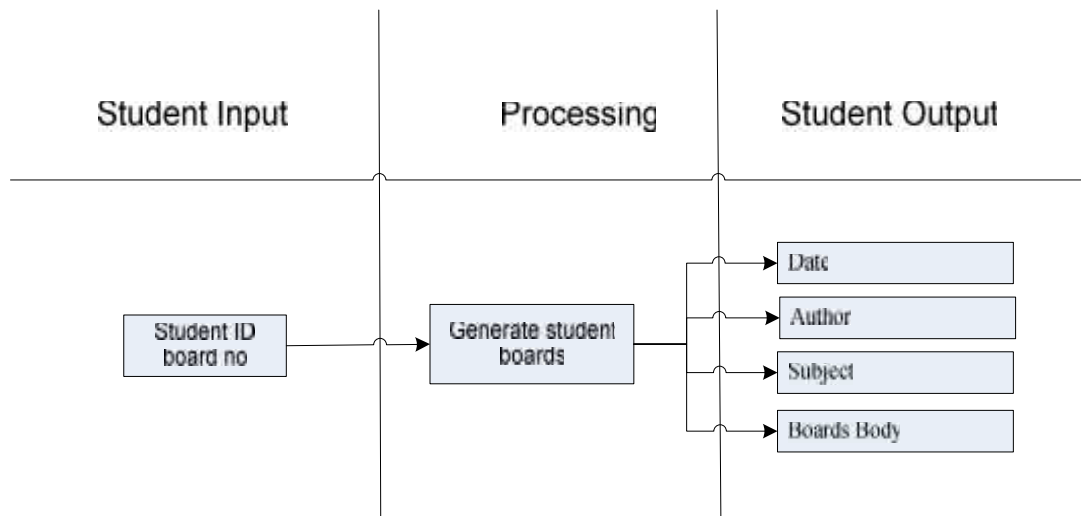
9- My Boards:

- a. Description :This function show the student boards only for current student
- b. Interface:
 - ◆ Input : Session (student ID)
 - ◆ Output : Date, Author, Subject, Board's Body
- c. Constraints:
 - ◆ Authenticated student only can show this page.
 - ◆ All information is read only.
- d. Flowchart:



Figure(3.17) My boards operation.

>> User interface design



Figure(3.18) My boards interface design.

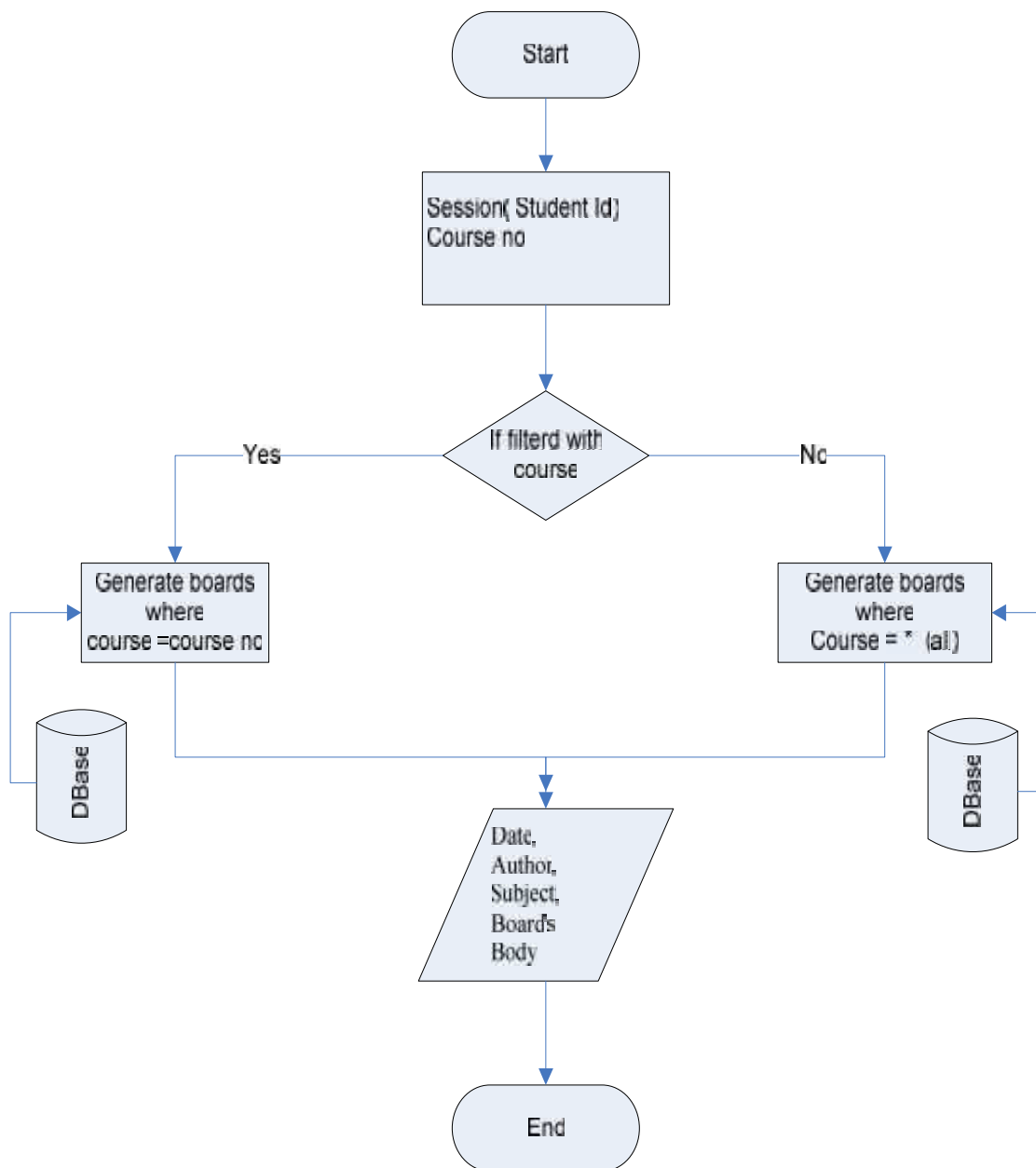


10- *Course Boards:*

- a. **Description :** This function shows the course boards for the current student courses .
- b. **Interface:**
 - ◆ **Input :** session(student Id), semester number in session, year in session.
 - ◆ **Output :** Date, Author, Subject, Board's Body
- c. **Constraints:**
 - ◆ **Authenticated student only can show this page.**
 - ◆ **All information is read only.**



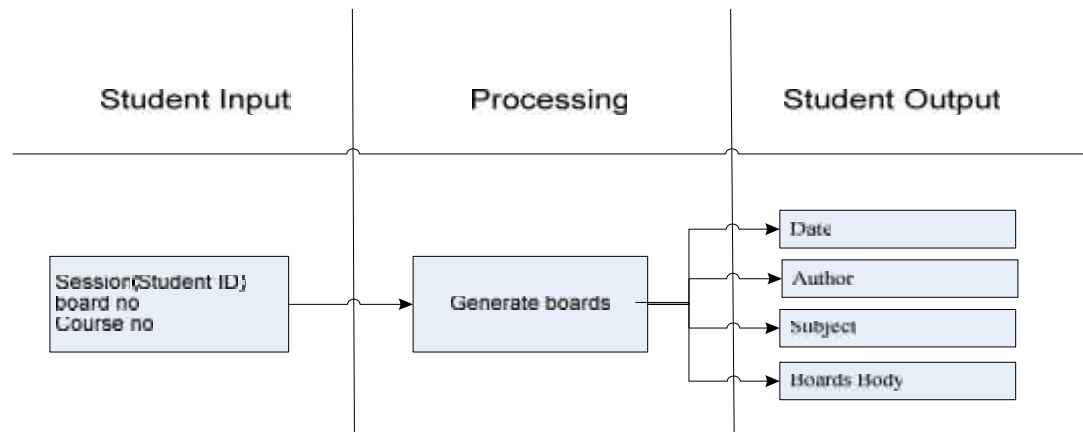
d. Flowchart:



Figure(3.19) Course boards operation.



>> User interface design



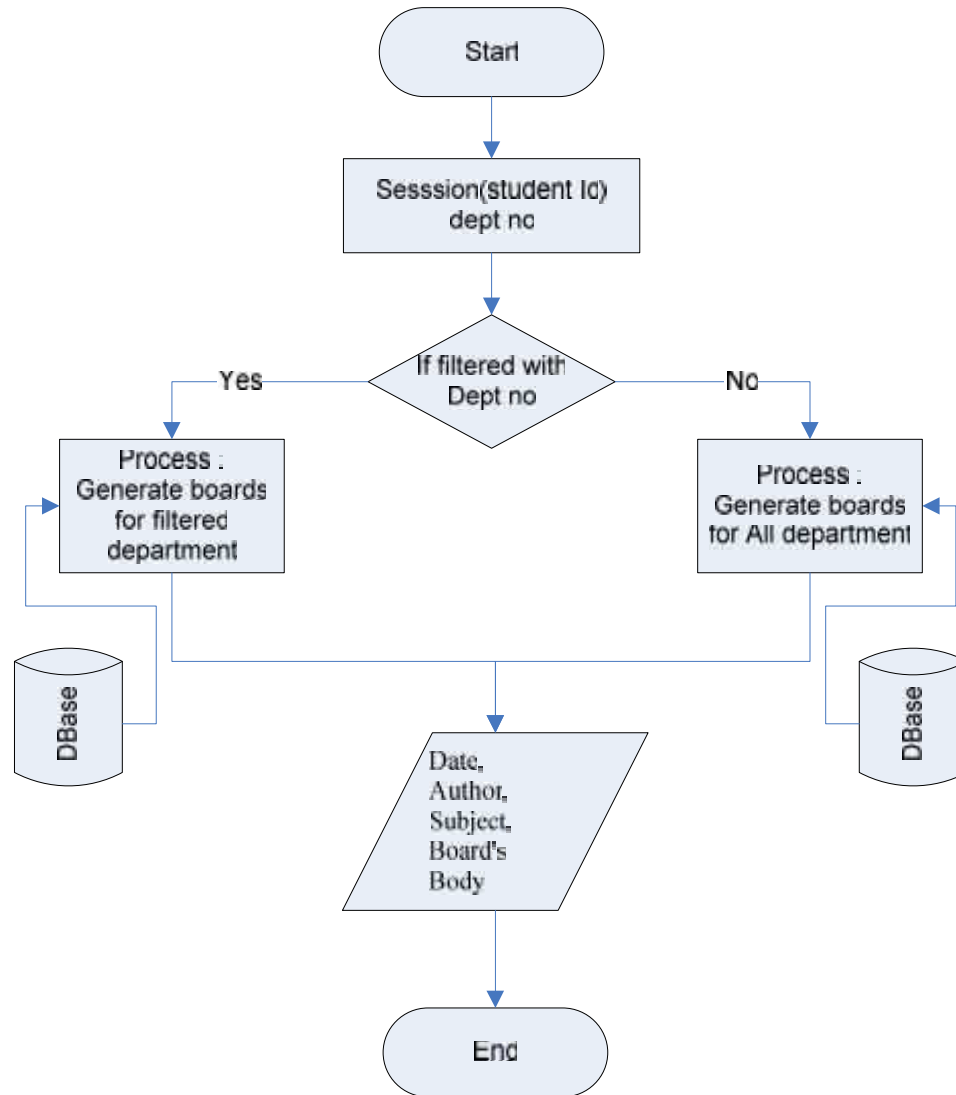
Figure(3.20) course boards interface design.

11- Department Boards:

- a. Description :This function show the department boards for the student in a specific department.
- b. Interface:
 - ◆ Input : session (student Id) , department number in session.
 - ◆ Output : Date, Author, Subject, Board's Body
- c. Constraints:
 - ◆ Authenticated student only can show this page.
 - ◆ All information is read only.



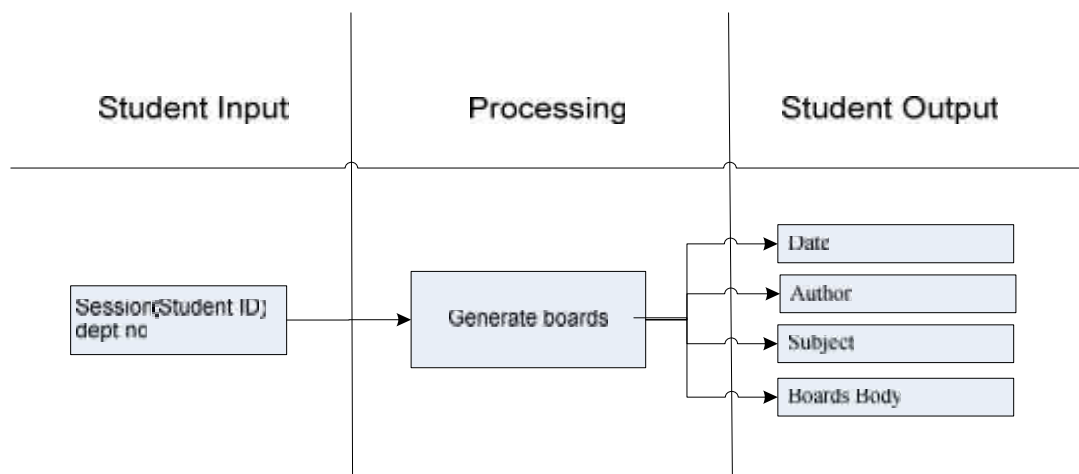
d. Flowchart:



Figure(3.21) Department boards operation.



>> User interface design



Figure(3.22) Department boards interface design.

12- College Boards:

a. description :

This function show the college boards for a student in specific college.

b. interface:

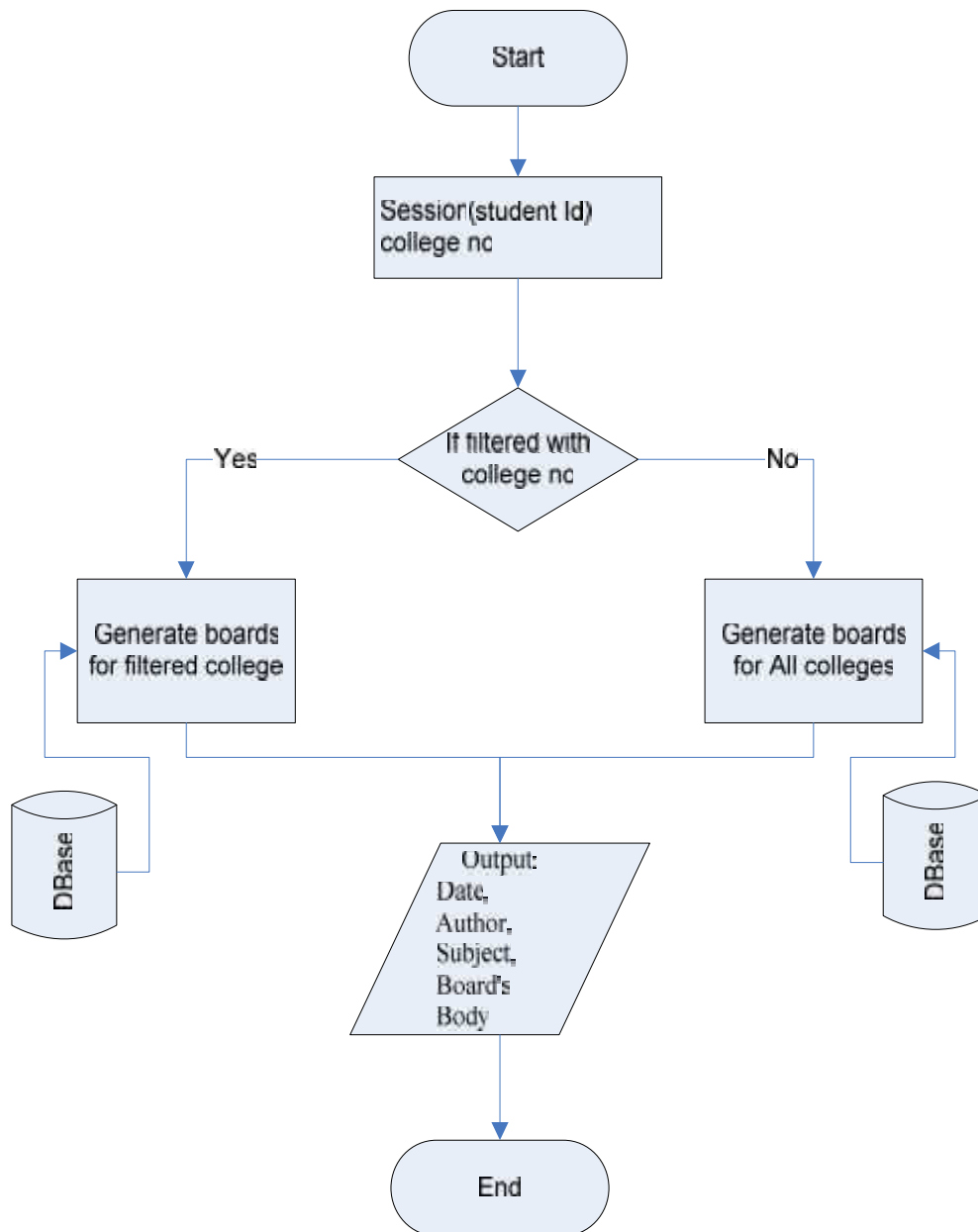
- ◆ Input : session(student Id) , college number in session
- ◆ Output : Date, Author, Subject, Board's Body

c. Constraints:

- ◆ Authenticated student only can show this page.
- ◆ All information is read only.



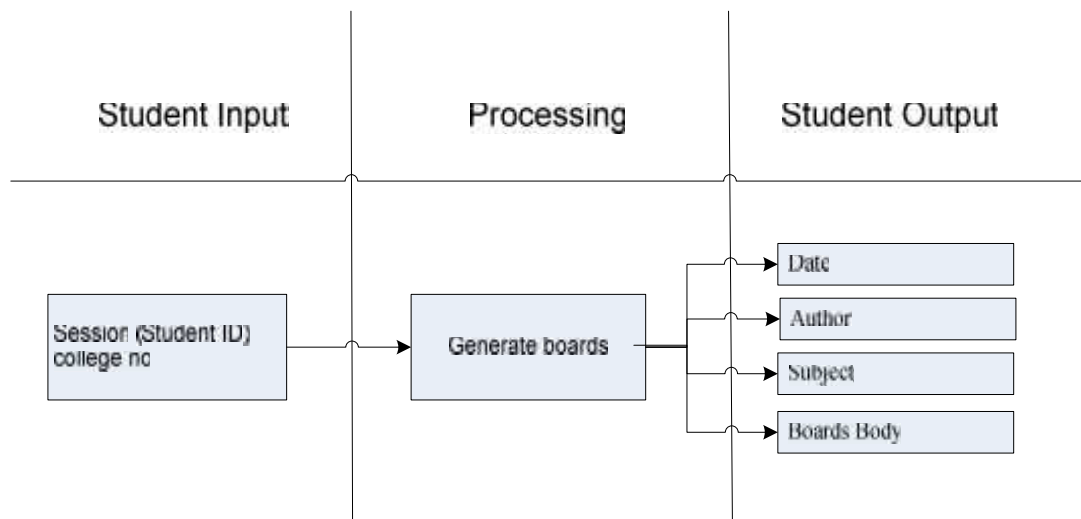
d. Flowchart:



Figure(3.23) College boards operation.



>> User interface design



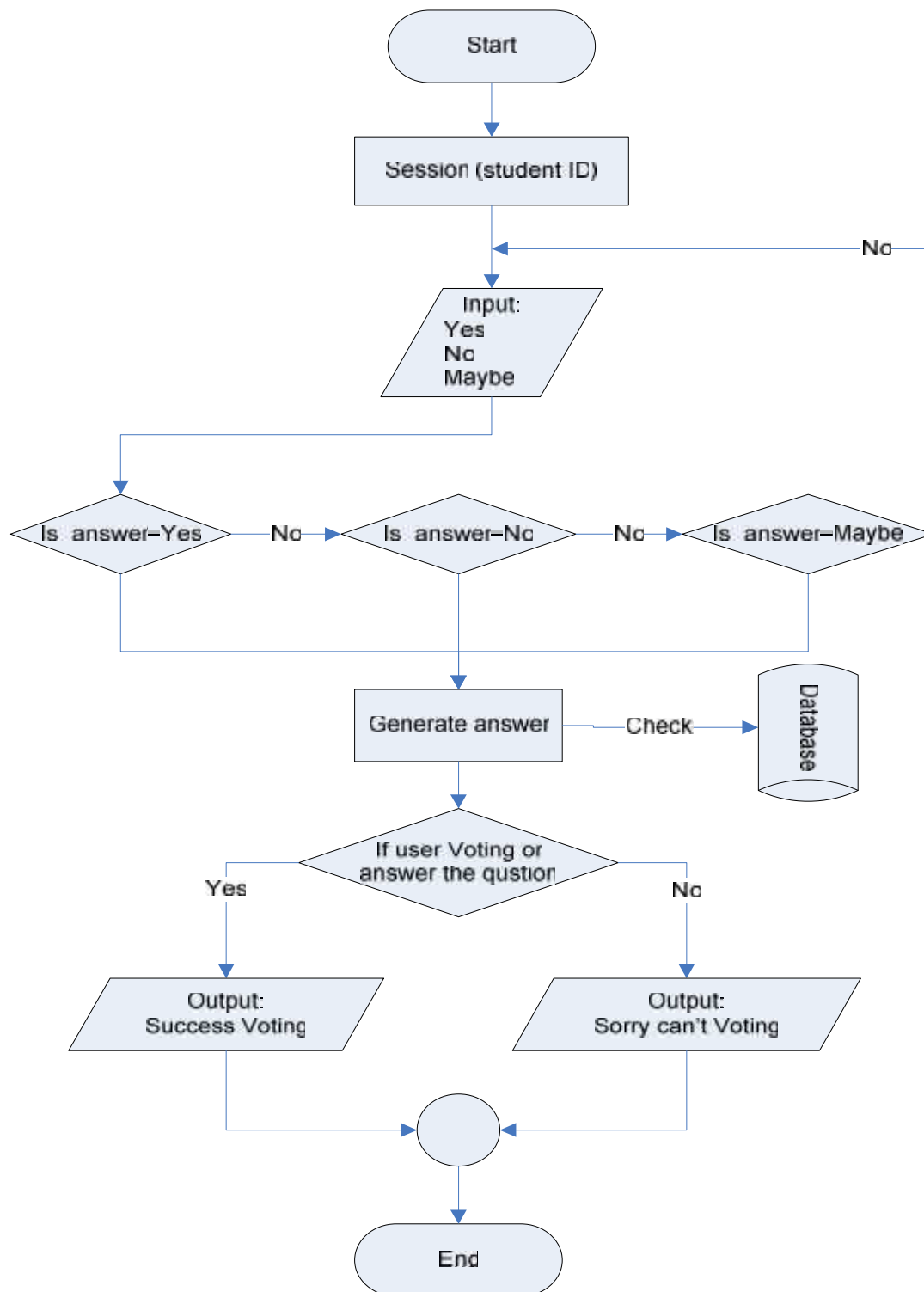
Figure(3.24) College boards interface design.

13- Student voting:

- a. description :This function enable the student to answer specific question for one time only.
- b. interface:
 - ◆ Input : selection only one answer (yes, or no, maybe)
 - ◆ Output : if he vote for first time, message with process done will appeared, else message that tell you that you were voting before will appeared.
- c. Constraints:
 - ◆ Student can't voting more than one time for the same question, and only one answer.
 - ◆ Student only read question and can't edit it.



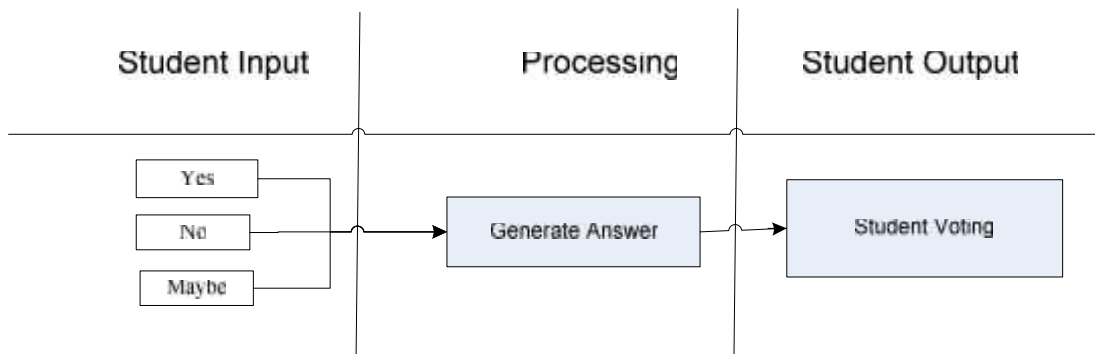
d. Flowchart:



Figure(3.25) Student voting operation.



>> User interface design



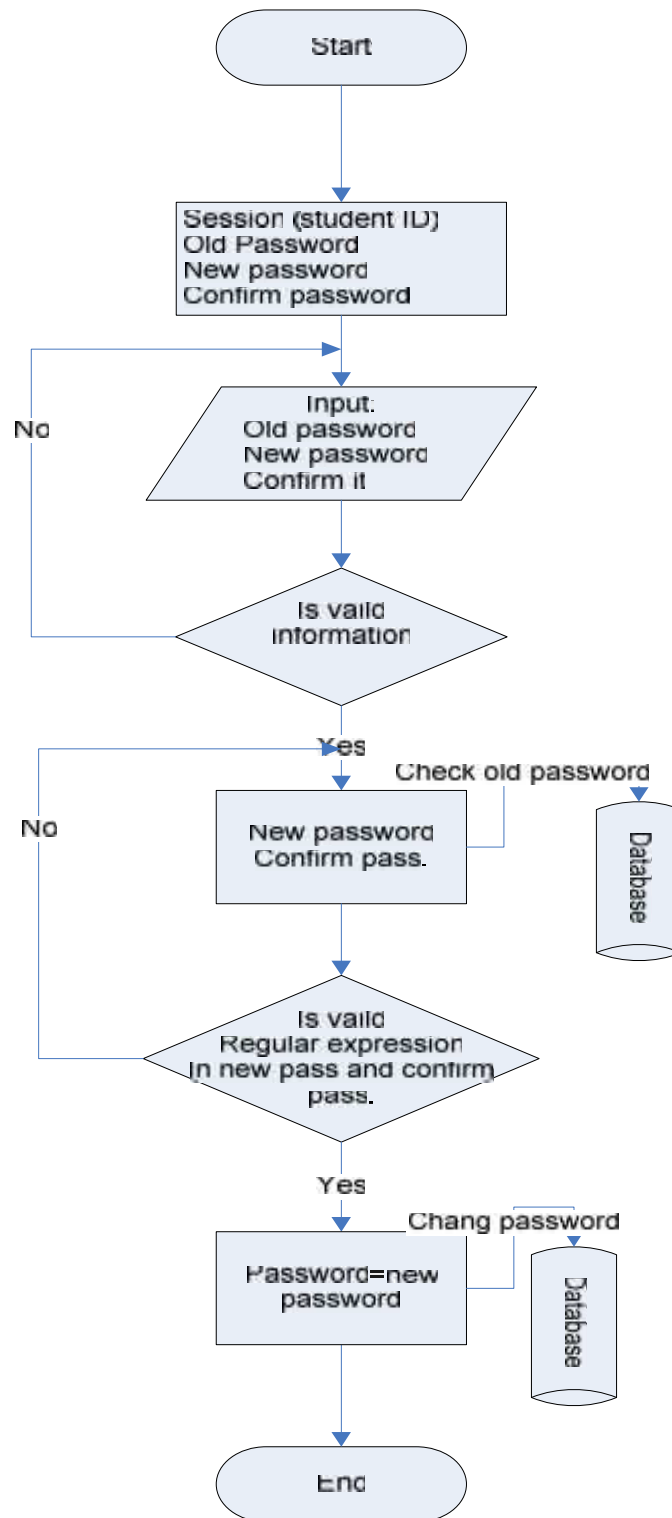
Figure(3.26) Student voting interface design.

14- Chang Password :

- a. Description :this function enable student to change his password .
- b. Interface:
 - ◆ Input : student id in session, old password, new password, confirm new password
 - ◆ Output : new password (your password was updated)
- c. Constraints:
 - ◆ New password and its confirmation must match.
 - ◆ The new password will take place at the next login .
 - ◆ New password must be at least 6 characters.



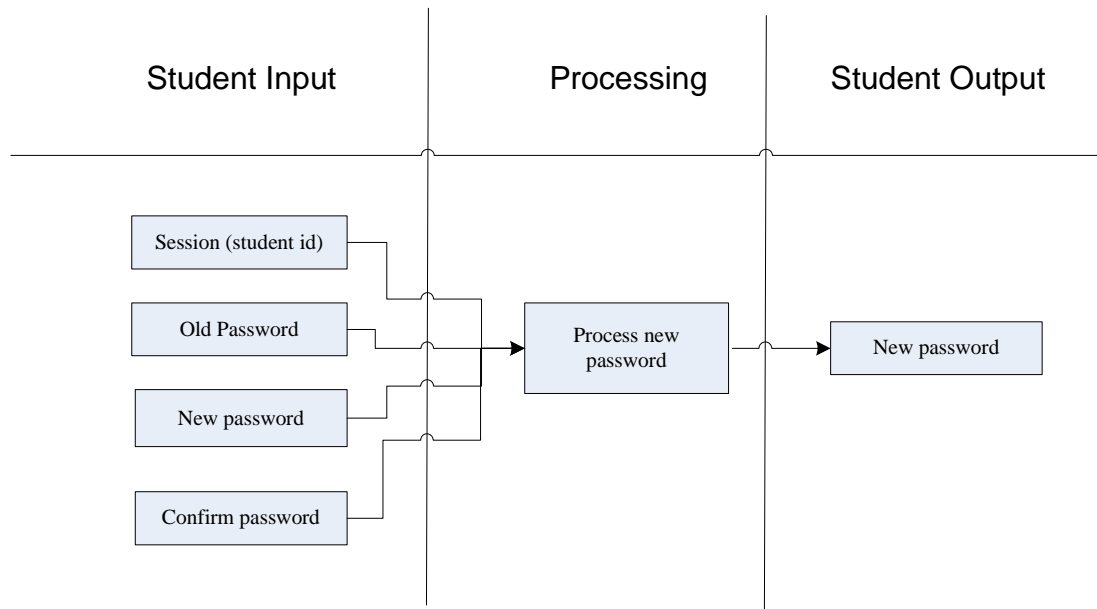
d. Flowchart:



Figure(3.27) Change password operation.



>> User interface design



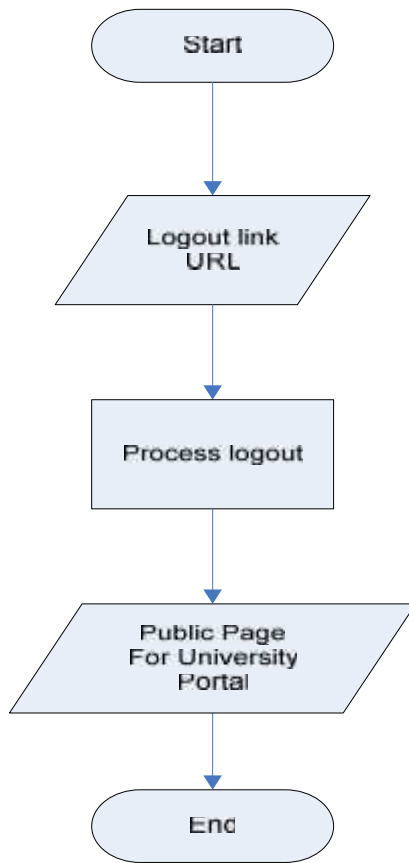
Figure(3.28) Change password interface design.

15- *Logout Student :*

- a. **Description:** This function logging student out of his session, his pass session will expired by take 0 value, and he go back to home page.
- b. **Interface:**
 - ◆ **Input:** Click on logout link.
 - ◆ **Output:** Home page.
- c. **Constraints:** None.

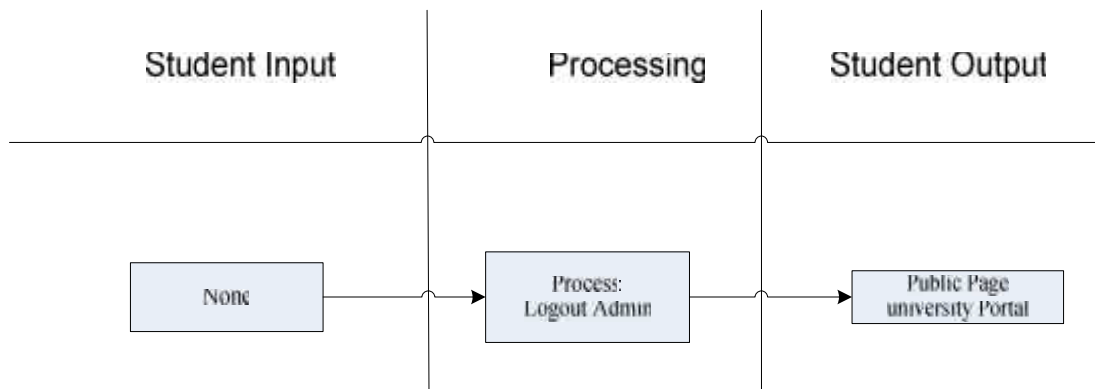


d. Flowchart:



Figure(3.29) Student logout operation.

>> User interface design;



Figure(3.30): Student logout interface design.

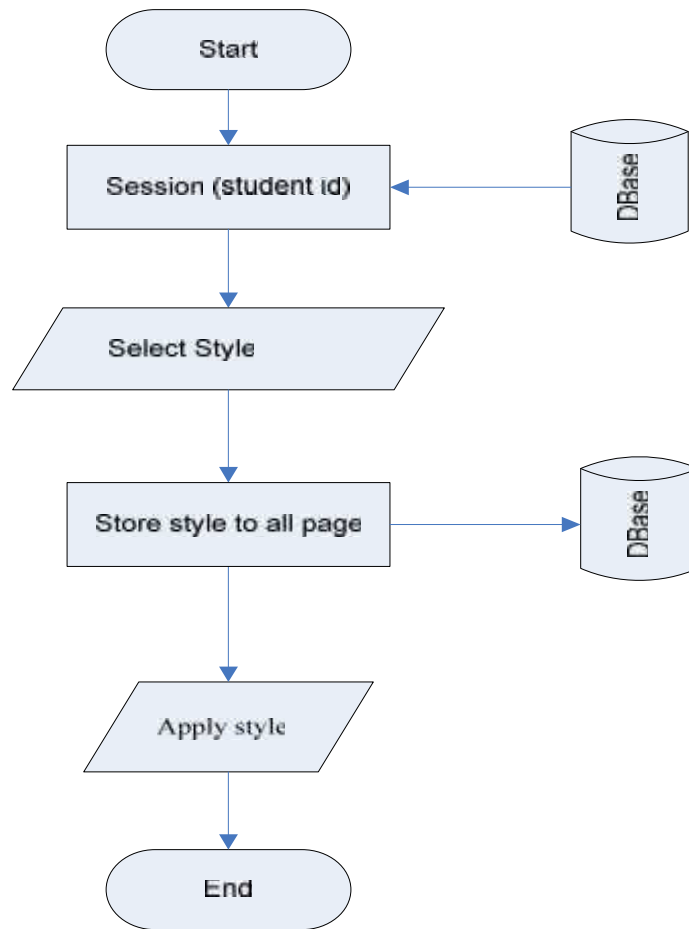


16- *My Style:*

- a. **Description :** This function enable student to customize his account interface by selecting one style from given list and apply it.
- b. **Interface:**
 - ◆ Input: Student id (username) in session, select his style.
 - ◆ Output: New interface style.
- c. **Constraints:**
 - ◆ Only authenticated student can show this page.
 - ◆ Students only can change there style.
 - ◆ New style apply only for the student who select it.
 - ◆ New style will apply after refresh the page.

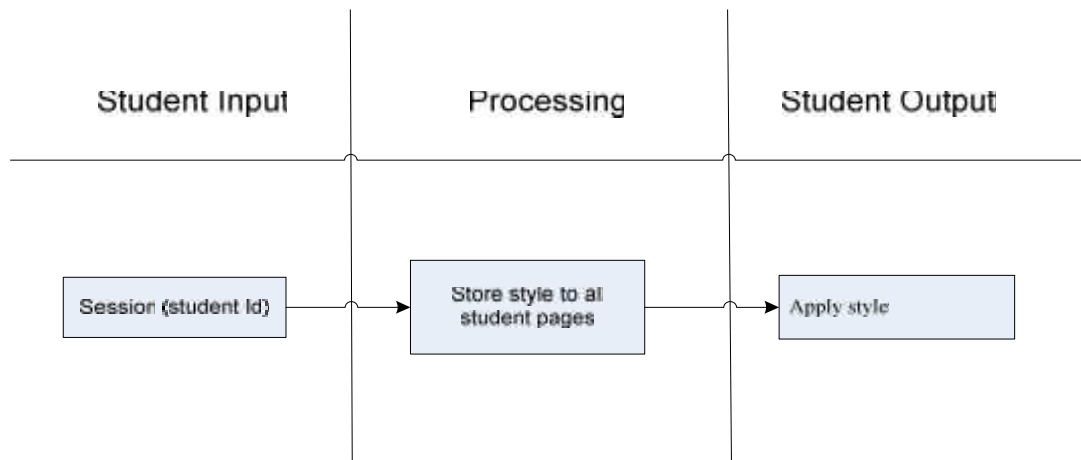


d. Flowchart :



Figure(3.31): My style operation

>> User interface design;



Figure(3.32): My style user interface



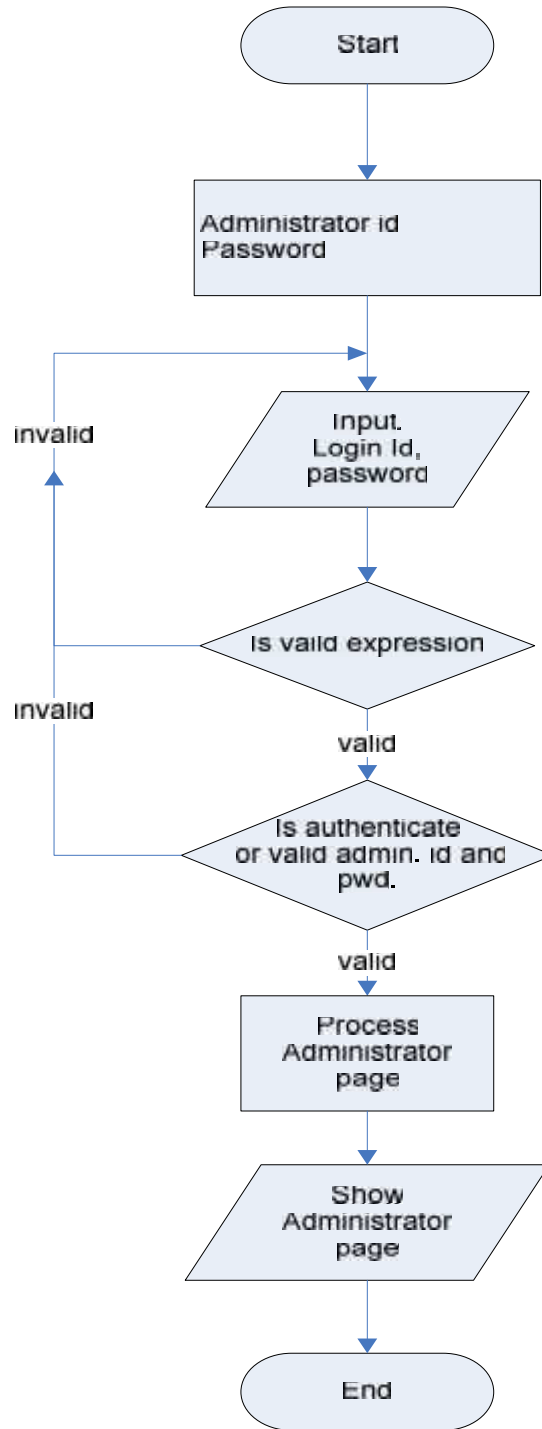
B. Administrator Functions design

1- Login Administrator :

- a. Description :This function enable Administrator to login to the portal.
- b. Interface:
 - ◆ Input : login id, password
 - ◆ Output : Administrator desktop (home page).
- c. Constraints:
 - ◆ The password should be a character with at least 6 characters.
 - ◆ The password must be encrypted.



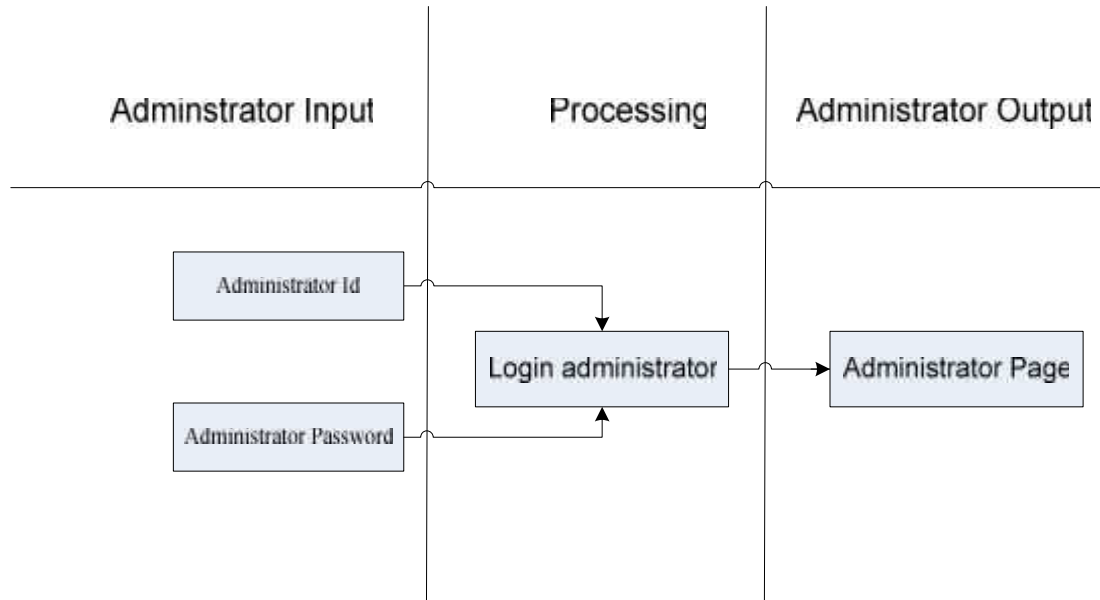
d. Flowchart:



Figure(3.33) Log in administrator operation.



>> User interface design



Figure(3.34) Log in administrator interface design.

2- Insert boards :

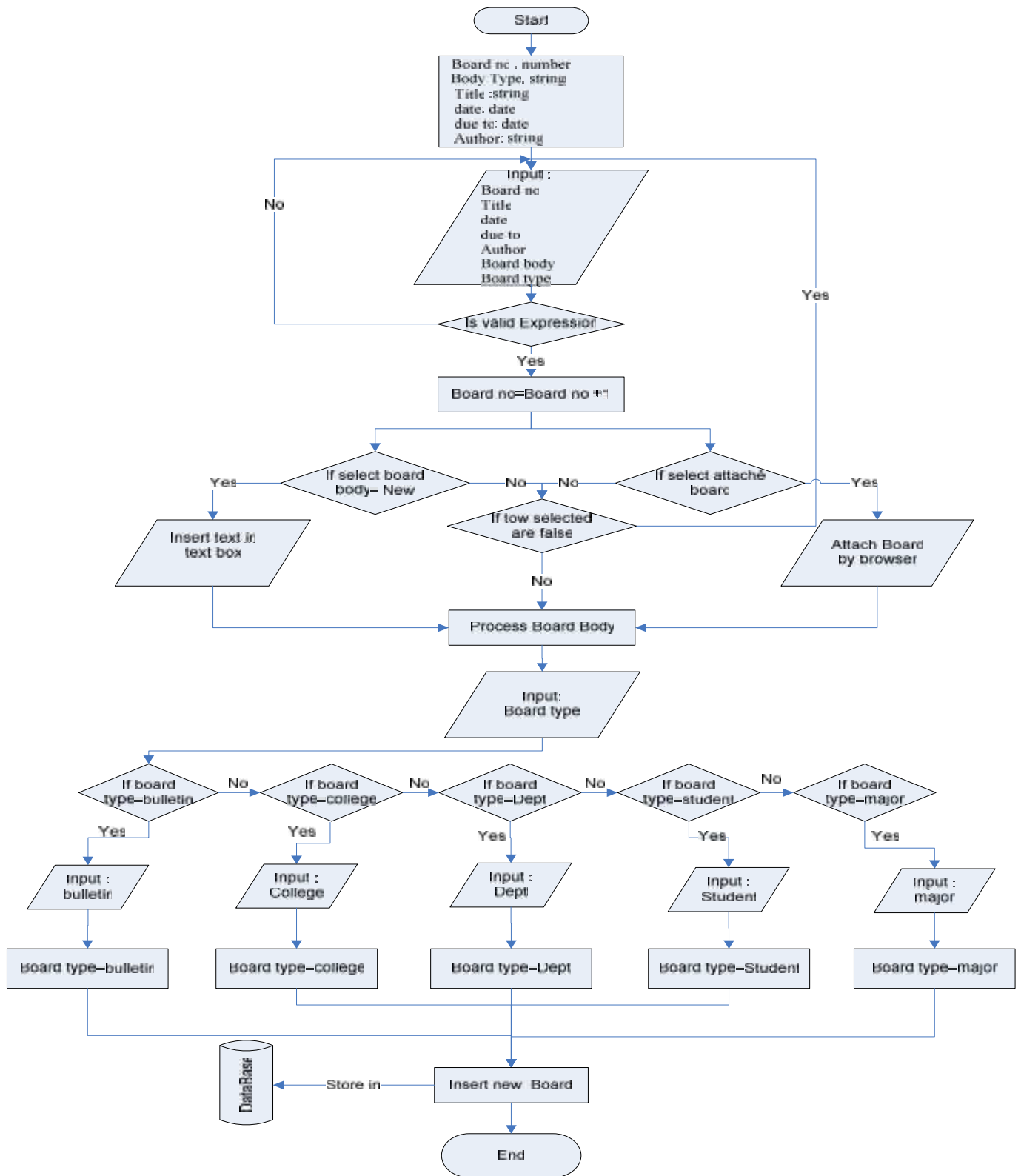
- a. description :This function enable Administrator to upload new boards for students, colleges, departments, courses, or overall university (General boards).
- b. interface:
 - ◆ Input : Board No, Body text or attachment or both, board type, Title, issue date, due to date, Author, destination.
 - ◆ Output : New Board uploaded.



c. Constraints:

- ◆ Board body can be text or attachment or both.
- ◆ Board can be uploaded for more than one destination.
- ◆ Authenticated administrator can basically upload four types of boards(student, college, department, courses), and public board
- ◆ Only authenticated administrator can show this page.

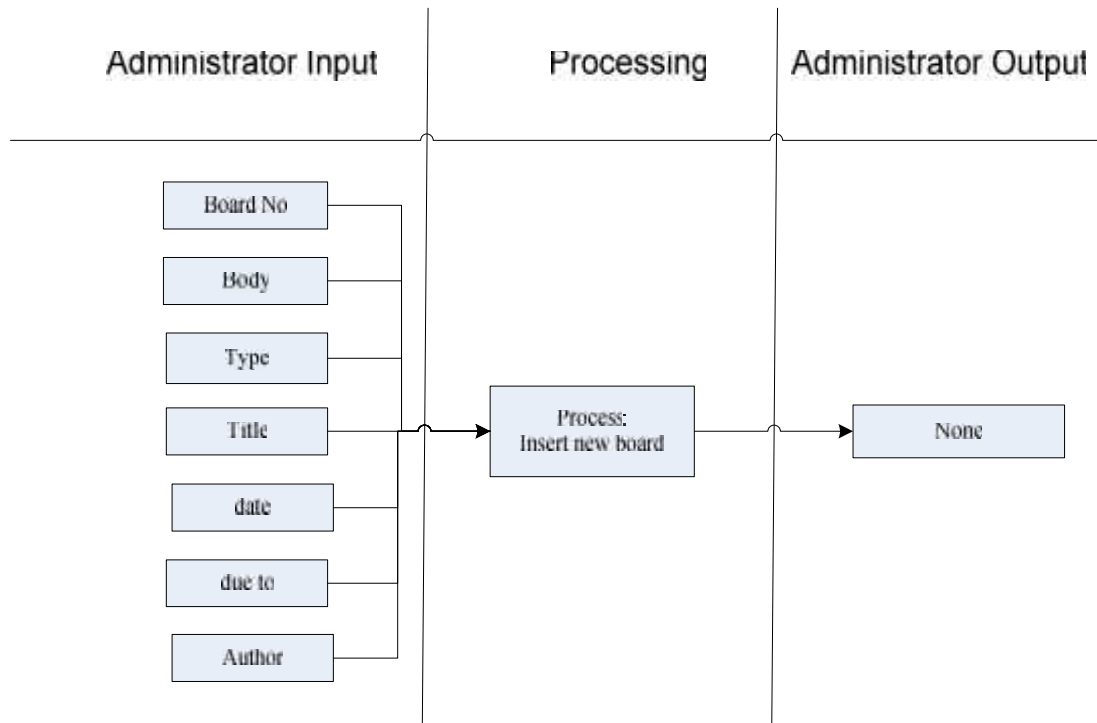
d. Flowchart:



Figure(3.35) Insert board operation.



>> User interface design



Figure(3.36) Insert boards interface design.

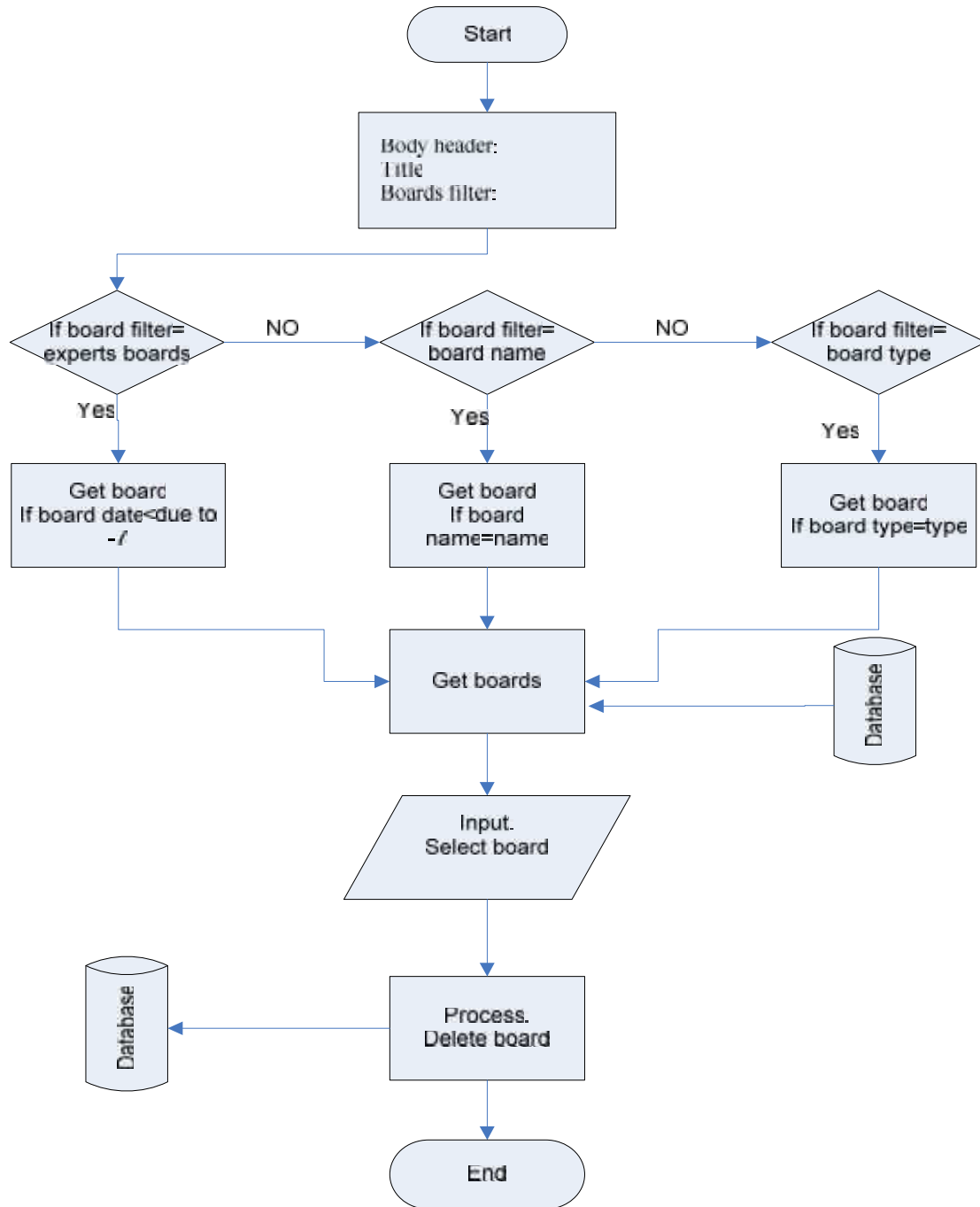
3- Delete Boards :

- a. description :This function allow the Administrator to delete the boards according several criteria (expired boards, by name, boards type) .
- b. interface:
 - ◆ Input : select board name to be deleted, put it in recycle bin, click delete button.
 - ◆ Output : deleted board
- c. Constraints:
 - ◆ Authenticated administrator only can delete boards.



- ◆ Boards can be deleted manually by administrator or automatically by the system (further work).

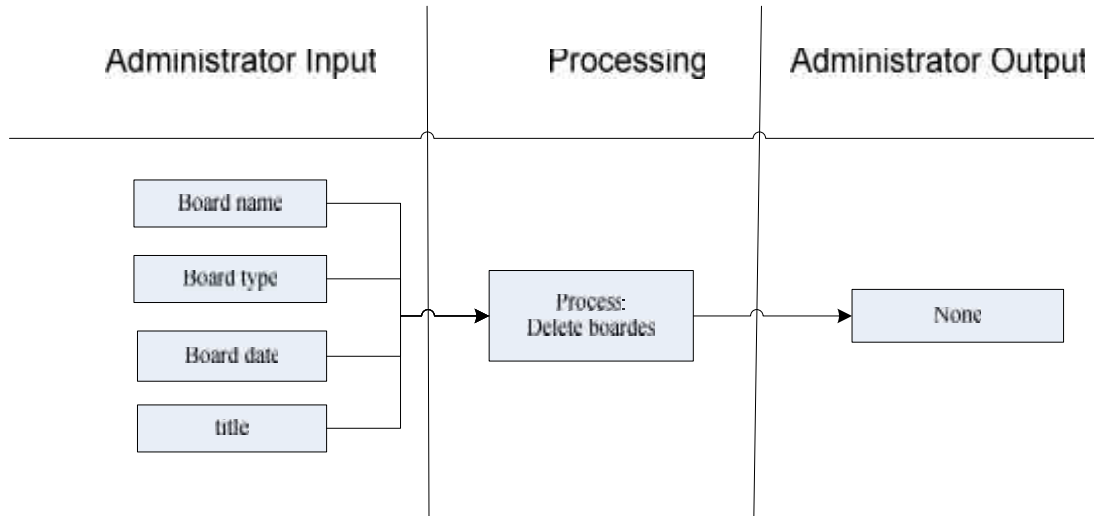
d. Flowchart:



Figure(3.37) Delete boards operation.



>> User interface design



Figure(3.38) Delete boards interface design.

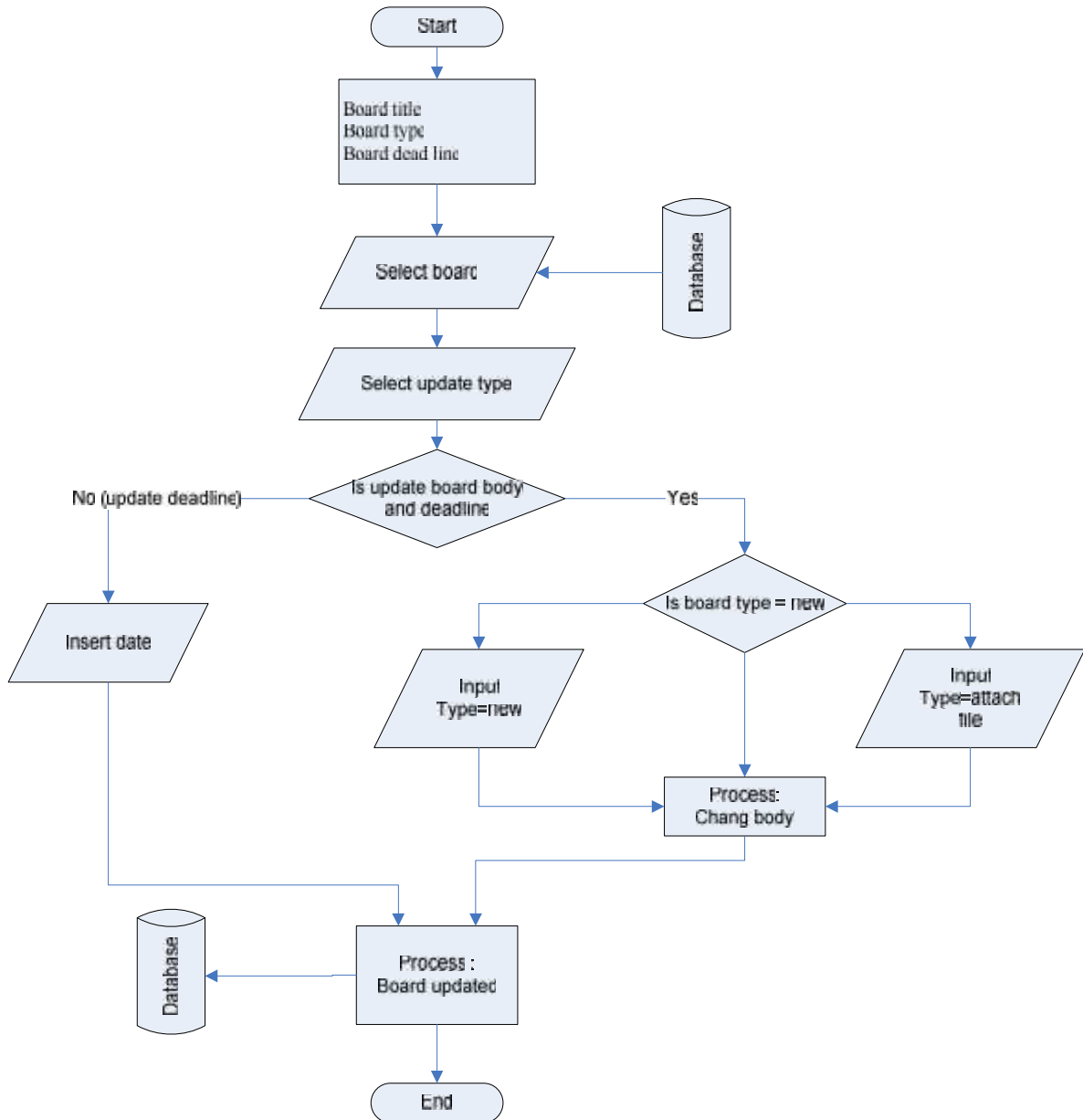
4- Update Boards :

- a. description :This function allow the Administrator to update the existing board body or board due to date or both.
- b. interface:
 - ◆ Input : board title, board type(exit, or new), new board dead line, or new board body, or both.
 - ◆ Output : updated board
- c. Constraints:
 - ◆ Boards that can be updated generated automatically in drop down list when page request.



- ◆ Only authenticated administrator can update boards.
- ◆ Updated boards will be appeared after refresh the page.
- ◆ Board due to date must be in correct format (month/day/year).

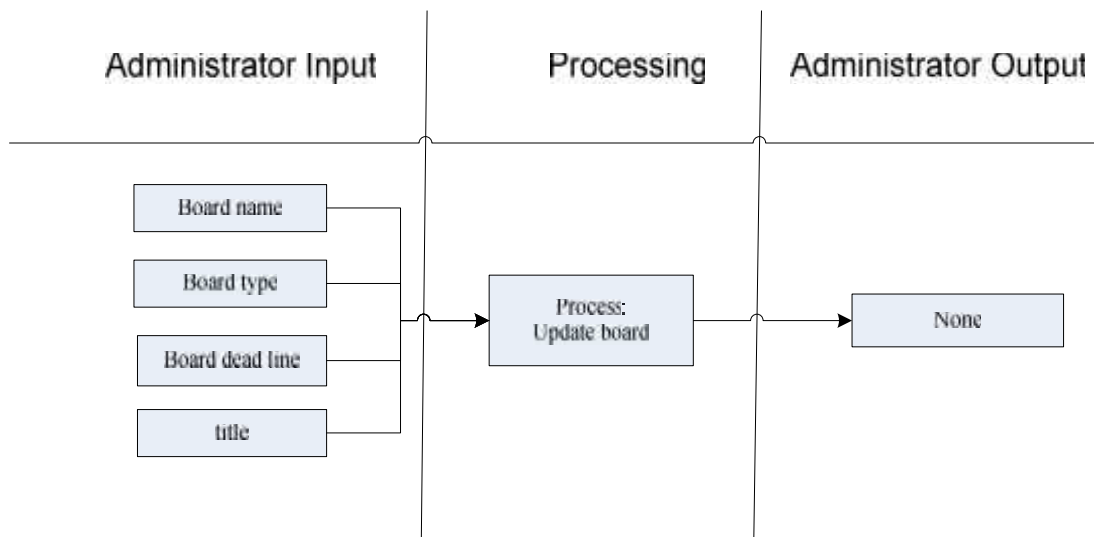
d. Flowchart:



Figure(3.39) Update boards operation.



>>User interface design



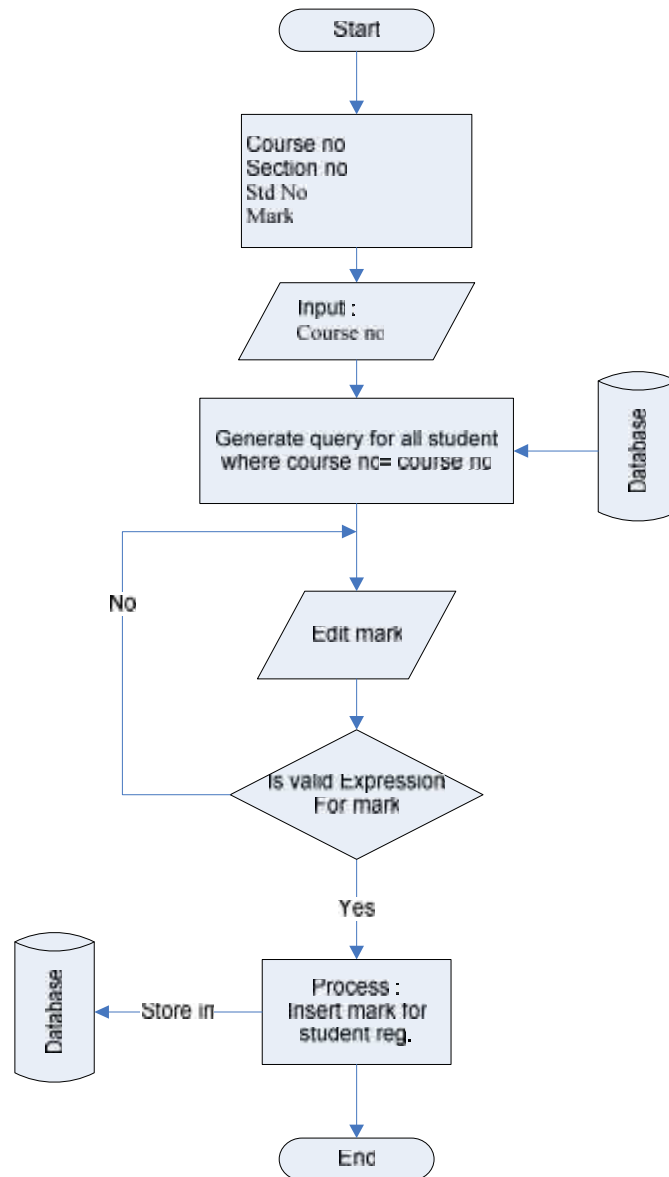
Figure(3.40) Update boards interface design.

5- Insert Marks :

- a. description :This function enable Administrator to update the students marks, depending on course number.
- b. interface:
 - ◆ Input : course no, student mark .
 - ◆ Output : Update students marks
- c. Constraints:
 - ◆ All enrolled courses will appear in drop down list automatically when request the page.
 - ◆ Only students that enrolled the selected course will appear in table with there information.
 - ◆ The inserted mark can't be less then 40 or more then 100, and can't be in negative value.



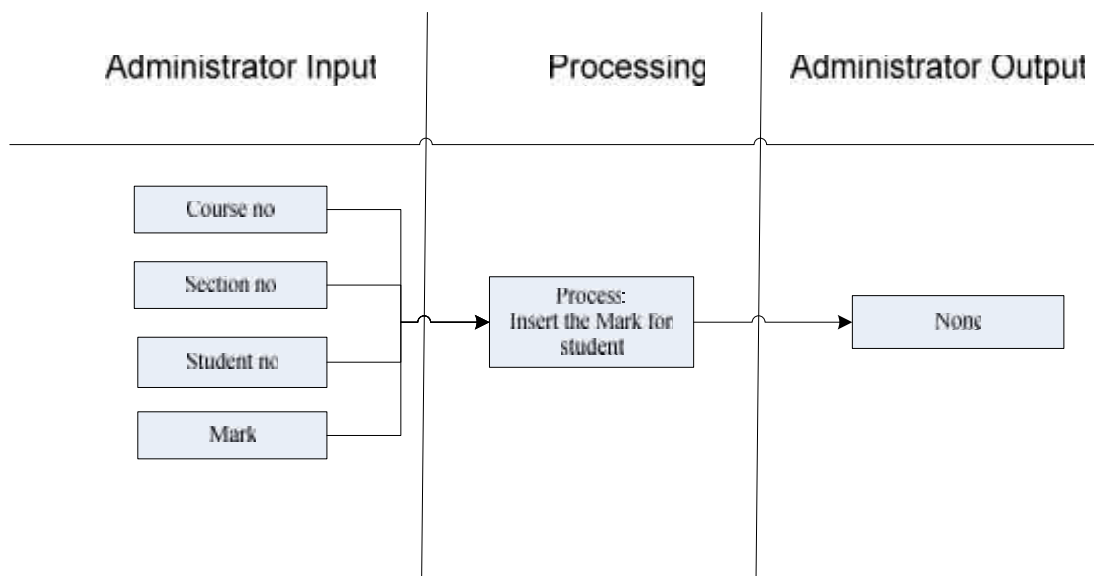
d. Flowchart:



Figure(3.41) Insert marks operation.



>>User interface design



Figure(3.42) Insert marks interface design.

6- Insert College :

a. description :

This function allow the Administrator to Insert the new college information to the portal.

b. interface:

- ◆ Input : College No, College Name, College Description
- .
- ◆ Output : new college

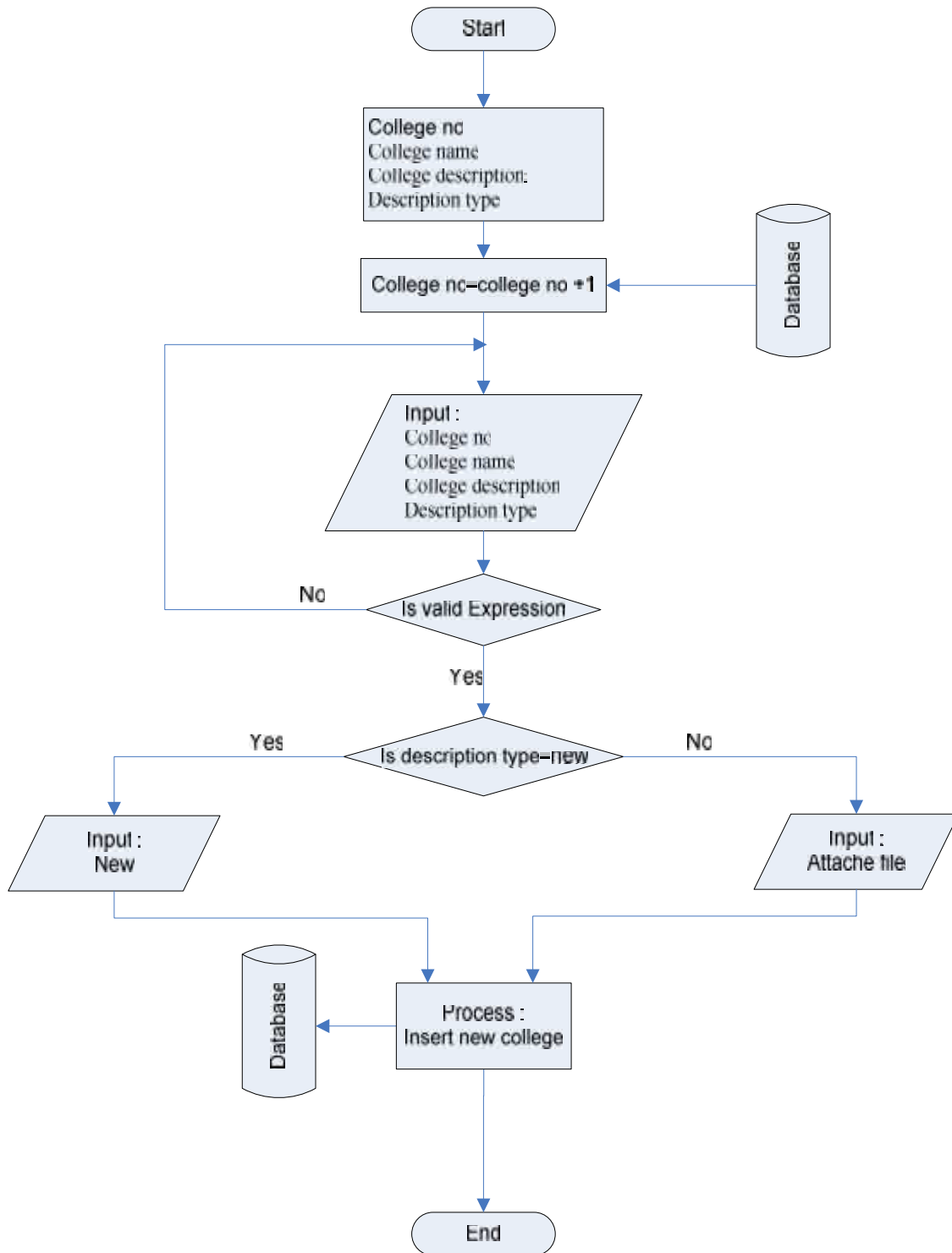


c. Constraints:

- ◆ New college number generated automatically .
- ◆ New college description can be either text or attachment file or both.
- ◆ New college name can't be number, or start with number.



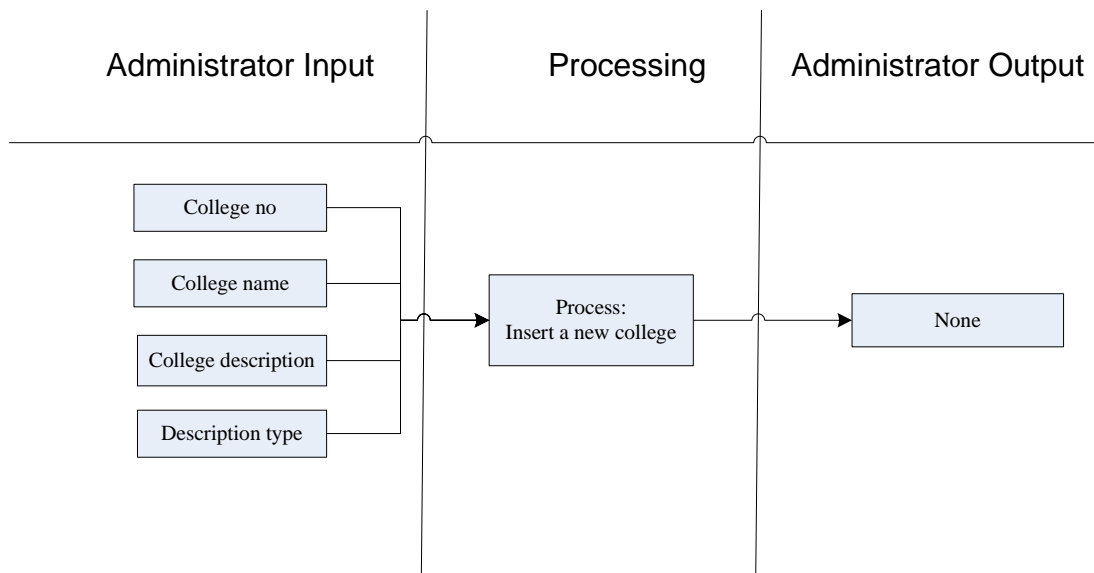
d. Flowchart:



Figure(3.43) Insert new college operation.



>>User interface design



Figure(3.44) insert new college interface design.

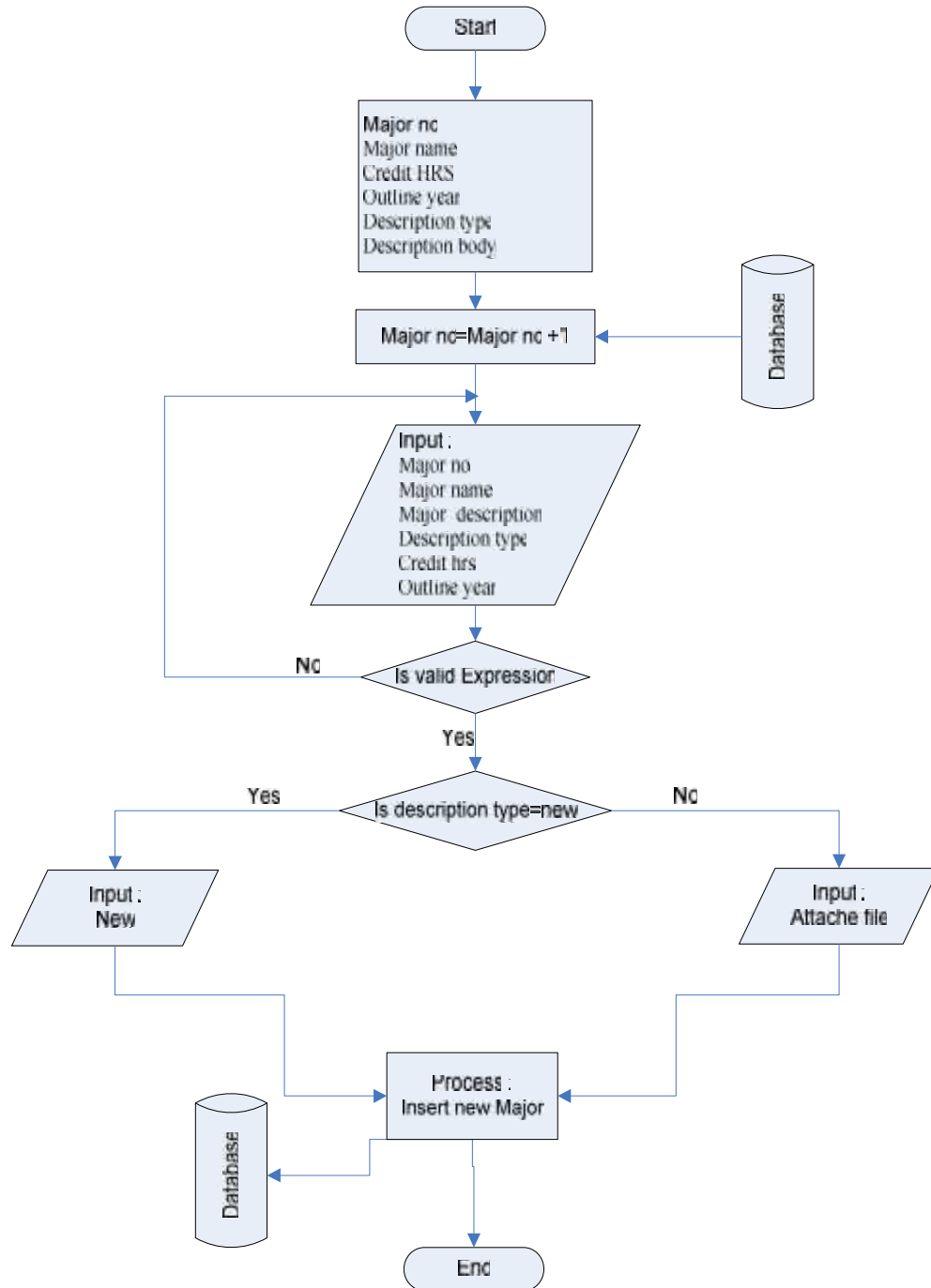
7- Insert Major :

- a. description :This function allow the Administrator to Insert a new major information to the portal.
- b. interface:
 - ◆ Input : Major No, Major Name, Major Description, Description type, Credit HRS, Outline year.
 - ◆ Output : new Major
- c. Constraints:
 - ◆ New major number generated automatically .
 - ◆ New major description can be either text or attachment file or both.
 - ◆ Outline must be number.



- ◆ Credit hours can't be less than 1 or more than 4.

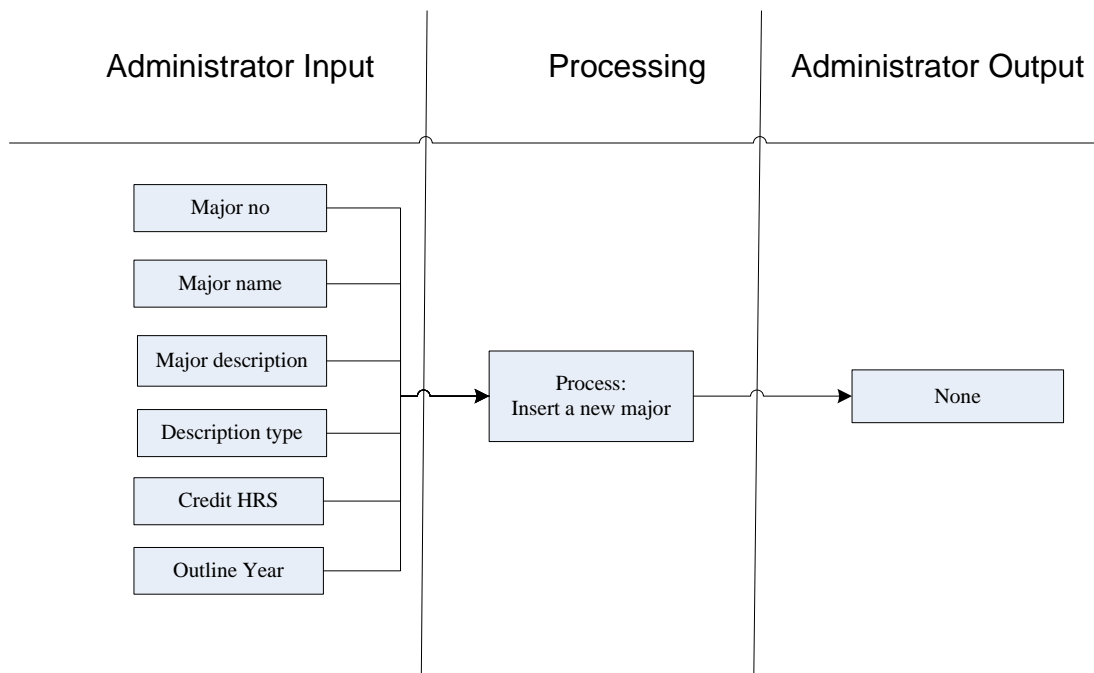
d. Flowchart:



Figure(3.45) insert major operation.



>>User interface design



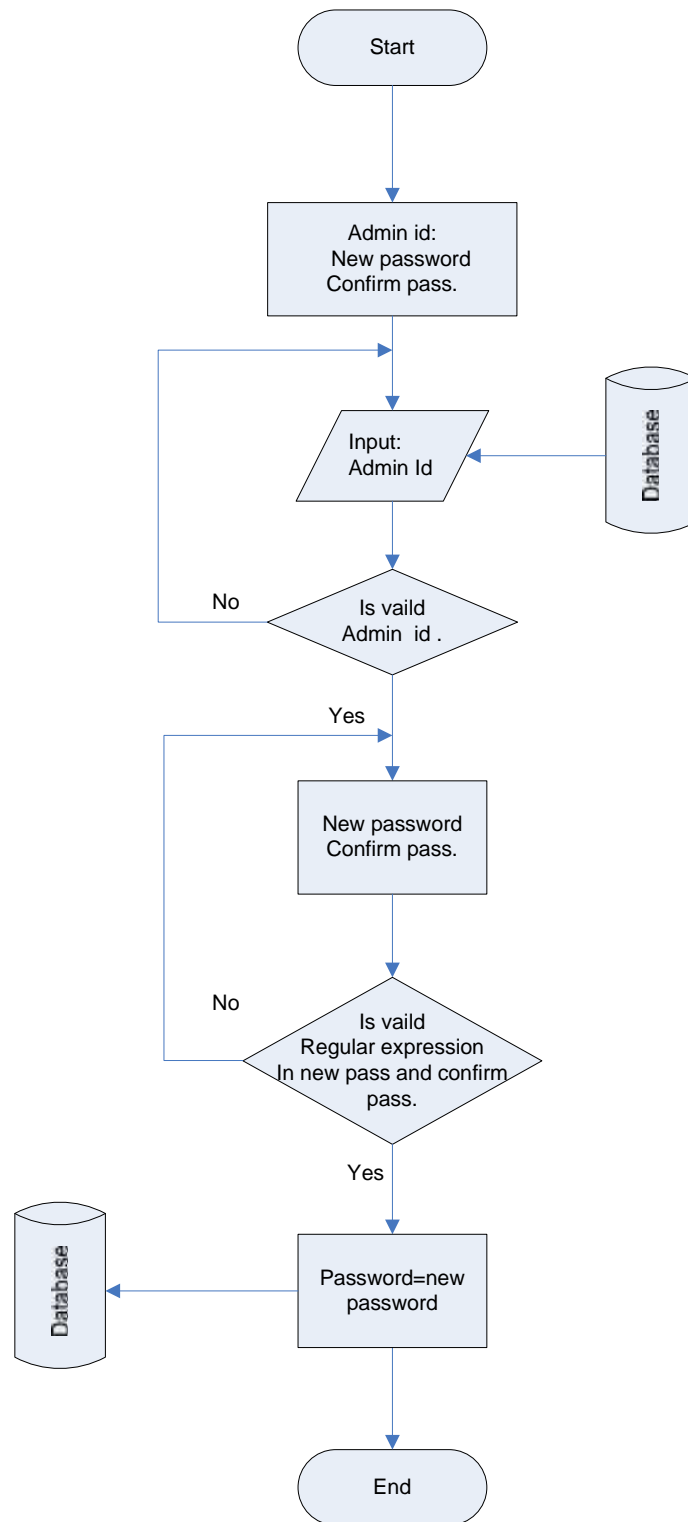
Figure(3.46) Insert major interface design.

8- Change Password for administrator :

- a. description :This function enable Administrator to change his password .
- b. interface:
 - ◆ Input : Admin id, old password, new password, confirm password
 - ◆ Output : new password
- c. Constraints:
 - ◆ New password and its confirmation must match.
 - ◆ The new password will take place at the next login .
 - ◆ New password must be in character and at least with 6 character.



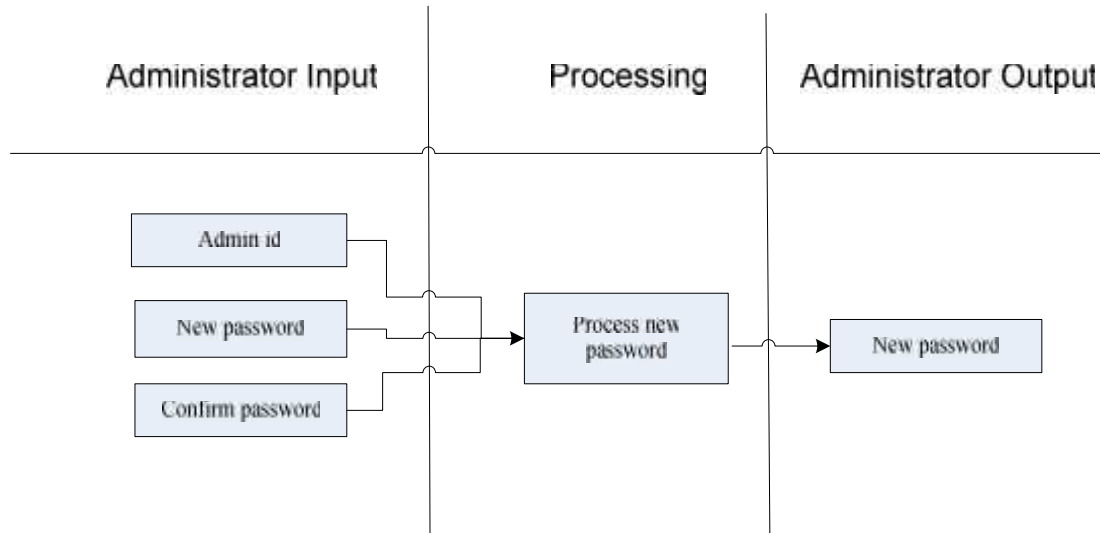
d. Flowchart:



Figure(3.47) Chang administrator's password.



>>User interface design



Figure(3.48) Change administrators password interface design.

9- Logout administrator

a. Description:

This function make administrator logging out from his session work, and terminate his pass session, and go back to home page .

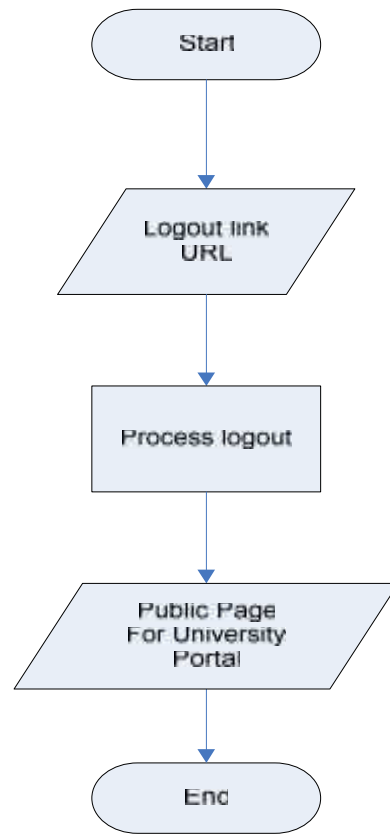
b. Interface:

- ◆ Input: Click on logout link.
- ◆ Output: home page.

c. Constraints: None.

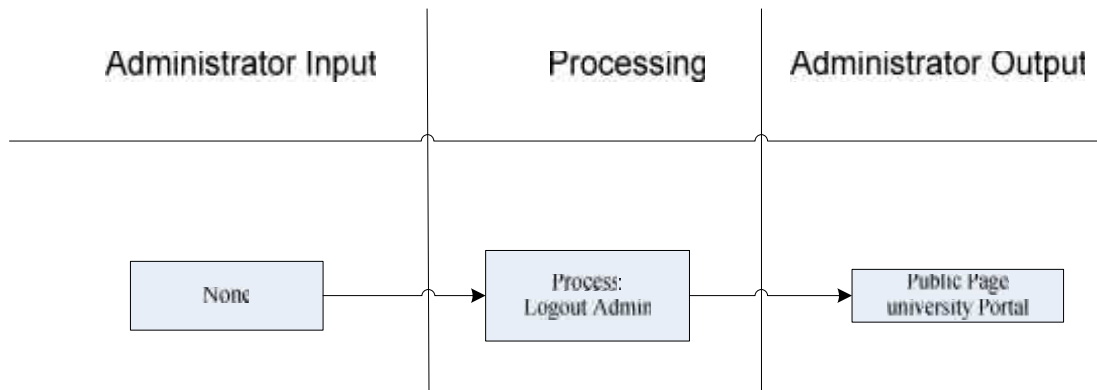


d. Flowchart:



Figure(3.49) Administrator logout operation.

>> User interface design:



Figure(3.50) Administrator logout interface design.



C. Instructor Functions design

1- Insert board :

a. description :This function enable instructor to upload new boards Only for courses who enrolled and student who teaching them..

b. interface:

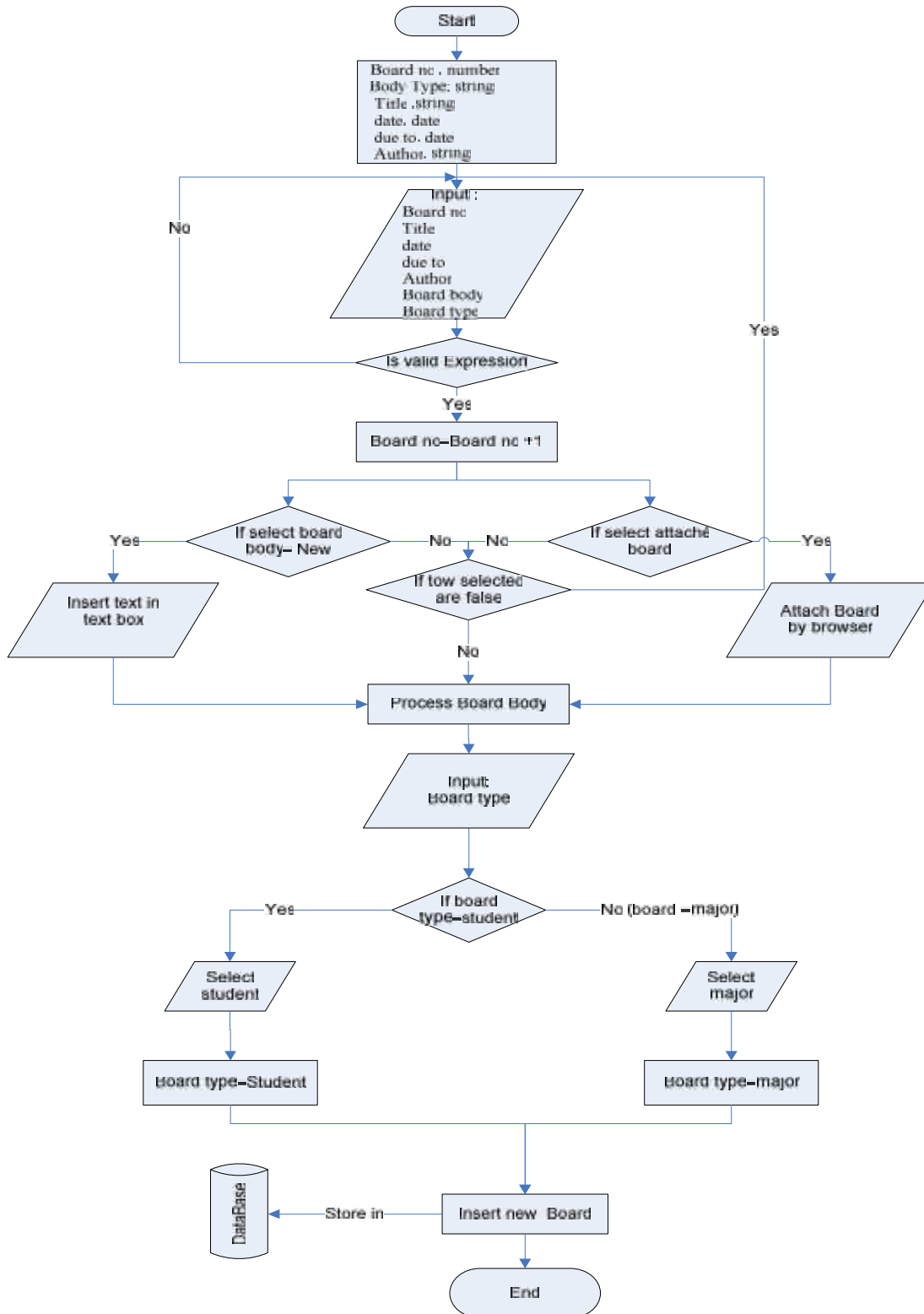
- ◆ Input : Board No, Body, Type, Title, issue date, due to date, Author, destination.
- ◆ Output : New Board.

c. Constraints:

- ◆ Instructor can only upload courses and students boards.
- ◆ Instructor can only upload boards for his students and his enrolled courses.
- ◆ Instructor have no authentication to delete or update boards.



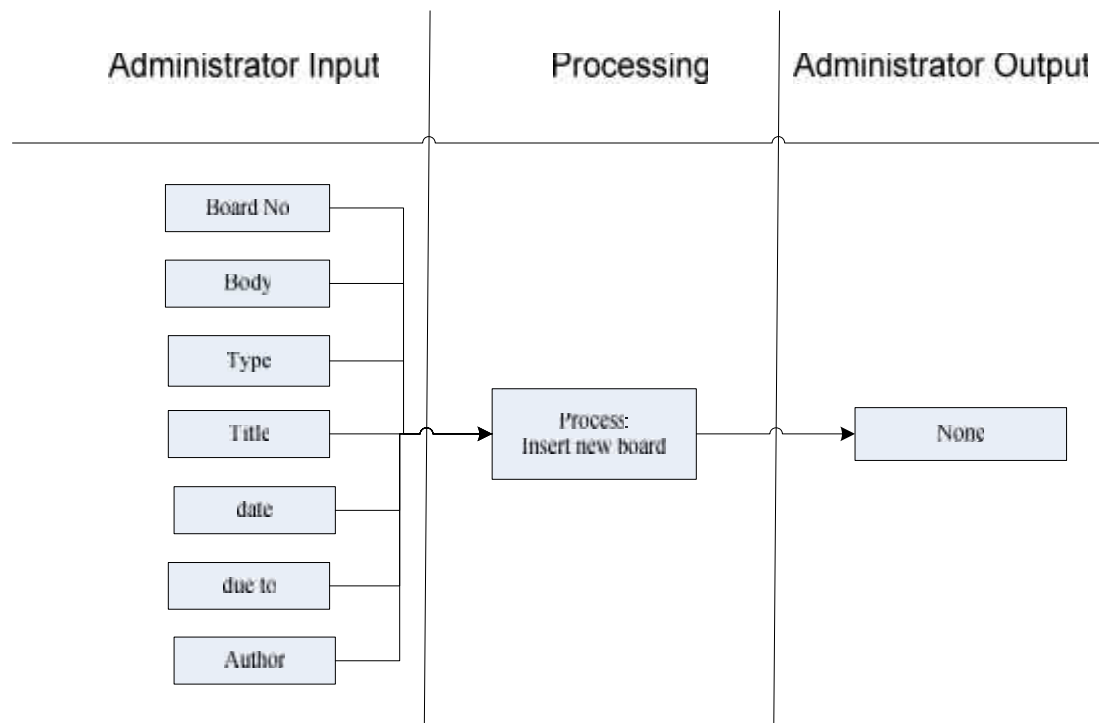
d. Flowchart:



Figure(3.51) Instructor insert board operation.



>>User interface design



Figure(3.52) Instructor insert board interface design.

2- Insert Marks :

a. description :This function enable instructor to update marks for his students in his enrolled courses according section .

b. interface:

- ◆ Input : course no, section no, student mark
- ◆ Output : updated students marks

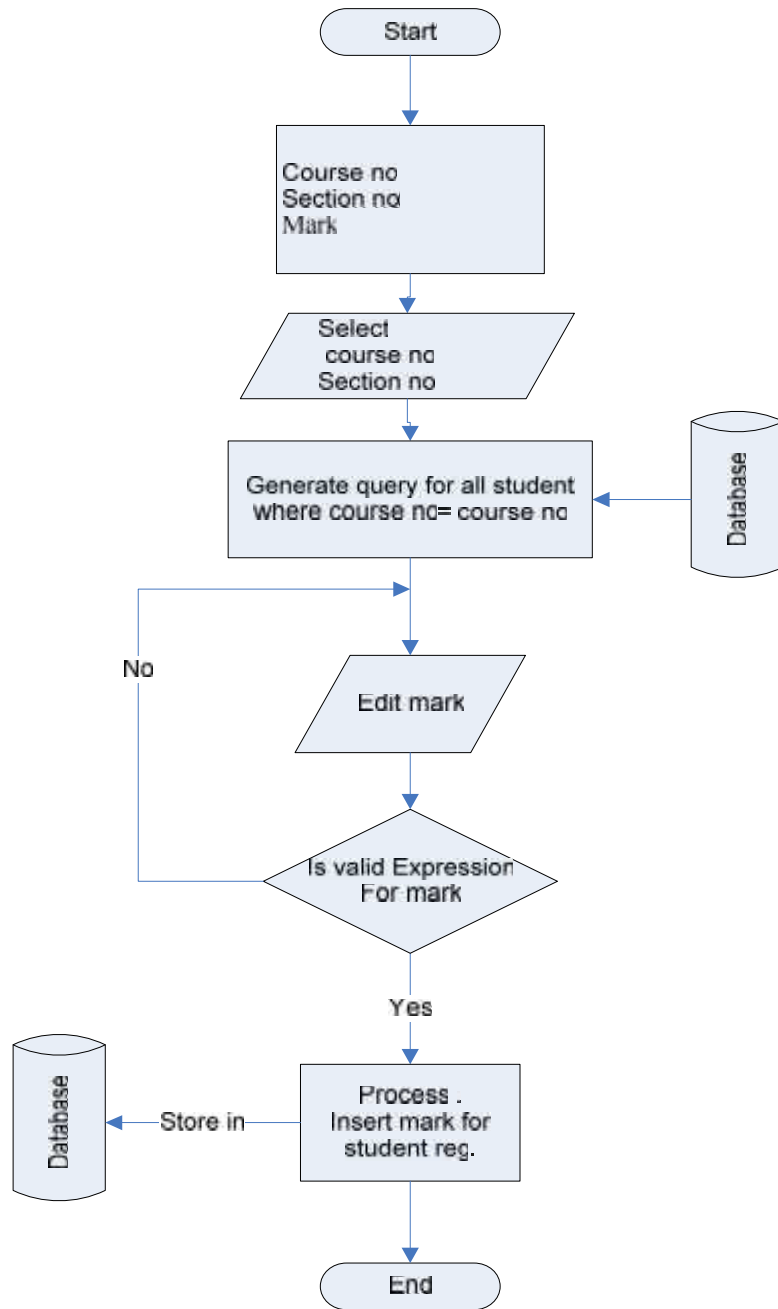
c. Constraints:

- ◆ Student mark can't be less than 40 or more than 100, or in negative value..



- ◆ Instructor can update marks only for students that enrolled his enrolled courses and specially for his sections

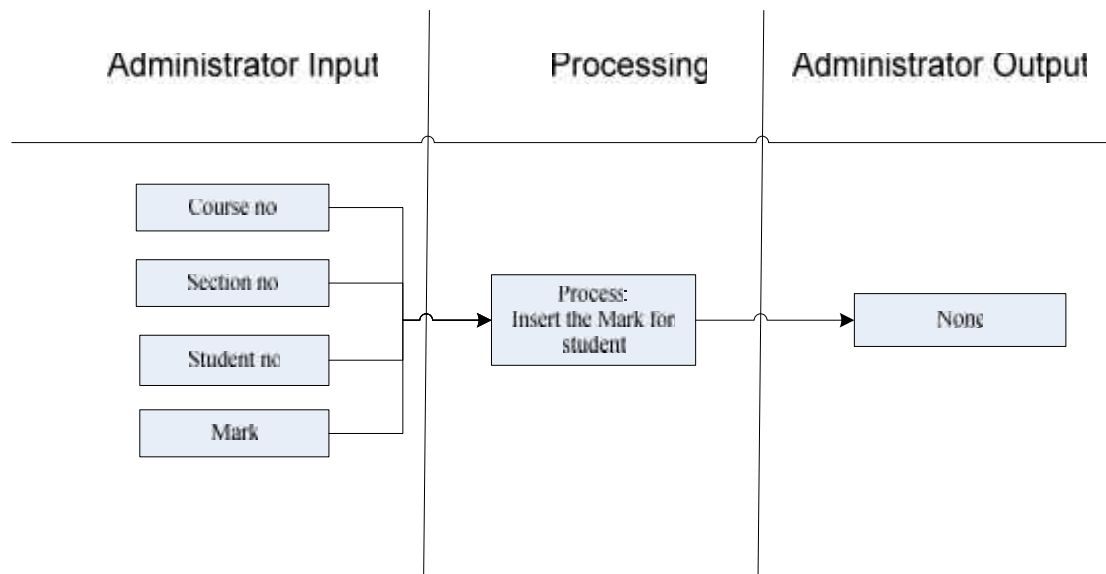
d. Flowchart:



Figure(3.53) Insert marks operation.



>>User interface design



Figure(3.54) insert marks interface design.

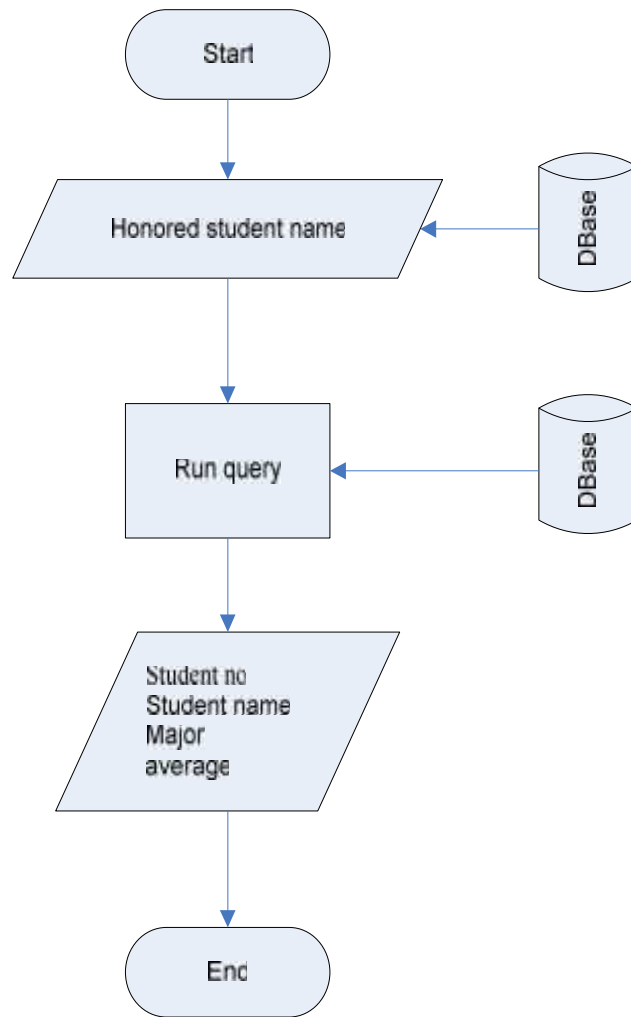
D- Public Functions design

1- Honored students :

- a. description : Display all honored students names in a vertical marquee.
- b. interface:
 - ◆ Input : student name.
 - ◆ Output : std no, std name, major, average.
- c. Constraints:
 - ◆ None.



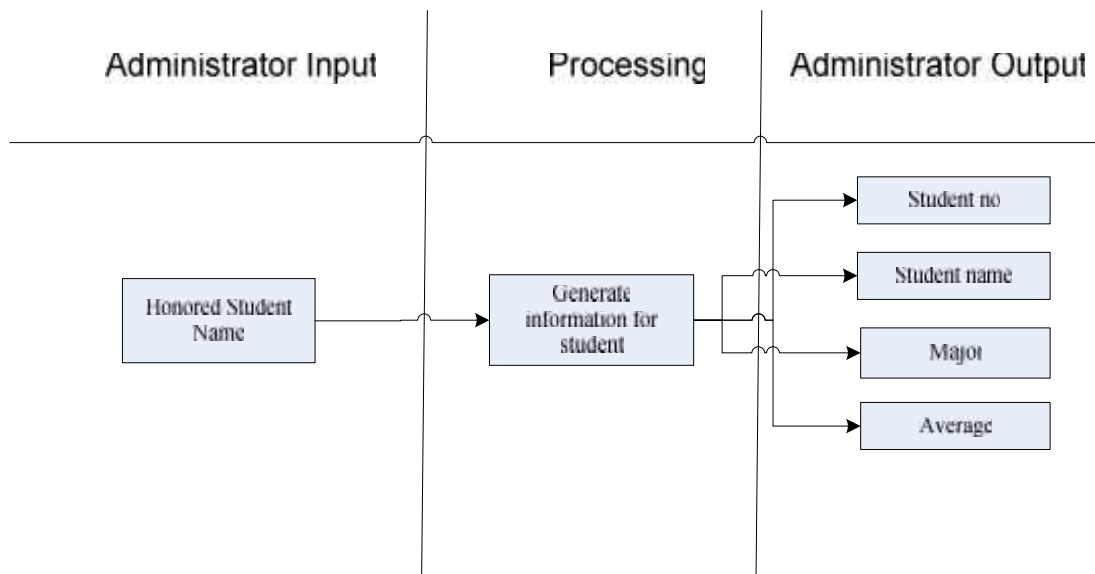
d. Flowchart



Figure(3.55) Honored students operation.



>>User interface design



Figure(3.56) Honored students interface design.

2- Public search :

a. description : search that devoted to all visitors to the web site, this search restricted on portal database for colleges, courses, departments, boards and employees.

a. interface:

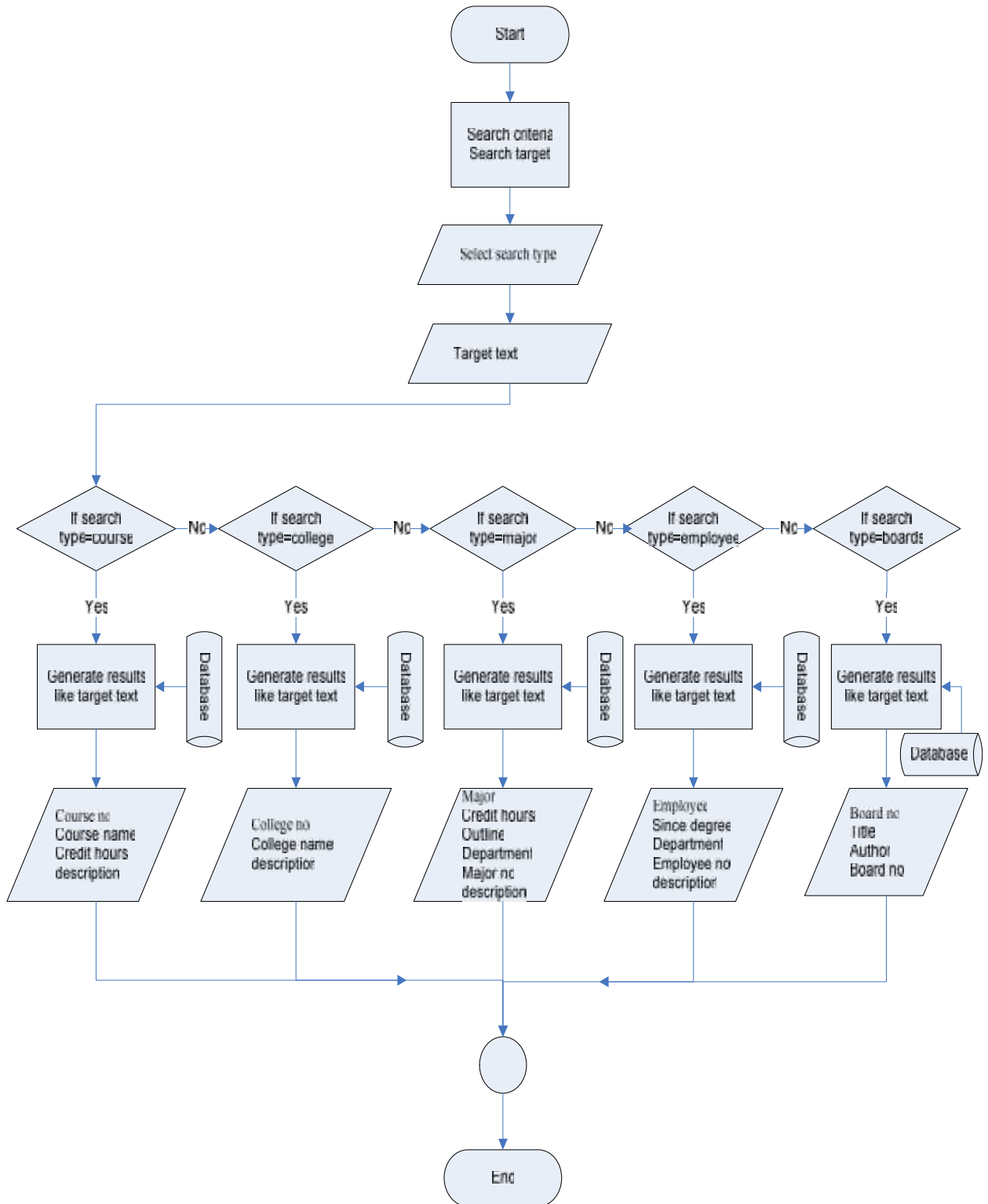
- ◆ Input: target search prefix.
- ◆ Output: desired search item.

b. Constraints:

- ◆ None.



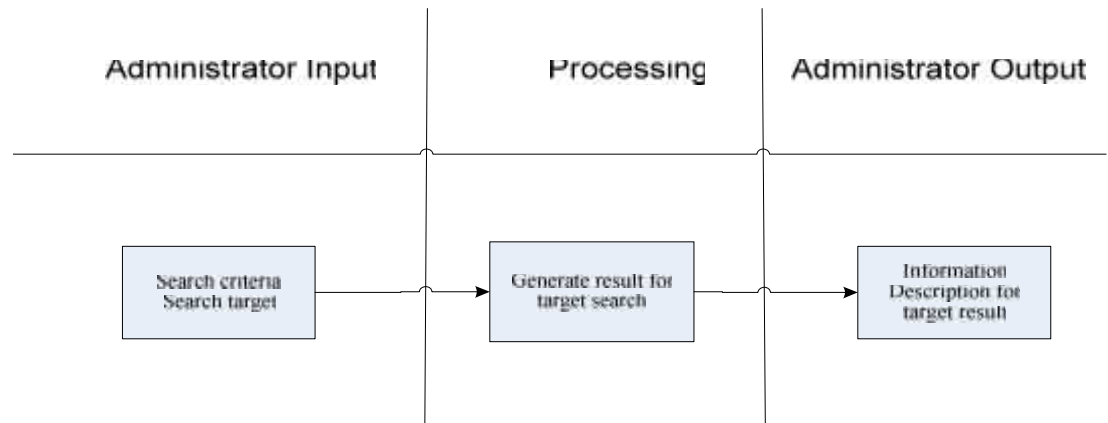
c. Flowchart



Figure(3.57) Public search operation.



>>User interface design



Figure(3.58) Public search interface design.

3- Marquee Boards :

a. description :

Displays all boards issued since seven days in a scrolled marquee on top of the home page, this marquee contains only colleges and department boards.

b. interface:

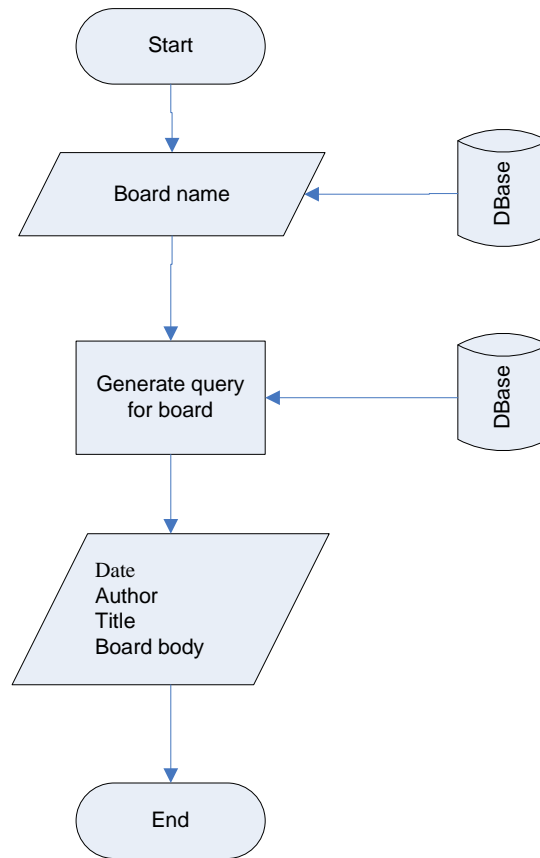
- ◆ Input: select board name from marquee.
- ◆ Output: board body.

c. Constraints:

- ◆ None.

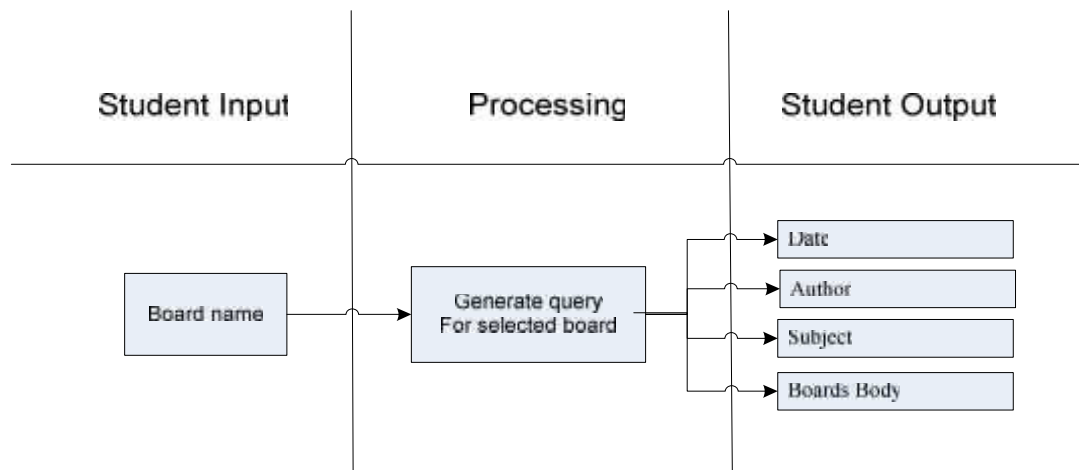


d. Flowchart



Figure(3.59) Marquee boards operation.

>>User interface design



F

Figure(3.60) Marquee boards interface design.



4- Graduate project :

a. description :

this function enable the PAP visitor to show all graduate projects information according major name. and he can download it if it belongs to attachments type.

b. interface:

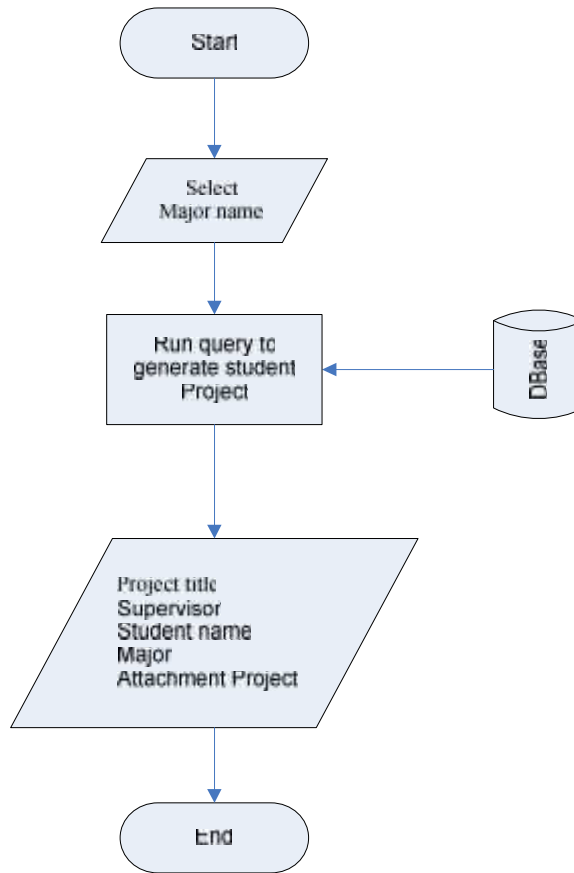
- ◆ Input: Major name.
- ◆ Output: project name, project supervisor, major, project attachment.

c. Constraints:

- ◆ None.

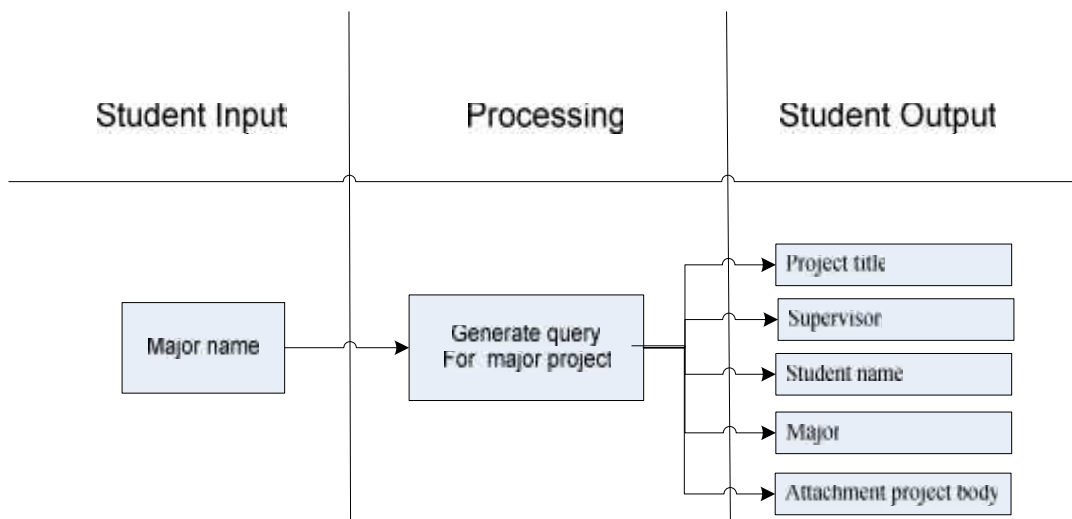


d. Flowchart



Figure(3.61) Graduate projects operation.

>>User interface design



Figure(3.62) Graduate projects interface design



3.3 *Input Output design :*

After describing requirements specification and system functions, we must show input/output screens. The designed forms shows how function works and shows how user can interact with it . and these steps of designing I/O considered as a first step for implementing and developing the software system .

In this chapter we design I/O user interface design, including student I/O design and Administrator I/O design.

A. Student interface design :

a. Student Input design:

- Student login:

Login

Please Insert Your ID And Password

Student Name :

Password :

Login

[Forgot My Password?](#)

Figure (3.63) student login screen.



➤ Student Menu :

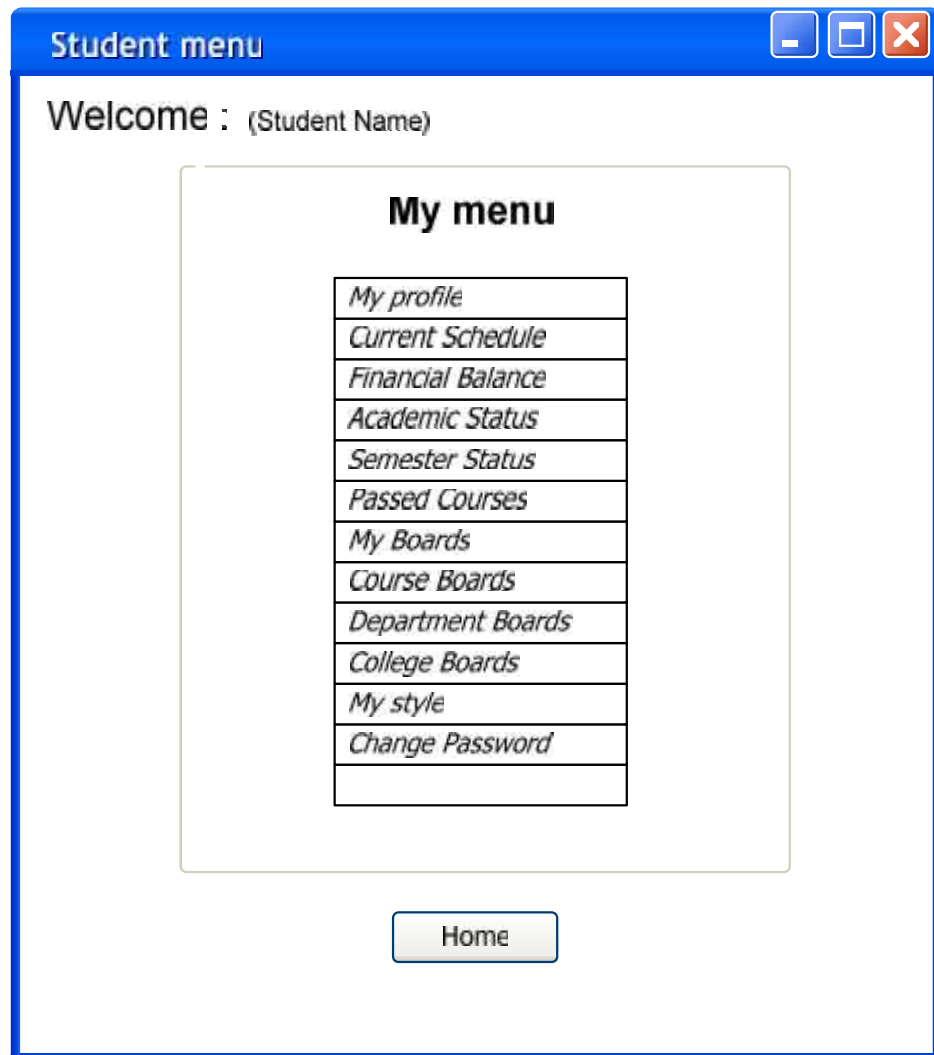


Figure (3.64) student menu.



➤ Chang My Password:

Change Password

Old Password

New Password

Confirm it

Figure (3.65) change password for student..



b. Student Output design :

➤ My profile :

My Profile [stdno]

My Name

First Name: **Family Name :**

My Address

Adress: **Telephone no:**

My Academic Info.

Academic no: **Science Degree:** **Major :** **Outline:**

[Back](#) [Home](#)

Figure (3.66) Student profile.



➤ Current Schedule :

Current Schedule [Std No]

Year : Semester :

Course No	Course Name	Section	Room No	Time	Instructor	Details
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>
						<i>select</i>

Figure (3.67) student current schedule..



➤ Financial Balance :

Financial balance

[Logout](#) [My Menu](#) [Help](#)

Financial balance [Std.No]

Year : Semester :

Financial Aids

assistance Scholarship Other

Balance

Credit Debit

Semester Balance

New

Total Balance

Deficit / surplus

Figure (3.68) student financial balance..



➤ Academic Status :

The screenshot shows a web application window titled "Academic status". At the top left, there is a "Logout" link. At the top right, there are "My Menu" and "Help" links. The main content area is titled "Academic Status" and contains the following fields:

- Year : [input field]
- Semester : [input field]
- First Name : [input field]
- Birth Place : [input field]
- Student No : [input field]
- Degree : [input field]
- Major : [input field]
- Department Name : [input field]
- College Name : [input field]
- Tawjithi Average : [input field]
- Tawjithi Branch : [input field]
- Community Work Hours : [input field]
- Passed : [input field]
- Remains : [input field]
- Accumulate : [input field]
- Major : [input field]
- Academic Alert : [input field]
- Dismiss : [input field]
- Delay : [input field]
- Level : [input field]
- Regular : [input field]
- Graduated : [input field]

Figure (3.69) student academic status.



➤ Semester Status :

[Logout](#) [My Menu](#) [Help](#)

Semester Status [stdno]

Year: Semester :

Course No	Course Name	Mark

Credit Hours :
Registered . Passed .

Honored : Semester Avg:

Figure (3.70) student semester status.



➤ Passed Courses :

Passed Courses [Std No]

[Logout](#) [My Menu](#) [Help](#)

Finished Hours

Semester : **Year :**

Filter Criteria

- Year
- Semester
- Course Type

Pick Value

Year: 2004 Semester: 1

Course Type: ****

Statistics

Acc. Avg. Pass CH.No. CH. Remains

Course No	Course Nam	Cours Typ	Credit H.	Mark	semester	Year

Figure (3.71) student passed courses.



➤ My Boards:

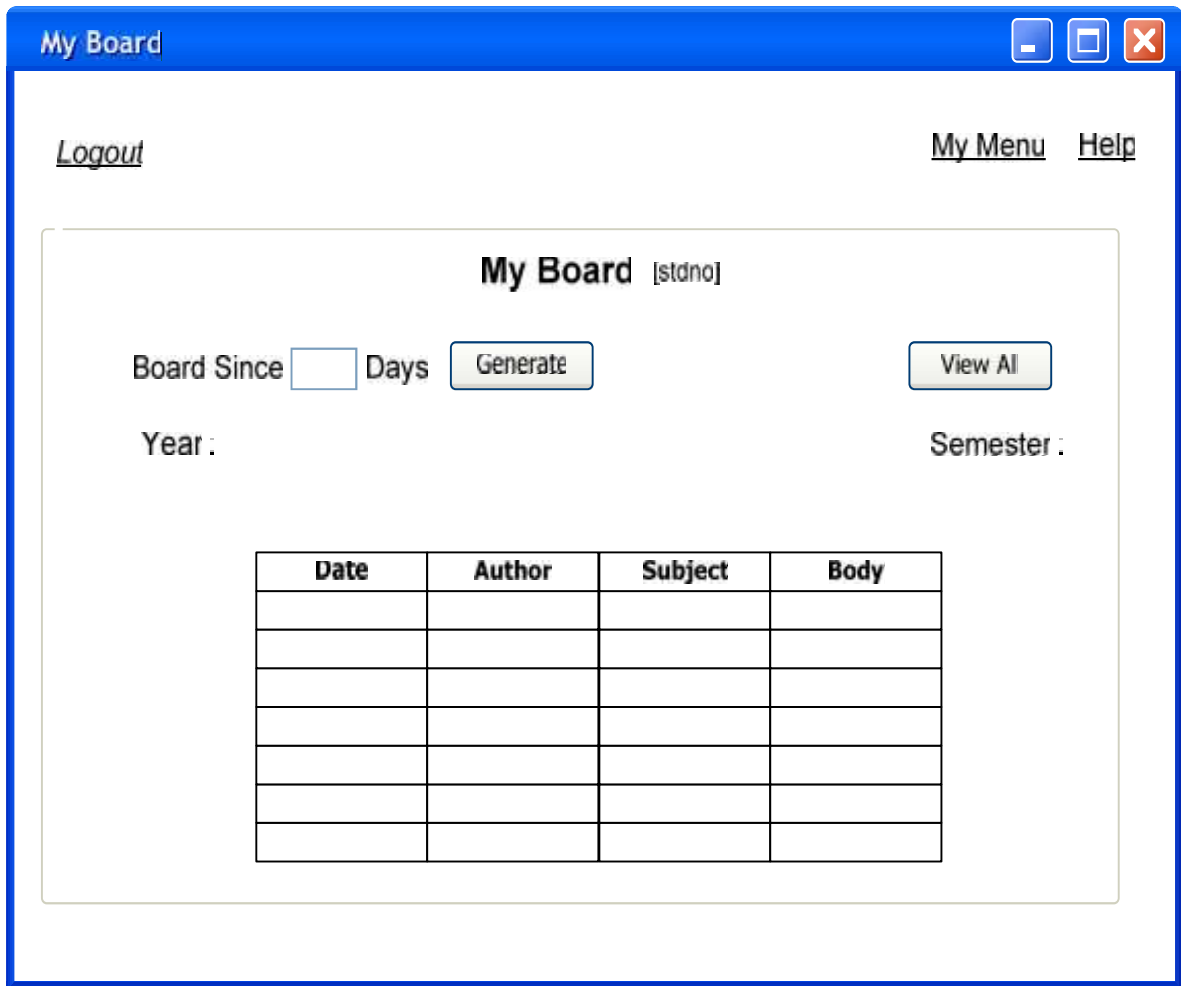


Figure (3.72) student boards.



➤ Course Boards

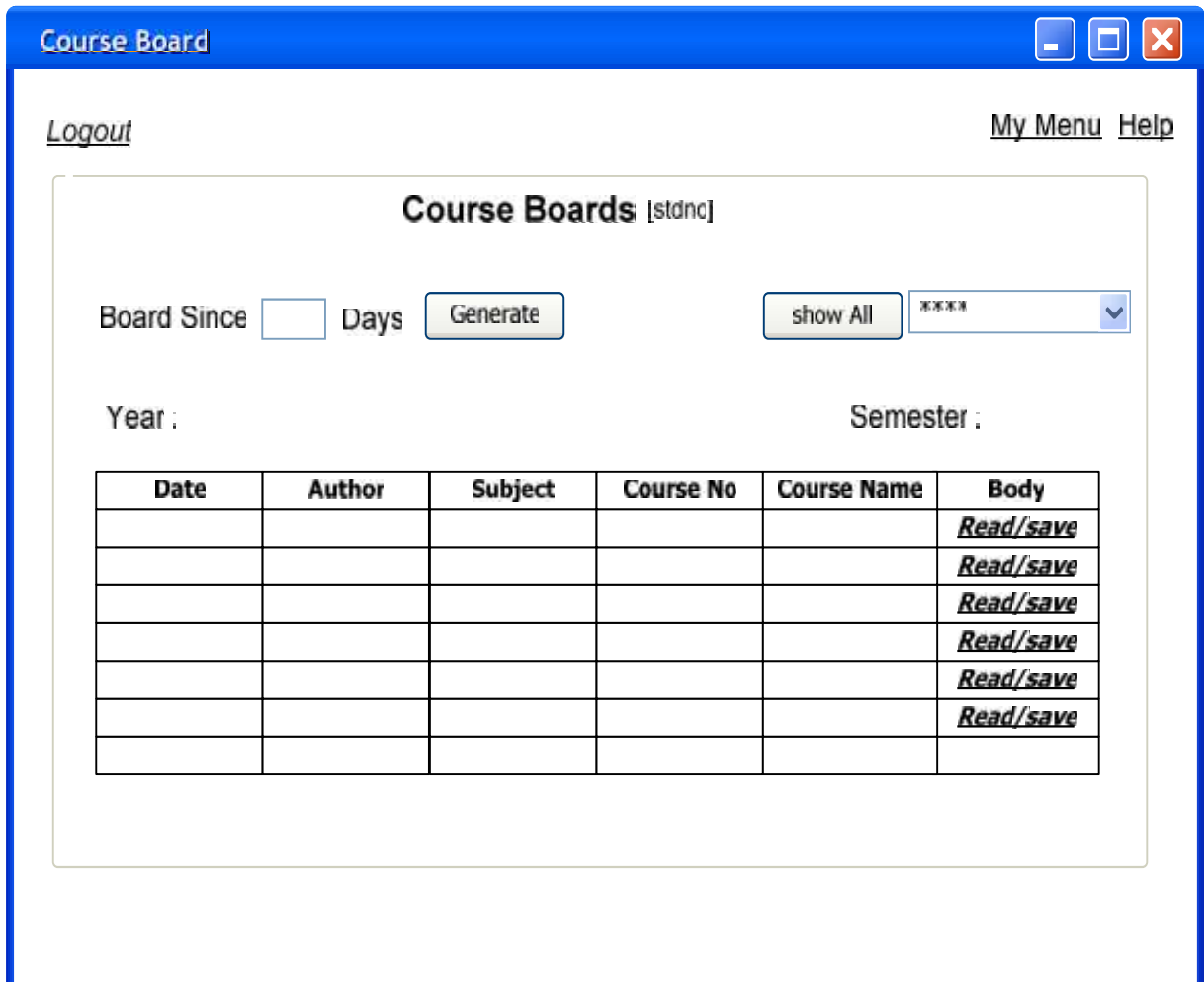


Figure (3.73) course boards.



➤ Department Boards :

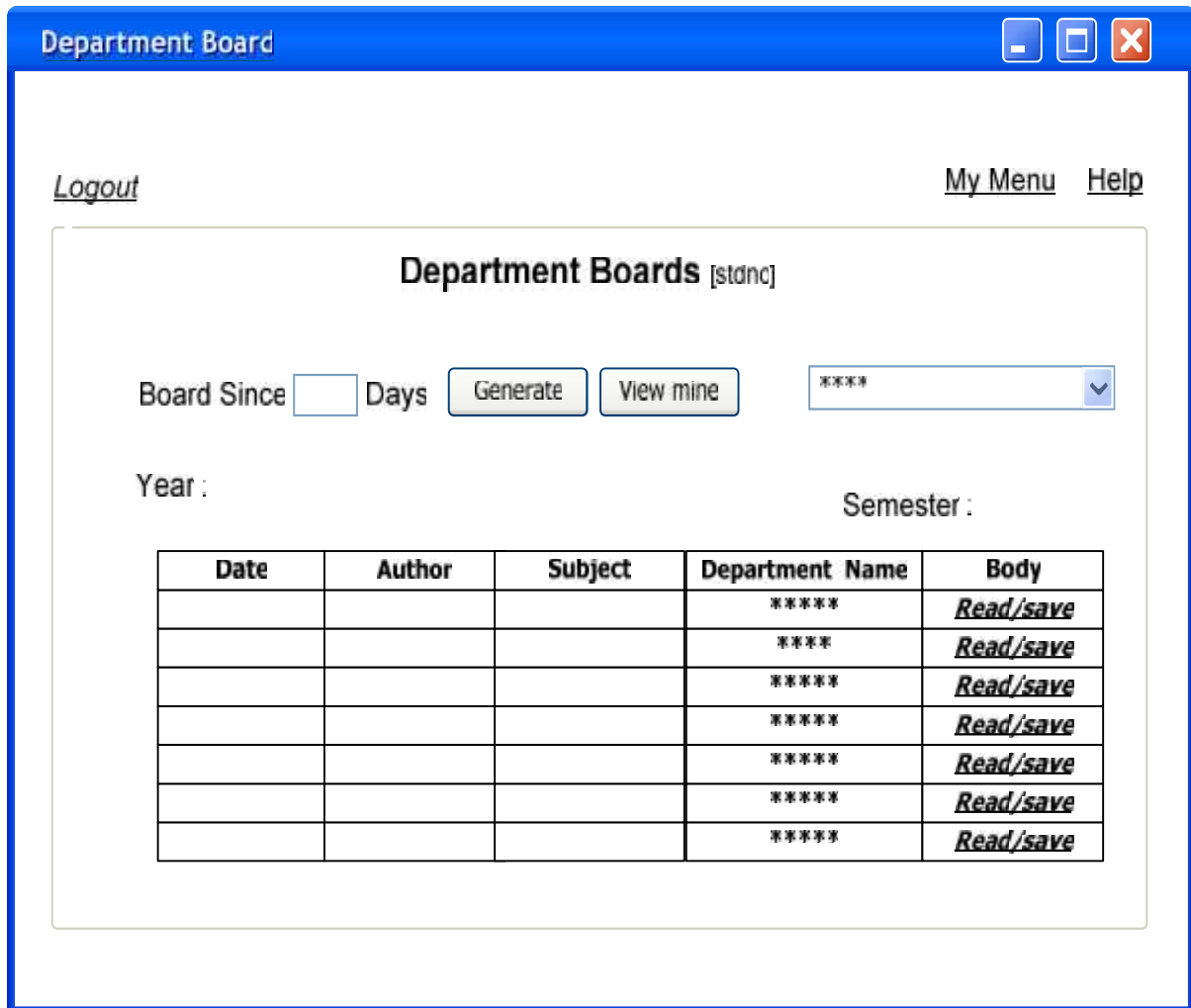


Figure (3.74) Department boards.



➤ College Boards :



Figure (3.75) College boards.



B. Administrator interface design :

a.Administrator Input / output design:

- Login administrator :



Figure (3.76) Administrator login screen.



➤ Insert boards :

The screenshot shows a web application window titled "Insert Boards". Inside the window, there is a form titled "Insert New Boards Form". The form is divided into several sections:

- Semester :** and **Year :** labels for identifying the board's context.
- Board Data:** A section containing input fields for "Date:", "Due to:", "Title:", "Board no:", and "Author:".
- Board body:** A section with two checked checkboxes: "Exit" and "New". Below them is a "Select board type" section with radio buttons for "Bulletin board", "College board", "Department board", "Course board", and "Student board".
- Board text:** A large text area for entering the board's content, with a "Browse" button below it for file selection.
- Upload the board:** A button at the bottom right of the form to submit the data.

Figure (3.77) Insert new boards.



➤ Delete boards :

Delete boards

Delete boards Form

Delete boards According

Expired boards
 Board name
 Board type

Get Boards

Target Board

Boards to delete

>
<
>>
<<

Empty

Delete

Figure (3.78) Delete boards.



➤ Update boards :

The screenshot shows a software window titled "Update boards:". Inside the window, the text "Update boards Form" is centered. Below this, there is a form with several sections:

- Available Boards Titles:** A dropdown menu.
- Update:** Two radio buttons: "Board & deadline" and "Deadline only".
- Board Type:** Two radio buttons: "Exit" and "New".
- Board Deadline:** A text input field followed by a "Change" button.
- A large empty text area with a vertical scrollbar on the right and a horizontal scrollbar at the bottom.
- A text input field followed by a "Browse" button.
- A "Update" button at the bottom right.

Figure (3.79) Update boards.



➤ insert mark :

Select the course number 2423

Course name : -----

Student no	Course no	Course name	Mark	Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit

Figure (3.80) Insert Marks.



➤ Insert college :

Insert College

Insert college form

Description type

Exit

New one

College description

College nd :

College Name :

Figure (3.81) Insert new college.



➤ Insert Major :

The screenshot shows a web application window titled "Insert major". The main content area is titled "Insert major form". It is divided into several sections:

- Description type:** Contains two radio buttons: "Exit" and "New one".
- Major no :** A text input field.
- Major Name :** A text input field.
- Credit H.:** A text input field.
- Outline year:** A text input field.
- Department name:** A dropdown menu.
- Major description/outline:** A large text area with a vertical scrollbar and a horizontal scrollbar. Below it is a "Browse" button.
- Upload major:** A button located at the bottom right of the form.

Figure (3.82) Insert new major.



➤ Questioner analysis

The screenshot shows a software window titled "Financial balance" with a blue header bar. Inside the window, the title "Questionnaire Form" is centered. Below the title, there are three columns of data:

- Result static**: A table with three rows: "Yes" with four dashes, "No" with two dashes, and "Maybe" with four dashes.
- Result percentage**: A table with three rows: "Yes" with four dashes and a percent sign, "No" with two dashes and a percent sign, and "Maybe" with four dashes and a percent sign.
- Total Balance**: A table with three rows: "Yes" with four dashes, "No" with two dashes, and "Maybe" with four dashes.

Below these columns, there is a label "Questioner text" on the left and a "Total Sharing" label followed by an empty text input box. To the right of the input box are two buttons: "Upload" and "New". At the bottom of the window is a large, empty text area with a vertical scrollbar on the right side.

Figure (3.83) Questionnaire analysis.



➤ control panel :

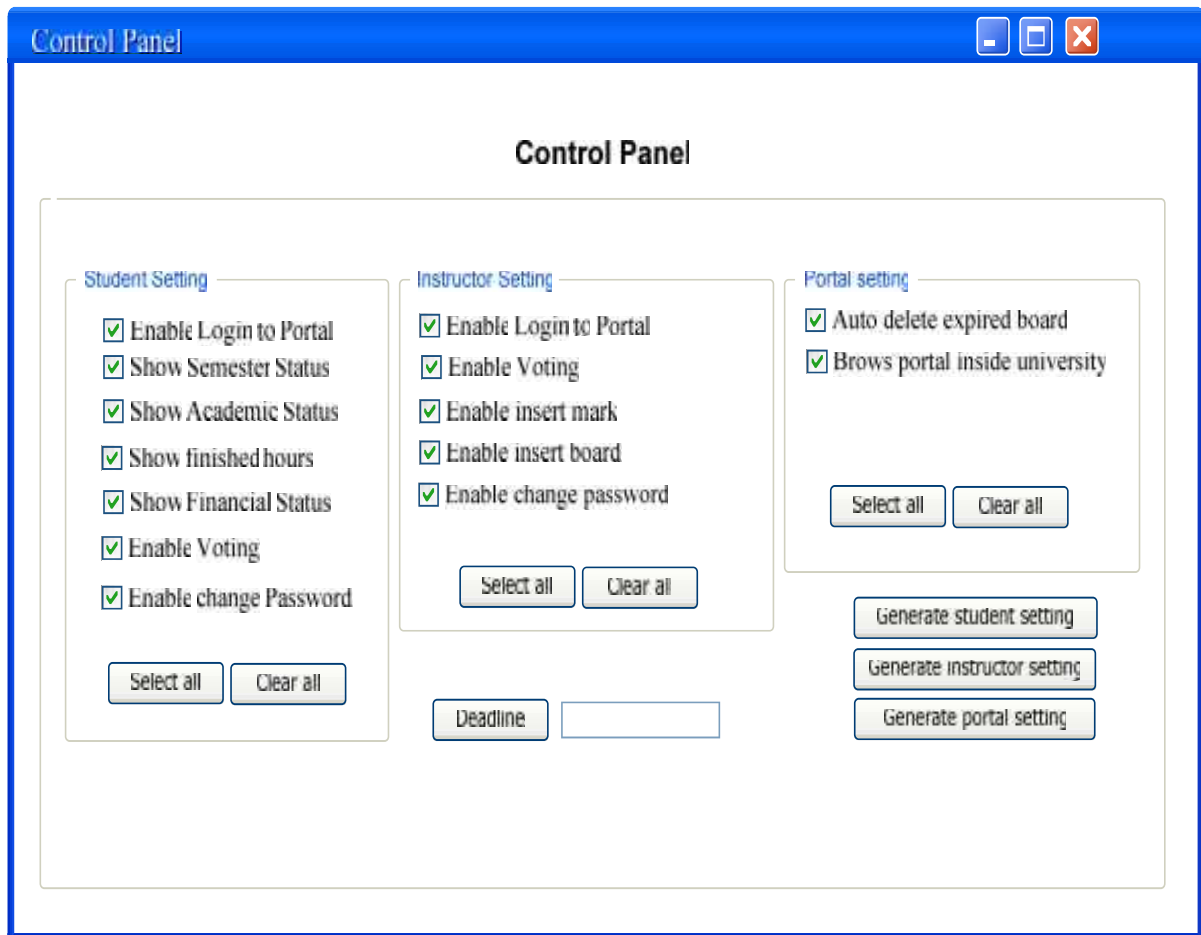


Figure (3.84) Control panel.



➤ User accounts :

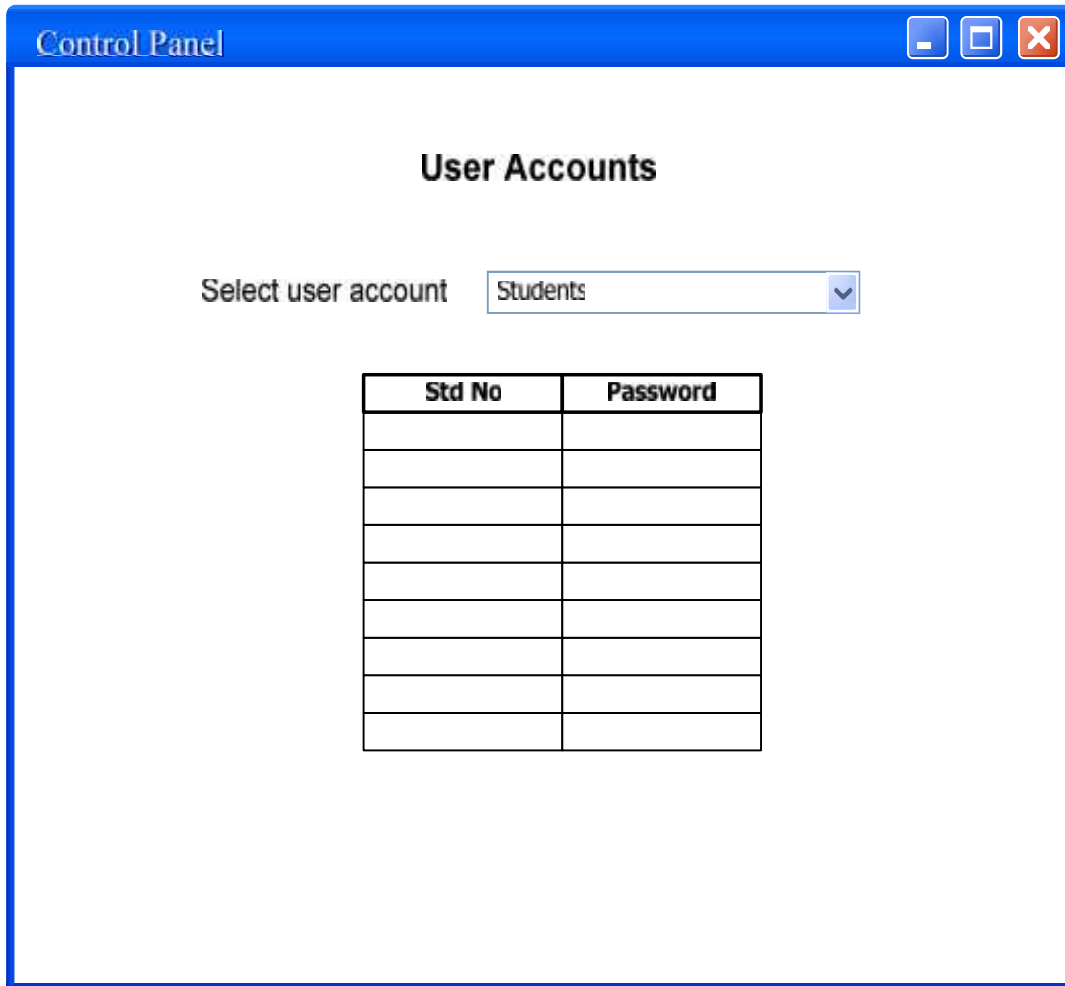


Figure (3.85) User accounts.



➤ change password :

Change Password

Change Password

Old Password

New Password

Confirm it

Update password

Figure (3.86) Change password.



C. Instructor interface design :

a. Instructor Input / output design:

➤ Insert board :

The screenshot shows a software window titled "Insert Boards" with a blue title bar. Inside the window, the main heading is "Insert New Boards Form". At the top, there are two labels: "Semester :" and "Year :". Below these is a section titled "Board Data" containing five input fields: "Date:", "Due To:", "Title:", "Board no:", and "Author:". Underneath the "Board Data" section is a "Board body" section. On the left side of the "Board body" section, there are two checked checkboxes: "Exit" and "New". Below these is a "Select board type" section with two radio button options: "Course board" and "Student board". On the right side of the "Board body" section, there is a "Board text" section featuring a large text area with a vertical scrollbar and a horizontal scrollbar. Below the text area is a "Browse" button. At the bottom center of the "Board body" section is an "Upload the board" button.

Figure (3.87) Insert new board by instructor.



➤ insert mark :

Student no	Course no	Course name	Mark	Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit
				Edit

Figure (3.88) Insert marks by instructor.



➤ My regular student :

Course no	Section	Student no	Student name

Figure (3.89) Instructor regular students.



D. Public interface design :

a. Public Input / output design:

➤ Search :

The image shows a search engine interface within a window titled "Search". The interface includes a label "Search For :", a dropdown menu currently showing "Course", an empty text input field for the search query, and a "Search" button.

Figure (3.90) Search engine.



3.4 Database design:

1. Students:

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric(9)	Yes	PK		Student Number
Std_fname	Varchar(50)	Yes			Student first name
Std_sname	Varchar(50)	Yes			Student second name
Std_tname	Varchar(50)	Yes			Student third name
Std_lname	Varchar(50)	Yes			Student family name
Birth_d	Nvarchar(50)	No			Student birth date
Birth_p	Varchar(50)	No			Student birth place
nationality	Varchar(50)	No			Student nationality
Social_s	Varchar(50)	No			Student social status
Std_id	Numeric(9)	Yes			Student identity number
Tawj_avg	Float(8)	Yes			Student taw. Avg.
Sex	Varchar(50)	No			Student gender
Address	Varchar(50)	Yes			Student address
Tel_no	Numeric(9)	Yes			Student tel. no.
Mjr_no	Numeric(9)	Yes	FK	Majors(mjr_no)	Student major number
Ac_no	Numeric(9)	Yes	FK	Academic_status(ac_no)	Student Academic status number
Ta_no	Numeric(9)	Yes	FK	Taw_branch(ta_no)	Student tawjihi number
pic	Nvarchar(50)	Yes			Student picture

Table (3.1) Students Table.



2. Academic status:

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Ac_no	Numeric (9)	Yes			Academic number
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Warning	Varchar (50)	No			Number of warnings
dismiss	Varchar (50)	No			Dismissed or not
Deelay	Varchar (50)	No			Delayed or not
Std_level	Numeric(9)	Yes			Student level
Regular	Varchar (50)	No			Regular or not
Graduated	Varchar (50)	No			Graduated or not
Hours_reg	Numeric (9)	Yes			Number of registered hours
Cwh_p	Numeric (9)	No			Community work hours
Sem_no	Nvarchar (50)	Yes			Semester number
Hours_p	Numeric (9)	No			Number of passed hours
Ac_year	Numeric (9)	Yes			Academic year
St_no	Numeric (9)	Yes	FK	Study_system (st_no)	
Std_avg	Float (8)	No			
Hours_p	Numeric (9)	Yes			

Table (3.2) Academic status Table.

**3. Boards:**

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>key</i>	<i>References</i>	<i>Description</i>
brd_no	Numeric (9)	Yes	PK		Board number
Brd_date	Datetime(9)	Yes			Board issue date
Due_to	Datetime(9)	Yes			Board due to date
Bodyt	Varchar (50)	No			Text body
Bodyb	Varchar (50)	No			Attachment body
Auther	Varchar (50)	No			Auther name
Subject	Varchar (50)	No			Title of the board

Table (3.3) Boards Table.

4. Colleges Boards:

<i>Field</i>	<i>Data Type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Col_no	Numeric (9)	Yes	PK		College number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.4) College boards Table.

5. Departments boards:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Dpt_no	Numeric (9)	Yes	PK		Department number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.5) Departments boards Table.

6. Courses Boards:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
crs_no	Numeric (9)	Yes	PK		Course number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.6) Courses boards Table.

**7. Students Boards:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
std_no	Numeric (9)	Yes	PK		Student number
Brd_no	Numeric (9)	Yes	PK,FK	Boards(brd_no)	Board number

Table (3.7) Students boards Table.

8. Colleges:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Col_no	Numeric (9)	Yes	PK		College number
Coll_name	Varchar (50)	Yes			College name
descr	Nvarchar (500)	No			College description

Table (3.8) Colleges boards Table.

9. Courses:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
crs_no	Numeric (9)	Yes	PK		Course number
Crs_name	Varchar (50)	Yes			Course name
Crđ_hours	Numeric (9)	Yes			Course credit hours
descr	Nvarchar (500)	No			Course description

Table (3.9) Courses Table.

10. Courses_type:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Co_no	Numeric (9)	Yes	PK		Course number
Co_t	Varchar (50)	Yes			Course type

Table (3.10) Course_Type Table.

**11. Passed courses:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK,FK	Students(std_no)	Student number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_Courses(crs_no)	Course number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes	PK		Academic year
Class_no	Numeric (9)	Yes			Section number
Mark	Float (8)	No			Course mark
Co_no	Numeric (9)	Yes	FK	Courses_type(co_no)	Course number

Table (3.11) Passed courses Table.

12. Offered_courses:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	courses(crs_no)	Course number
Class_no	Numeric (9)	Yes	PK		Section number
Time	Nvarchar (50)	Yes			Lecture time
Room_no	Numeric (9)	No			Room number
Sem_no	Nvarchar (50)	Yes	PK		Semester number
Ac_year	Numeric (9)	Yes			Academic year

Table (3.12) Offered courses Table.

**13. Major courses:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Crs_no	Numeric (9)	Yes	PK,FK	Courses(crs_no)	Course number
Mjr_no	Numeric (9)	Yes	PK,FK	Majors(mjr_no)	Major number

Table (3.13) Major courses Table.

14. Departments:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Dpt_no	Numeric (9)	Yes	PK		Department number
Dpt_name	Varchar (50)	Yes			Department name
Descr	Nvarchar (500)	No			Department description
Col_no	Numeric (9)	Yes	FK	Colleges(col_no)	College number
Cwh_r	Numeric (9)	No	PK		Number of required community work hours
Ac_no	Numeric (9)	Yes	FK	Academic status(ac_no)	Academic year

Table (3.14) Departments Table.

15. Employee classes:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK,FK	Employee(emp_no)	Employee number
Crs_no	Numeric (9)	Yes	PK,FK	Offered_courses(crs_no)	Course number
Sem_no	Nvarchar (50)	No			Semester number
Class_no	Numeric (9)	Yes	Pk		Section number

Table (3.15) Employee classes Table.

**16. Employee:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK		Employee number
Emp_name	Varchar (50)	Yes			Employee name
Dpt_no	Numeric (9)	No	FK	Departments(dpt_no)	Department number
St_no	Numeric (9)		FK	Study_system(st_no)	Study system (scientific degree)
Id_no	Numeric (9)	Yes			Identity number

Table (3.16) Employee Table.

**17. Financial status:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Fin_no	Numeric (9)	Yes			Financial account number
assistant	Float (8)	No			Assistant value
scholarship	Float (8)	No			Scholarship value
others	Float (8)	No			Other financial aids
credit	Float (8)	Yes			Student surplus balance
debit	Float (8)	Yes			Student deficit balance
Std_no	Numeric (9)	Yes	PK		Academic year

Table (3.16) Financial status Table.

18. Administrator login:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
username	Nvarchar (50)	Yes	PK		Administrator username (id)
pwd	Nvarchar (50)	Yes			Administrator password

Table (3.17) Administrator login Table.

19. Students login:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		Student number (id)
pwd	Varchar (50)	Yes			Student password

Table (3.18) Students login Table.

**20. Instructor login:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Emp_no	Numeric (9)	Yes	PK		Employee number
pwd	Nvarchar (50)	Yes			Employee password

Table (3.19) Instructor login Table.

21. Majors:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Mjr_no	Numeric (9)	Yes	PK		Major number
Mjr_name	Varchar (50)	Yes			Major name
Hours	Numeric (9)	Yes			Major credit hours number
Descr	Nvarchar (500)	No			Major description
Dpt_no	Numeric (9)	Yes	FK	Departments(dpt_no)	Department number
outline	Numeric (9)	No			Major outline

Table (3.20) Majors Table.

22. Projects:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		student number
Proj_no	Int (4)	Yes			Project number
Proj_name	Varchar (50)	Yes			Project name
Proj_body	Nvarchar (500)	Yes			Project body
Emp_no	Numeric (9)	Yes	FK	employee(emp_no)	Employee number

Table (3.21) Projects Table.

**23. Setting:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Set_no	Int (4)	Yes	PK		Setting account number
Set_s	Int (4)	Yes			Setting status value
Set_date	Datetime (8)	No			Setting deadline date

Table (3.22) Setting Table.

24. Student style:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Std_no	Numeric (9)	Yes	PK		student number
Main_header	Varchar (50)	Yes			Main header image path
Name_header	Varchar (50)	Yes			Name header image path
Title_header	Varchar (50)	Yes			Title header image path
Box	Varchar (50)	Yes			Box image path
Center_box	Varchar (50)	Yes			Center box image path
Box_header	Varchar (50)	Yes			Box header image path
Centerbox_header	Varchar (50)	Yes			Center box header image path
logout	Varchar (50)	Yes			Logout image path

Table (3.23) Student style Table.

**25. Study system:**

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
St_no	Numeric (9)	Yes	PK		Study system number
St_name	Varchar (50)	Yes			Study system name

Table (3.24) Study system Table.

26. tawjihi branch:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Ta_no	Numeric (9)	Yes	PK		Taw. number
Ta_name	Varchar (50)	Yes			Taw. branch

Table (3.25) Tawjihi branch Table.

27. Vote:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Vote_no	Numeric (9)	Yes	PK		Vote number
Vote_body	Varchar (500)	Yes			Vote body text

Table (3.26) Vote Table.

28. Vote result:

<i>Field</i>	<i>Data type</i>	<i>Required</i>	<i>Key</i>	<i>References</i>	<i>Description</i>
Vote_no	Numeric (9)	Yes	PK,FK	Vote(vote_no)	Vote number
V_yes	Int (4)	No			Vote yes value (1)
V_no	Int (4)	No			Vote no value (1)
V_maybe	Int (4)	No			Vote maybe value (1)
who	Numeric (9)	Yes	PK	employee(emp_no)	Userid for who voted

Table (3.27) Vote result Table.



3.5 Test Plan:

Here we describe briefly the methodology that we have adapted to test the system, steps that will be followed in the system testing are described bellow:

Testing steps:

1- Unit and Module testing:

We will use the white and black test Box testing to ensure that each function or module will operate as expected, by inserting valid and invalid inputs to show how the system will handle it.

2- Sub-system testing:

In this stage we will test each sub-system individually to ensure that each sub-system is operates as expected and meet its requirements.

3- Integration testing:

The integration of all sub-systems will be tested so that to ensure that the subsystems work together properly as expected, and meets its requirements.

4- System testing:

The system with all subsystems and functions will be tested to ensure that it functions properly, it meets its specification, and show if there defects throw system running.



3.6 Programming Language and Coding

There are many languages that can be used to develop a system such ours, but the most two effective languages are JAVA language and ASP.NET using Microsoft Visual Studio .NET as the development tool, here we describe the advantages of each, and why our selection was on the ASP.NET (we describe the ASP.NET technology in details in chapter 1) :

1- Device Independent:

Both Java and ASP.Net are a device independent languages, this means that any user can open the page that written using theses two languages from any explorer without any additional components or drivers, because he will just receive an HTML code.

2- Security

Java and ASP.Net have a high level of security during transmitting data; they provide us with many algorithms and techniques.

In the ASP.NET there are a build in valuator that ensure the user's input before any generation on the server, so if there any unusual input the webpage it self will not return to the server.

In Java the programmer must do all algorithms and validations manually.

3- Server side

One of the most powerful advantages of the ASP.NET technology is that it do not need to make any efforts on the client side, all operations and functions will work on the server.

4.1 Introduction

In this chapter we will describe the process of coding and implementation of the system and describe the environment that we use it in these processes.

The development of our software system to work properly to meet its predetermined requirements on the internet architecture needs a set of certain software and hardware products found in a platform configured to be suitable for the deployment process.

This system is an internet application that depends on a number of technologies that need to be installed, maintained, and updated, continuously. There exist a large number of software development packages that belong to different companies such as Microsoft.

This system is built basically on a group of Microsoft technologies such as the MS SQL server 2000 and ASP.NET 2003, IIS, Microsoft internet explorer 6.0, and Microsoft windows XP.

We also included some other applications that serve for the user interface design such as Macromedia Flash MX, Adobe Photoshop 8.0, and PhotoImpact XL.

This chapter describes the packages of software and hardware used in the system for the operation phase and how they are installed and prepared for work, the database and web interfacing, and the user interface implementation.

4.2 Establishment of development environment

▪ *Hardware Environment*

For the system development we use the following hardware:

1. Three PC Pentium 4:
 - a. 2.40 GHz speed.
 - b. 256 MB RAM.
 - c. 40 GHz H.D.D
 - d. Monitor, mouse, and keyboard.
 - e. 52x MAX CD-ROM.
2. One printer.
3. Scanner.
4. Tow flash memory with 256 MB.

▪ *Software Environment*

For the system development environment we use the following software:

1. Microsoft Windows XP Professional edition with IIS web server and server extensions.
we use this platform or operating system because our system is built on Microsoft visual studio.NET technology , and this type of operating system is the best platform for this technology we can use.
2. Microsoft visual studio.NET 2003 with ADO.NET.

From this package we use ASP.NET technology which is a new powerful technology designed to create web applications, making complete error handling, and provide data access tools.

- Why we use ASP.NET technology?

We use it because of:

- Ease of use.
- Reduce development time.
- Multiple languages within a project.
- Integrated browser.
- Debugging support.
- Customizable interface.
- High compatibility with windows environment.
- High compatibility with database.

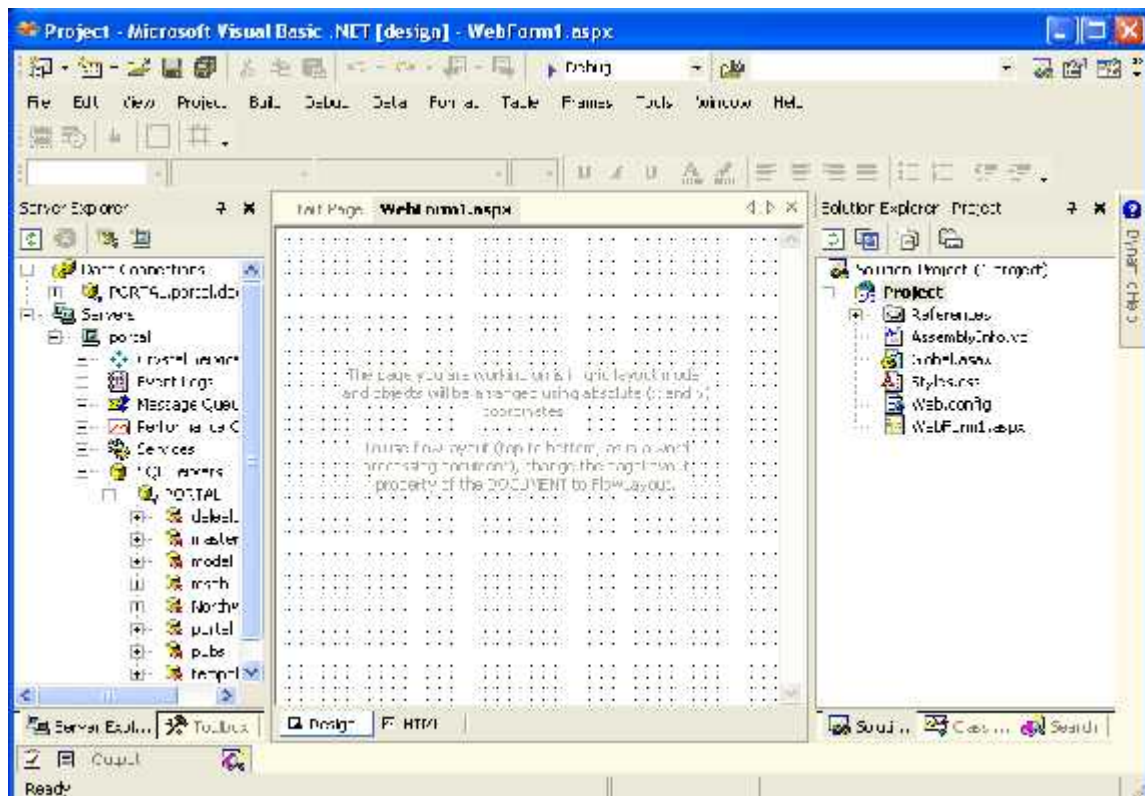


Figure (4.1) ASP.NET New Application Window.

To install visual studio.NET frame work, we need firstly to install a windows component which named IIS (Internet Information Services) on the windows by following these steps:

- ✓ Open control panel.
- ✓ Double click on add/remove programs icon.
- ✓ Click on add/remove windows components icon.
- ✓ Check the internet information system (IIS).
- ✓ Click on details button.
- ✓ Check options.
- ✓ Insert CD that labeled Windows XP Professional.
- ✓ Click next.
 - Setup will started to copy the required files to the hard disk.
- ✓ Click finish to close the windows components wizard.
- ✓ Close the add/remove programs window.

After IIS installation you can be able to install visual studio.NET program by using flexible integrated wizard.

3. Microsoft SQL server 2000.

An enterprise edition of the Microsoft SQL server 2000 is a good system to creating, accessing, and managing the database system of our application.

- Why we use SQL server 2000?
 - Ease of use.
 - High compatibility with ASP.NET.
 - High flexibility.
 - High security.
 - Ease of make a connection, query, and retrieve data.
 -

Unlike ADO, ADO .NET is specifically designed for data connections located in a disconnected environment, so it the best choice when developing and implementing internet based applications.

The major point here is that why SQL server 2000? And how to configure it after installation?

As we described in this section, that the connectivity and manipulation of the database in the .Net is provided by the ADO.Net technology, now because SQL server 2000 is integrated with the .NET technology tools, it certainly should be compatible for the access by ADO.NET. However, this integration appears to be more efficient and secure especially in such systems (web applications).

By configuring the SQL server 2000 to the windows only authentication mode, which is the preferred method to use when connecting a web application to SQL server 2000 DBMS, this method does not need any user name or password to be transferred back and forth between servers, only the confirmation that the user has been authenticated by trusted sources is required to process the database request.

The last configuration of the SQL server 2000 is to work effectively and ensuring that the integration more comfortable is to add a new account in the login group of the SQL server. This account (ASPNET) is created by the .NET framework and it should be added to the login group of the SQL server. Figure (4.3) shows how to add this account to the logins group of the SQL Server.

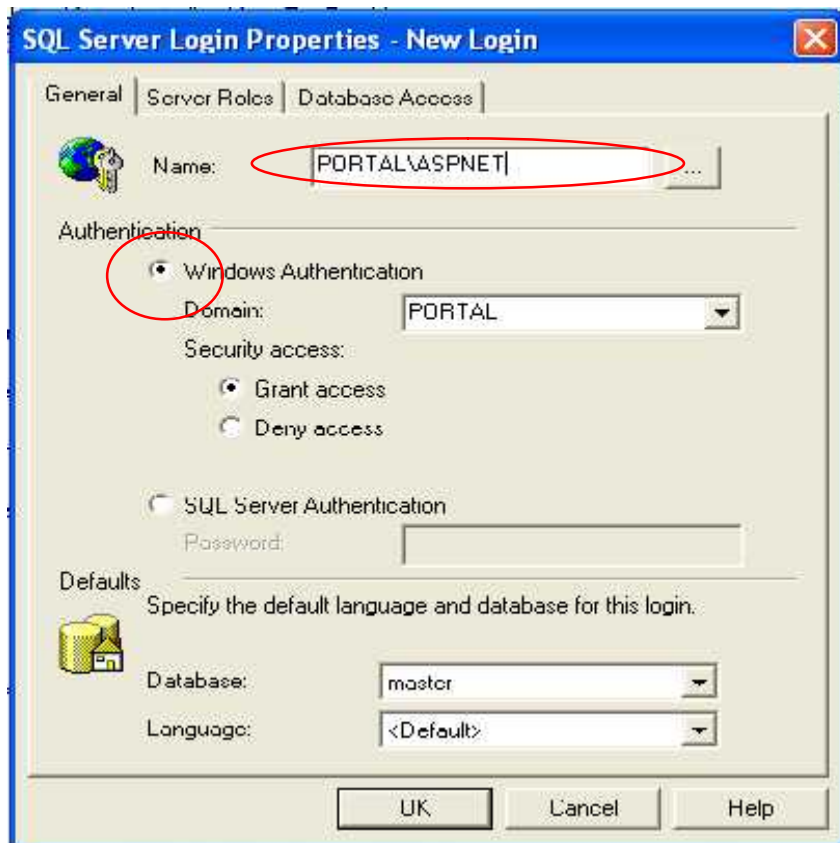


Figure (4.3) Adding ASPNET account on SQL Server logins group

4.3 Database implementation

The system database is implemented using the SQL server 2000 enterprise manager with the following properties:

- A. SQL server 2000 database name: portal.
- B. Database normalization, it normalized to insure that all tables and relations are not contained redundancy data and other database problems , so that it make it correct.
- C. Database creation : all keys which include, primary and foreign are created to ensure database consistency and correct relations.

D. Database connection

At this stage, the path to the data stored in the database could be opened and used as a two channel path.

In fact, to configure a connection, it is better to do it programmatically i.e. by coding, but using wizards that create connections is preferred at the starting point of the database connection establishment process.

The core here is whether a connection secure or not, but in configuring the SQL server 2000 we have chosen the windows only authentication, now in this stage we have to integrate the connection security with the SQL server 2000 security mode, thereby making the connection secure, this could be done by wizard, as it appears in figure (4.4) shown bellow, we select the integrated security option to be used when creating a connection to the database using Visual Studio.NET

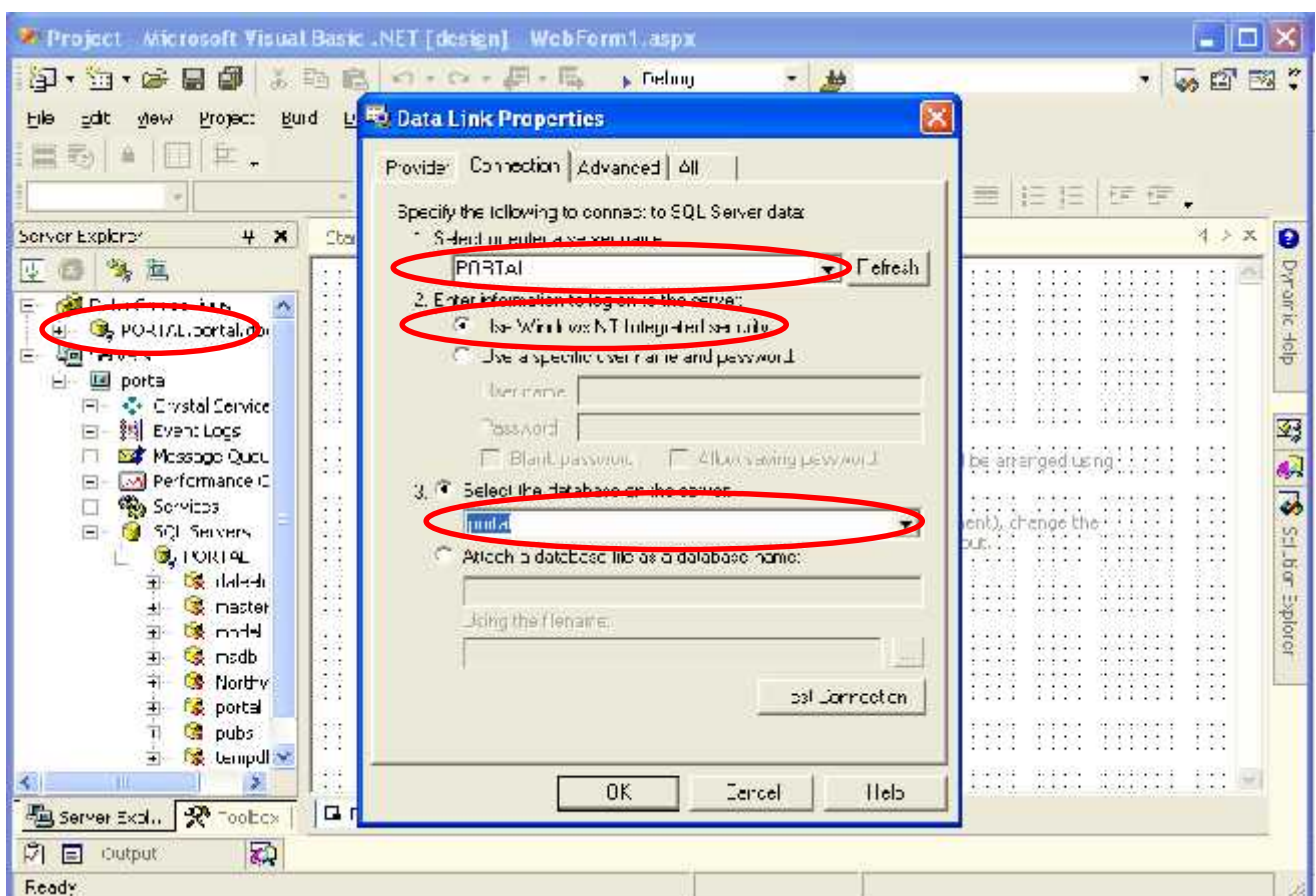


Figure (4.4) Creating Integrated windows security data link

After connect with database we can add, delete, update, and select data from database.

Particular example after making connection with database:

This example to write data to database:

❖ Code writes at ASP.NET code behind page:

- o Create SQL connection string:

```
Me.SqlConnection1.ConnectionString "workstation
id=PORTAL;packet size=4096;integrated
security=true;data source=PORTA" & "L;persist
security info=False;initial catalog=portal"
```

- o Definition of SQL command:

```
Dim cmd1 As New
SqlCommand("insertcolb",SqlConnection1)
cmd1.CommandType = CommandType.StoredProcedure
```

- o Definition parameters:

```
Dim p1 As New SqlParameter("@col_no",
SqlDbType.NVarChar)
Dim p2 As New SqlParameter("@brd_no",
SqlDbType.NVarChar)

p1.Direction = ParameterDirection.Input
p2.Direction = ParameterDirection.Input

cmd1.Parameters.Add(p1)
cmd1.Parameters.Add(p2)

cmd1.Parameters("@col_no").Value =
Session("colno")
cmd1.Parameters("@brd_no").Value = brdno.Text
```

- o Execute command:

```
SqlConnection1.Open()

cmd1.ExecuteNonQuery()

SqlConnection1.Close()
```

❖ Code write at SQL server 2000

```
CREATE PROCEDURE [dbo].[insertcolb]
@col_no numeric (9),@brd_no numeric (9)
AS
insert into brd_col values (@col_no,@brd_no)
GO
```

4.4 Supporting software :

Many other software tools where required to improve our system, these are used to support the appearance of GUI of our web application. So we used macromedia Flash 6.0, adobe Photoshop 8.0, and PhotoImpact 8.0. All of these software's are described precisely in chapter tow (Software Requirements Specifications).

4.5 Operating the system:

To operate the system properly. Many steps should be executed before the system operates as expected in its environment:

- Configuring the network (physical connection).
- Assigning appropriate IP addresses to the computer in the network.
- Setting up the .NET framework.
- Building the system Database.
- Creating the Database connection with security options.
- Setting up the system on the server (publishing through IIS).

To operate the system from development environment follow the below steps:

- ❖ From start menu select Microsoft Visual Studio.NET 2003.
- ❖ Then will appear window that ask you either to open existing project or open new project, select open existing project.
- ❖ Double click on project that named " PAP ".
- ❖ Then you can make running and brows the web.



Figure (4.5) Select Microsoft Visual Studio.NET 2003

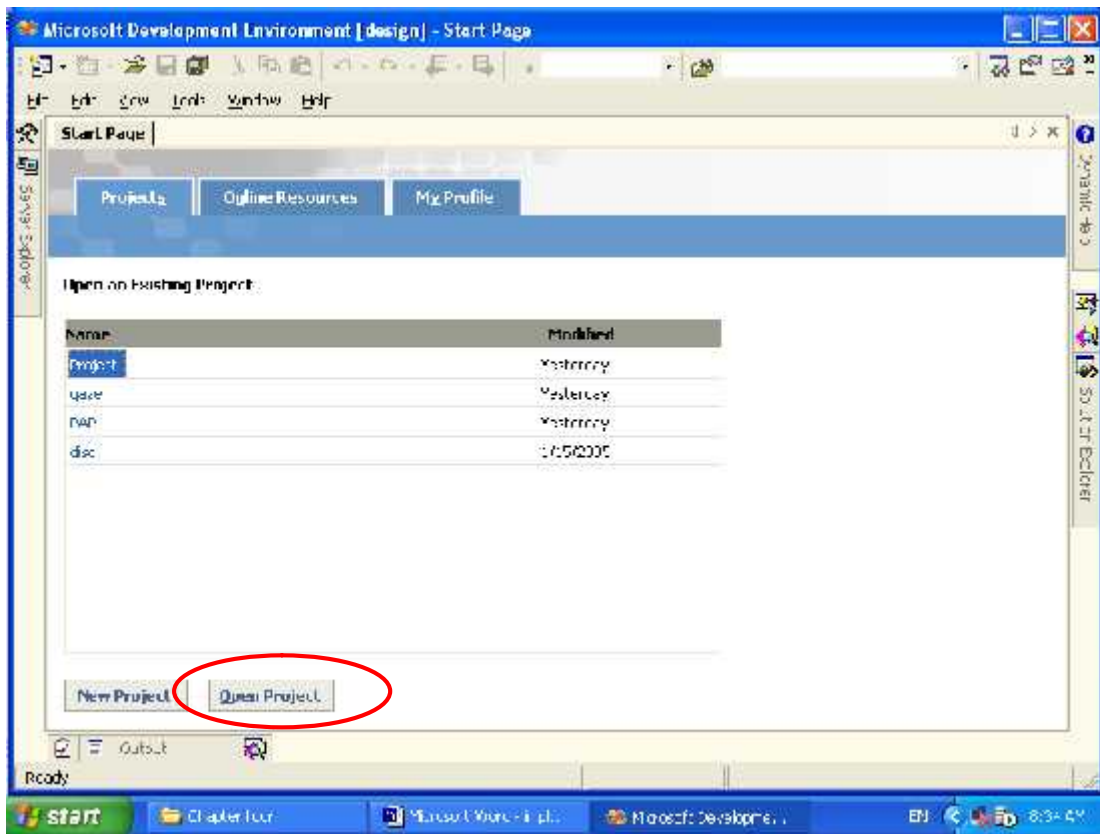


Figure (4.6) Select Open Project

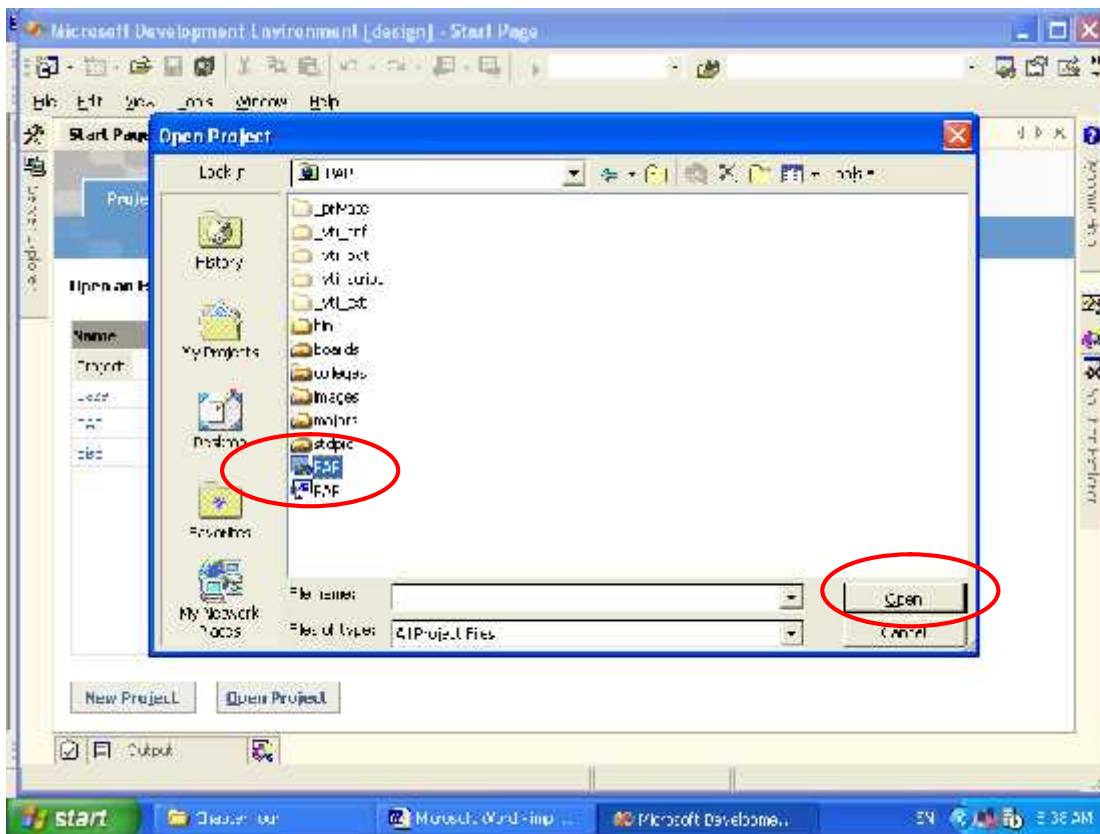


Figure (4.7) Select PAP project

- ❖ Or you can brows the web direct from Microsoft Internet Explorer 4.0 or later (6.0 is recommended).

Write the following URL on address text box in explorer then click enter:

<http://localhost/PAP/index.aspx>

4.6 Coding:

The principle improvements in programming are the reusability and code auto generation, this improvement and others reduce the time of programmer to write code. For development of a system, in our case a web-based application we used visual Studio.NET, which simplifies the development of powerful, reliable software application by providing familiar and shared development environment. It contain pre-built component, programming wizard, as well the ability to use component built using various language. In Visual Studio.Net there is a single integrated development environment (IDE), which provides a sense of what you see is what you get (the visual programming environment).

The usage of this tool for the purpose of programming and coding reduce the time and efforts and thereby increasing the performance.

When using Visual Studio.Net as a programming environment we gain the benefits of the separation between writing the logical code (the program functionality) from one side and the design of the appearance and graphical user interface (GUI) from the other side.

This application occurs by the new way that enables the programmer not to spend his time or determining programmatically where each control on a page should appears, however he could write the code that make the logic of each control operation in separate page called code behind page. We have appended the source code written manually for the main functionalities in our application.

4.7 Development process implementation.

The system was first created by going to Microsoft visual studio.NET development environment and named as PAP.

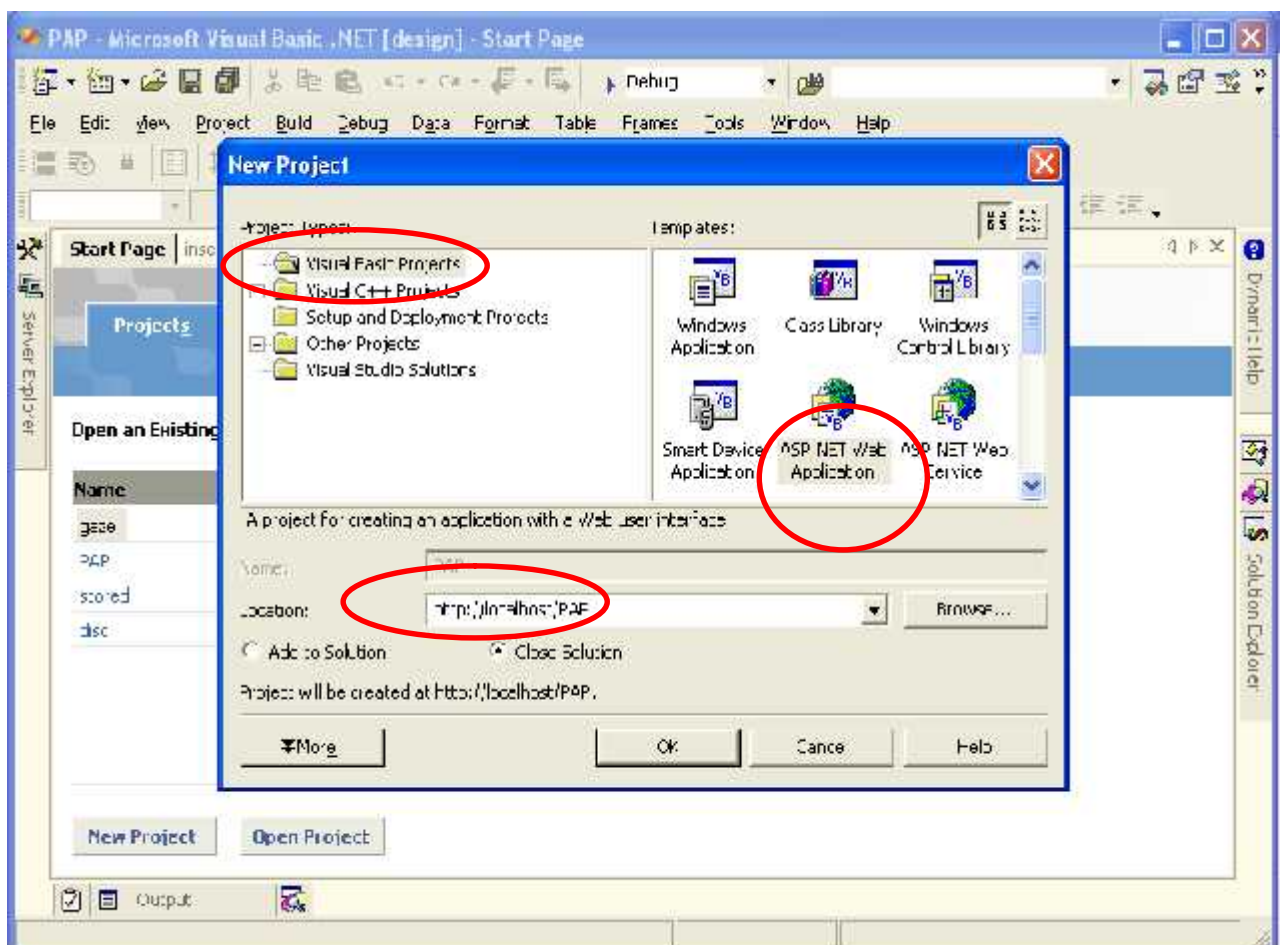


Figure (4.5) Development Process (create new project)

6.1 Introduction

At this chapter we will provide and explain the process and techniques that guidelines the system administrator to keep tracing and maintaining the system after running it.

At this stage, the system which had been developed and tested is needed to take place in the real working environment. However, it is a kind of joke to think about customers as if they were the developers of the system, they are just end users, so those users must be provided by sufficient information and guidance about system deployment and how it could be maintained. In this chapter, we describe the real working environment within which the system will operate, as well how it could be established, how it could be migrated, and how it is maintained.

In this chapter we will describe:

- Maintenance plan.
- Migration.

And we will talk about system maintenance which we divided to:

- IIS maintenance.
- SQL server 2000 maintenance.
- .NET framework maintenance.

6.2 Maintenance plan

In this section we will describe some procedures that should be taken in our consideration to handle any system failure, errors, and other types of problems and exceptions that maybe appear or occur during system life cycle.

1. Backup

Database is the nerve of the system, so it is the most important component of the system, because it contains all tables , views, and stored procedures that contain the data of the system, so we should keep this data from losing and harmful by specific means such as take a backup for this database periodically to enable maintain the system in the future.

This is an issue where the adaptation of the backup methodology (type and time interval) is determined and implemented by the company working on the system itself. The backup on the system database could be configured by means that are provided by the producer company of the DBMS that we have used in our software system which is Microsoft® on its DBMS product (SQL Server 2000).

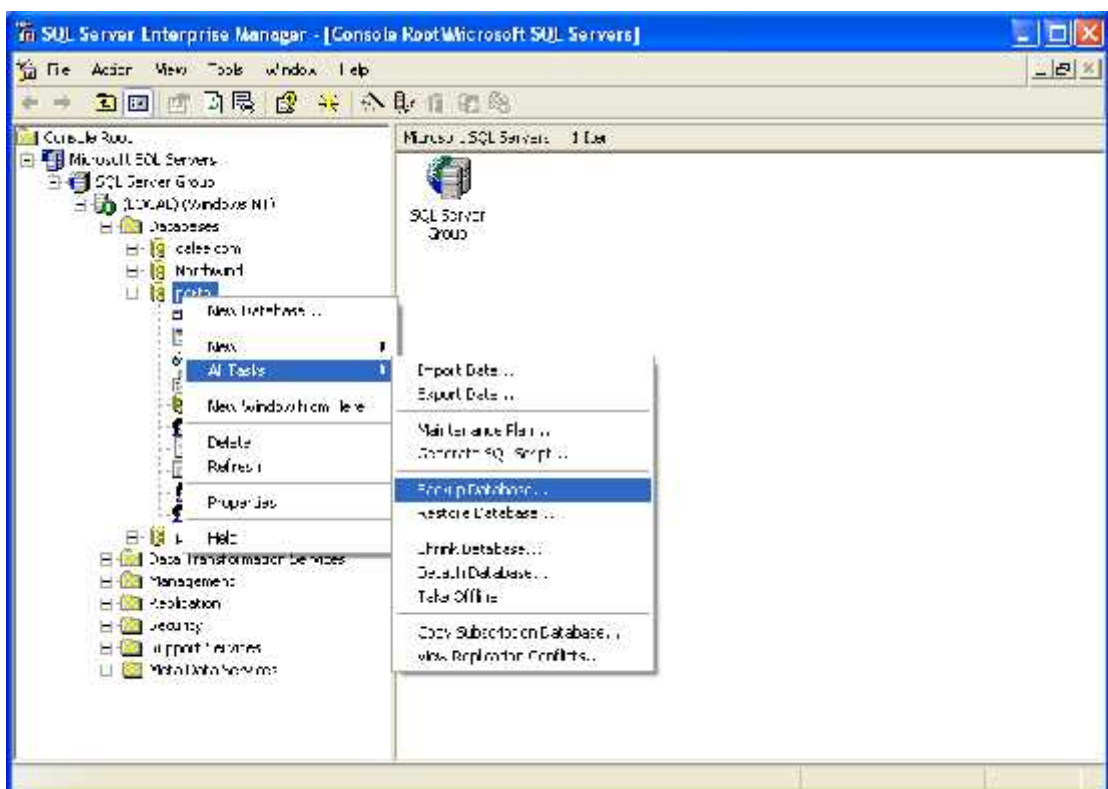


Figure (6.1) Backup Database

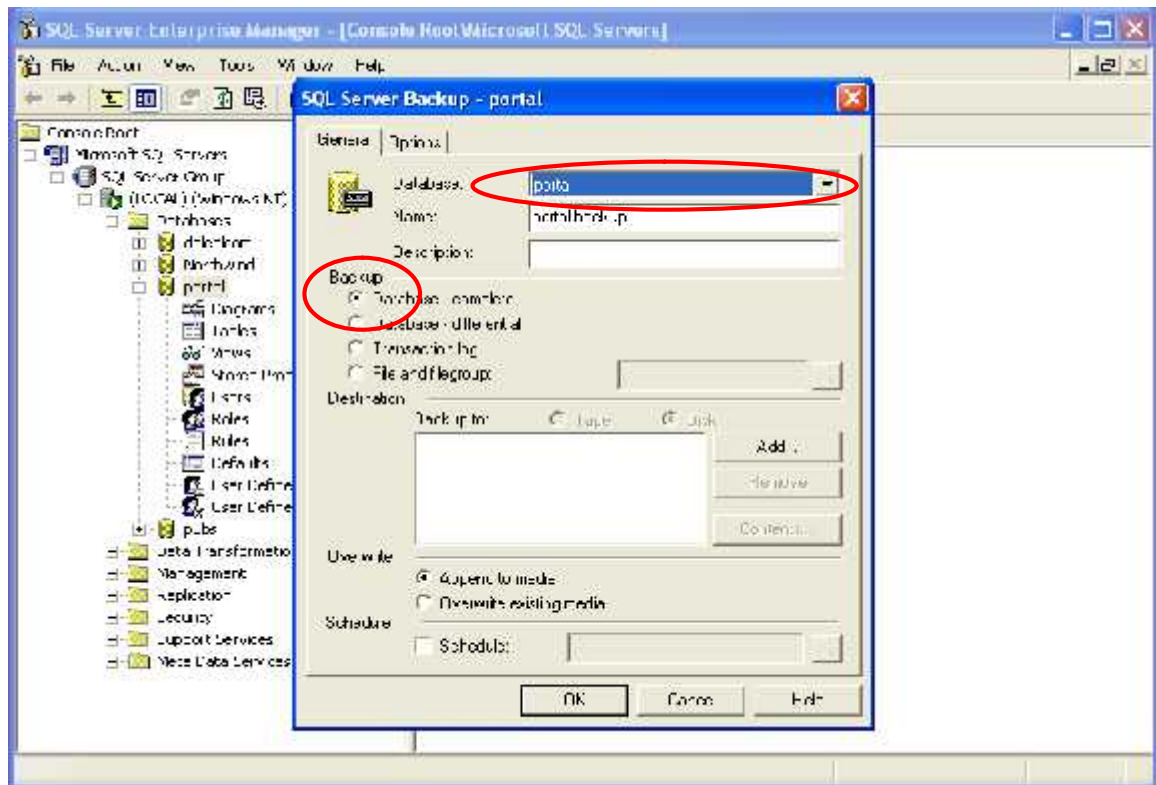


Figure (6.2) Backup options

2. Error reporting

When errors occur, certain actions are to be taken as the contract agreement describes the conditions and situations of the maintenance and the responsibility allocation on the contract sides, such as System Service Request (SSR) which take the following form in many cases:

System Service Request

REQUESTED BY ----- DATE -----

DEPARTMENT -----

LOCATION-----

CONTACT-----

TYPE OF REQUEST

URGENCY

- New system
- System enhancement.
- System error correction.

- Immediate, operations are impaired or opportunity lost.
- Problems exist, but can be worked around.
- Business losses can be tolerated until new system is installed.

PROBLEM STATEMENT

SERVICE REQUEST

IS LIAISON -----

SPONSER -----

----- TO BE COMPLETED BY SYSTEMS PRIORITY

BOARD -----

- Request approved Assigned to -----
- Recommend revision Start date -----
- Suggest user development.
- Reject for reason -----

6.3 Migration

The deployment of the system must be preceded by certain steps so that to work properly within its environment; the production environment has to be established, configured, and a decision of operating on the new system must be taken considering all constraints and risks of the process of migration to the new system. Toward deploying and migrating to the new system we describe here the steps that must be done:

1. Establishment of the production environment:

The minimal requirements of deploying the system are described in chapter one (system specification), and the needed configurations for the machine running the system are described in chapter four (Coding and implementation). For example, we say that our software system will not operate on a machine that doesn't have the .NET Framework, so that the company that decides to work on our system must have all of the production environment elements available.

2. Deciding to deploy the new system :

Here we say that our system was tested and we found that it works well as it should be, the system can operate immediately whenever a suitable production environment is created, but we say that our software is an integrated system that must run with its all parts integrated, nevertheless the system may not work properly. Now, the way of migration to it depends on the technicians and managers recommendations in the company that wishes to deploy this new system.

3. Running the system

After the system being complete it can be running.

6.4 Internet Information Services maintenance.

IIS is the core of web server components , since without it the server cannot be consider as web server ,so it must be maintained to make system operate correctly, because IIS represent a gateway for the application to be published over the internet.

Without IIS we can't publish the web over the internet , so success publish process is dependent on the consistency, security , and make a correct configuration for IIS.

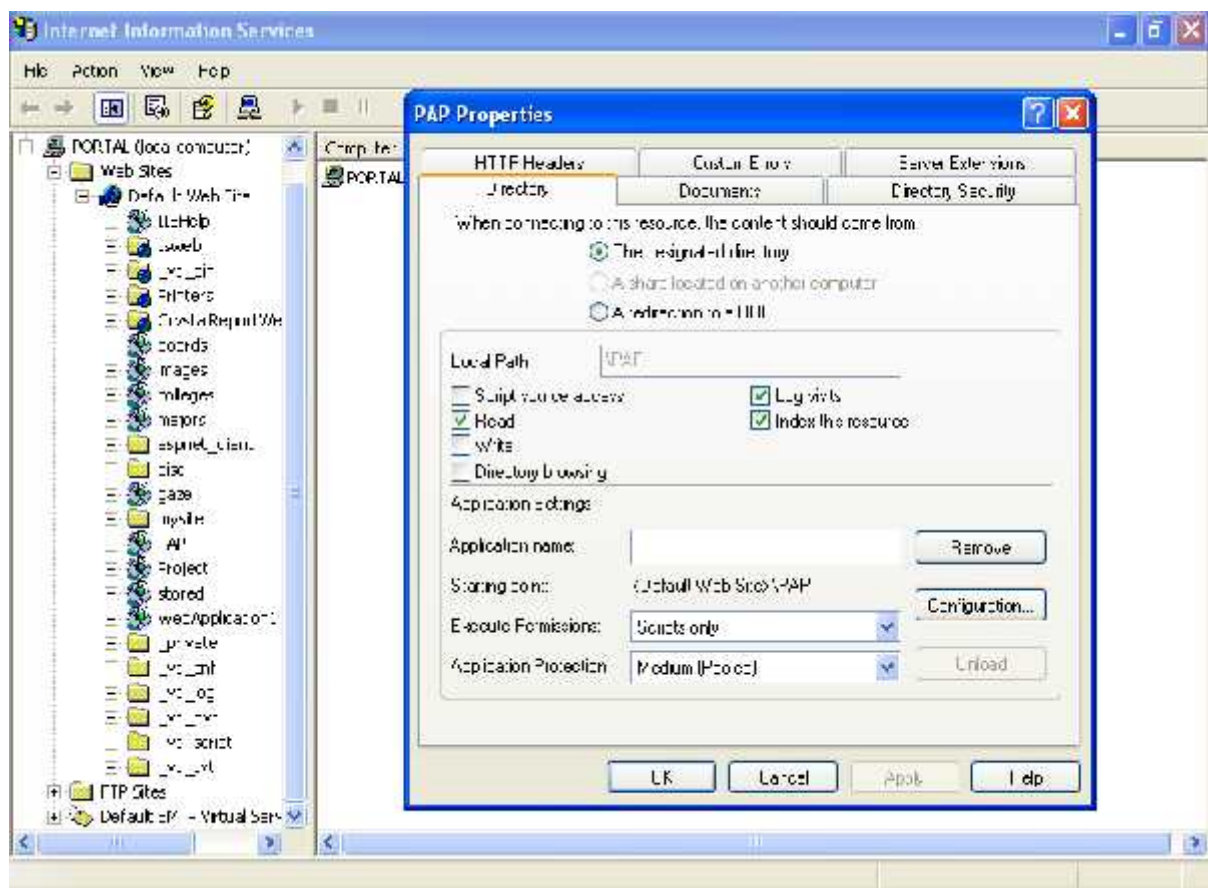


Figure (6.3) IIS Window

6.5 SQL server 2000 maintenance

The most important part of the system is database, since it contain the data used in the system, so it must be very secure much as better, so we must keep a track with it by maintaining database by using specific techniques as we mention before like backup .

Main components in SQL server 2000 according our system

- a) The database of " portal"

Which include all the system tables, views, and stored procedures.

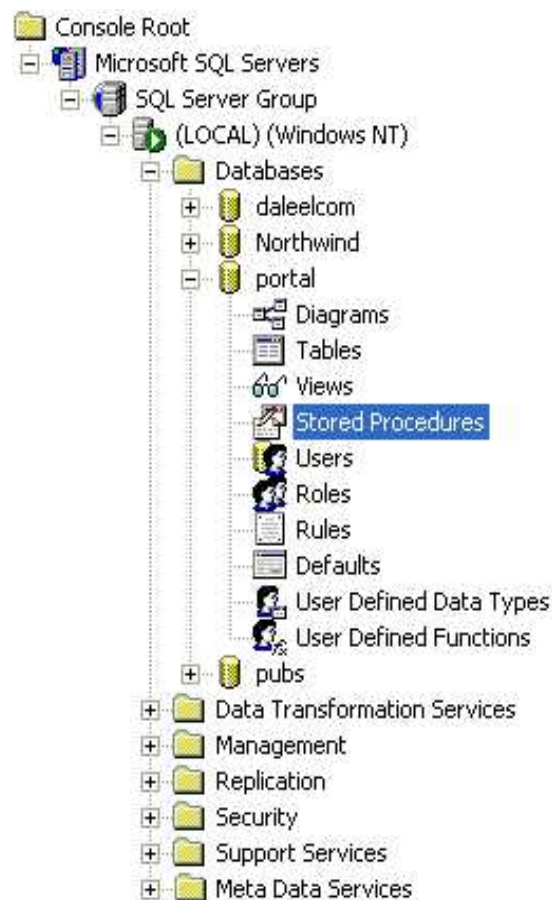


Figure (6.4) SQL server database

b) Security

This section include all roles that implemented on database, and users with their privileges allowed for them, and you can personalize your system setting by creating new users and roles and give each user specific privileges to allow access on database .

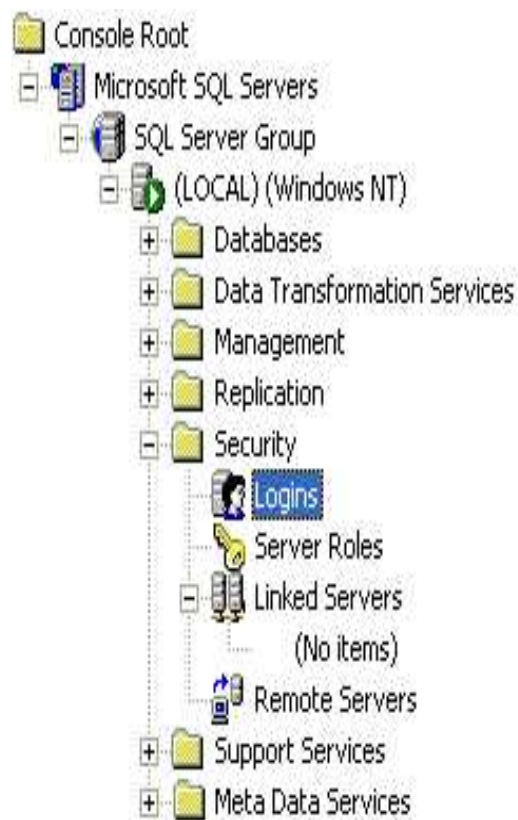


Figure (6.5) SQL server 2000 security section

6.6 *The .NET Framework maintenance*

.NET framework is an infrastructure of the Microsoft .NET technology, which we use to development our system one of it's components (ASP.NET).

Visual studio.NET collect all project files in one consol window known as solution explorer, through this window administrator can pick any file and make any justification on it or updated it.

If any error or problem was occur and cause system failure or it damage the visual studio.NET program, we can repair this problem by inserting the .NET framework CD and pick a repair choice , and the system will be repaired, or we can install new refresh copy from .NET framework.

Recommendations and future work

▪ *Recommendations :*

- ❖ We recommend to connect our Portal System with registration database, thus we work on a virtual database to some extent similar to real registration database, because we can't obtain the real skeleton of the registration database.
- ❖ We recommend to expand our system to include e-learning system and e-registration system to serve the students as possible.
- ❖ In addition to that, we recommend to publish our system on ppu.edu domain.
- ❖ We strongly recommend to build e-library database system and connect it with portal, in order to simplify student search transactions and reading.
- ❖ Build university forum that create virtual interactive area for only university students to exchange their opinions and ideas about given problem.
- ❖ Provide students with available accommodations .

▪ *Future Work:*

- ❖ Multiple language interface supported by Portal (English, Arabic).
- ❖ Online help for potential problems that students might face.
- ❖ Strengthen the security during student Portal connections, by using more encryption algorithm techniques.
- ❖ Build e-magazine that receive participations only from university students.
- ❖ Build e-library for programs that serve different students in different majors.

Chapter two:

Software requirements specifications

2.1 Introduction:

This chapter will explore in detail the system requirements specifications, functional description, validation criteria that contain data flow diagram (DFD), data structure presentation, data dictionary and database requirement.

2.2 Requirements specifications:

A. Student requirements specifications:

- The student requests the PAP site from web server using internet explorer.
- The students insert his username and password at log in area in the log in Page.
- If the student inserts a valid username and password he will see the student main menu that contains the following items:

1. The student My Profile:

By clicking on My Profile in students main menu the student will show a new page that contains his information as follow:

- Name information: first, second, third and final name.
- Address information: address and telephone number
- Academic information: academic number, scientific degree, major and outline.
- Add my picture: if the student wants to insert his picture to database for using it in the portal.

2. the student current schedule:

By clicking on the current schedule, the student can see the following :

- Course number.
- Current semester.
- Course name.
- Instructor.
- Section.
- Time.
- Room number.

3. the student financial status:

When the student chooses to click on the financial status he will see his related financial information that contain the following:

- Financial aids: aid, scholarship and other.
- Financial balance: credit and debit.
- Semester balance.
- Total balance.
- In addition to that semester classification and academic year information shown on the top of the page.

4. the student academic status:

This section shows in detail all academic information for each student from the beginning until the current semester. When the student chooses academic status from the main menu he will show the following report:

- Student full name.
- Birth date.
- Student number.
- Degree: defines the scientific degree of the student.
- Major: defines the student major.

- Tawjihi information: branch and average.
- Community work hours: show passed and remained work hours.
- Department: defines the student's faculty.
- College: specifies the student's college.
- Averages: displays accumulative and major averages.
- Academic alerts: specifies how many academic warnings the student has gained.
- Dismiss: this item tells in yes or no if the student dismissed or not.
- Delay: specifies if the student delay for the current semester or not.
- Level: defines the student level in semesters.
- Regular: specifies if the student is regular or not.
- Graduated: specifies if the student is expected to graduate or not.

5. Semester status:

If the student clicks on this link then he can see his marks only for the current semester only in a table that contains the following:

- Course number.
- Course name.
- Course mark.

In addition to that, he can see his semester average and if he honored or not.

6. Passed courses:

By clicking this link, the student can show all of his passed courses since starting his study until the current semester and he can display these courses in three different ways:

- By filtering courses according study year. So only the courses studied at that year displayed by choosing the year from dropdown list that shown automatically after clicking the year checkbox.
- By filtering courses according semester or according a combination of year and semester. So we can select the year and semester from

dropdown lists that shown automatically. After that, the passed courses can be displayed by applying the filtering condition.

- By filtering according the type of the course (college requirement, department requirement, elective, selective) and then display these courses in combination of type and year or type and semester or type and (year and semester).

The student can see some statistics such as:

- Accumulative average.
- Passed courses.
- Credit hours remained.

After performing any of previous filtering criteria, all resulted courses will be displayed in the table the table below. This table contains

- Course number.
- Course name.
- Mark.
- Course type.
- Credit hours.
- Semester.
- Academic year.

7. Student boards.

This type of boards devoted to offer special area to display each student boards only. So no other students can see these boards or memos. The board contains:

- Date.
- Author.

8. courses boards:

In this section the student can show all of his courses boards and he can filter these boards by their date and he can specify what the course board he want. Consequently, he can show it or show all the boards.

9. department boards:

Here we can tell the student about his department boards, the portal enables the student to pick his department and then show the board. Here the student he can put the number of days ago to see only the boards issued within this time period.

10. college boards:

here the student show his college related boards by selecting the college name from a dropdown list, and can show only the boards within a specified period of time.

11. change password:

From this area the student has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

12. General voting:

From the student main menu web form, student can show the question to be answered and he can show the questionnaire results and he can answer in yes, no or maybe.

13. New boards

In this section, the student can see all recent boards types related to him. By default these boards since three days ago and the student can insert the number of days ago to check the issued boards within this time period.

14. Social guide center.

This section devoted to social support system for students; here the administrator (social guide) can provide the students with solutions and suggestions and focus the light on sensitive problems that the students face in their academic life.

15. Documents.

From here the student can print his academic status with finished hours and marks, and he can print paper that insure that this students is regular students in the university.

B. administrator requirements specifications:

1. insert boards:

Provide administrator the ability to insert boards. In this section the administrator can insert new boards by providing the primary board data that include the following:

- Date.
- Due-to date.
- Board title.
- Board number.

- The author.
- The board text.

This section has many tools to facilitate the insertion process; if the board body saved at any location the administrator can brows it as any file instead of typing it, then he can specify the board type.

2. Delete boards:

Provides administrator the ability to delete boards. This form specialized to enable the administrator to delete expired boards that exceeded the specified due-to date.

By selecting the expired board's radio button, board name and type and then click on get boards' button to attach these boards in a dropdown list called target boards, after that he can put the intended boards to be deleted in a new list called boards to delete and insure the deleting process by clicking the delete button.

3. Update boards:

Provides the administrator the ability to update boards; This form used to make some editing to the boards by selecting the board name from the available boards and perform the update process to body and deadline date only. Finally, submitting the process by clicking change button.

4. Insert Marks:

Provides administrator the ability to insert marks. Here the administrator can show courses that offered at this semester, and then he select any course to get the sections of that course, and display them in drop down list, and then he select any section to view the students enrolled at that course, and then

select any student to insert new mark or update the existing mark related to him.

5. Insert new college.

This form used in the case of university expansion and a new college is opened. Here administrator can add a brief description for the new college and the college number determined automatically and he can insert the college name and click insert button to save the information in the database.

6. Insert new major.

This form enable the administrator to insert new major at any college or department, a brief description can be added and the primary major information can be added too such as major number, major name, credit hours and outline year, the process completed by clicking on the insert button to save data in the database.

7. Provides the administrator the ability to add general question for voting.

The student can answer in yes, no or maybe.

8. From the administrator main menu, administrator can provide students with the questionnaire results and voting analysis that often come from database and calculated automatically, the student can vote one time only.

9. Control panel.

From this link the administrator gain control over all the portal processes, links and forms, he can activate or deactivate any link or control in the students accounts.

10. Users accounts.

In this form administrator should select the PAP user type and then he will show all the authenticated users with their ID's and their usernames.

11. Change password.

From this area the administrator has the ability to insert his username and change his password and replace it by a new one and confirm it. Then click update button to complete the changing process.

C. Instructor requirements specification:

1. Insert Boards

Provide instructor the ability to insert his course's boards and his student's boards only. In this section the instructor can insert new boards by providing the primary board data that include the following:

- a. Date.
- b. Due-to date.
- c. Board title.
- d. Board number.
- e. The author.
- f. The board text.

This section has many tools to facilitate the insertion process; if the board body saved at any location the instructor can browse it as any file instead of typing it, then he can specify the board type.

2. Insert Marks

Here the instructor can show only courses that he gives, and then he select any course to get the sections of that course that he gives and display them in

drop down list, and then he select any section to view the students enrolled at that course, and then select any student to insert new mark or update the existing mark related to him.

3. My regular students.

In this form the instructor can view report that contain names, courses, and sections that related to instructor course, and he can print it.

4. Chang my password.

From this area the instructor has the ability to change his password and replace it by a new one and confirm it. Then click update button to complete the changing process.

D. Public requirements specifications:

This requirement does not depend on the student or administrator accounts on other words it is visible and accessed to any portal visitor. This requirement includes:

1. Log in area devoted for administrators and students that enter the portal by using valid username and password; in addition to that these can remember their password and username by clicking forget my password link.

2. Help: this link used to conduct the portal visitors how to benefit form services and facilities provided by the portal.

3. PAP links: this section includes:

- Bulletin boards: this link leads the user to view all public university boards, and he can show details about each board by clicking on read link.
- Get my account on line: here an authenticated user can retrieve his account by asking him some questions. The system asks the user to enter his username. If he authenticated, the system asked him some additional questions for supporting security, if his answers correct then the system gives him his account information.
- Academic calendar: in this area the user can see the user, this academic year calendar that contains all holydays and occasions during academic year.

4. Search engine:

This search engine limited to courses, colleges, majors, employees and boards, at the first, the student should select what he going to search from dropdown list then type a word or statement who looking for. Finally click on portal search button, this search limited on portal database.

5. PAP FAQs:

This link used to serve students with dynamic help that provided by portal specialists that including common Frequently Asked Questions and try to answer them.

6. General activities:

This area contains current general activities take place or expected to occur.

7. Honor board:

This corner displays scrolled marquee with links names of honored students.
To show details of each student you have to click on name link.

2.3 Functional description:

This section explores all the portal functions in detail.

1. The user requests the PAP home page from web server.

Function: user requests the page from web server.

Description: this function provides ability to browse the portal.

Input: the portal URL

Source: user

Output: Portal Home page.

Destination: web server.

Require: insert correct site address.

Pre-condition: availability of internet service.

Post-condition: displaying the portal Home Page.

2. Log in as student.

Function: student log in.

Description: enabling the student to access his account, see his information and use the student menu using valid username and password.

Input: student username and password

Source: student and log in form.

Output: The student main menu.

Destination: SQL database server.

Require: valid log in and correct username and password.

Pre-condition: Home Page and disconnected student.

Post-condition: connected student and student menu displayed.

3. Log in as administrator.

Function: log in as administrator.

Description: identifying the administrator by using valid username and password.

Input: administrator username and password

Source: administrator and log in form.

Output: administrator main menu.

Destination: SQL database server.

Require: valid log in and valid username and password.

Pre-condition: Home Page and disconnected student.

Post-condition: connected student and student menu displayed.

4. The student my profile.

Function: displaying the student information

Description: by clicking the my profile link from the main menu a new page will be displayed and contained the student information

Input: clicking my profile link from student main menu.

Source: the student and student menu web form.

Output: the student profile.

Destination: SQL database server.

Require: valid log in and single click on my profile item.

Pre-condition: Home Page and no student displayed in main menu.

Post-condition: the student can see his profile.

4.1 upload picture.

Function: upload picture.

Description: this function used to give the student the ability access the database for the purpose of saving his picture in database to be used by the portal administrator and show it with the student related activities such as honor board.

Input: click on brows button to specify image location.

Source: client primary or secondary memory.

Output: saved picture in the portal database.

Destination: SQL database server.

Require: single click on brows button.

Pre-condition: specifying the picture location.

Post-condition: saved picture in the portal database.

5. the student current schedule

Function: current schedule.

Description: this function enables the student to show his registered courses and their time schedule, in addition to room number, section and course instructor.

Input: clicking current schedule for student main menu.

Source: the student and student main menu web form.

Output: current schedule web form.

Destination: SQL database server.

Require: valid log in and single click on the current schedule item.

Pre-condition: Home Page and no schedule information displayed.

Post-condition: the student can see his current schedule.

6. financial status

Function: display student's financial status.

Description: this function displaying the student financial status by clicking on financial status he will show financial information related to the student.

Input: clicking financial status from the student main menu.

Source: the student and student main menu web form.

Output: financial status web form.

Destination: SQL database server.

Require: valid log in and single click on the financial status item.

Pre-condition: Home Page and no financial information displayed.

Post-condition: connected student and student menu displayed.

7. Academic status.

Function: displaying academic status.

Description: this function enables the student to show detailed academic report of academic information from the beginning until the current semester.

Input: clicking academic status from student main menu.

Source: the student and student main menu web form.

Output: academic status report.

Destination: SQL database server.

Require: valid log in and single click on academic status in student main menu.

Pre-condition: Home Page and no academic information displayed.

Post-condition: connected student and student menu displayed.

8. semester status

Function: displaying semester status.

Description: this function enables the student to see courses and their marks for latest semester.

Input: clicking semester status from student main menu

Source: the student and student main menu web form

Output: semester status report.

Destination: SQL database server.

Require: valid log in and correct username and password.

Pre-condition: Home Page and no semester information displayed.

Post-condition: connected student and semester status displayed.

9. Passed courses.

Function: displaying passed courses.

Description: this function enables the student to show his passed courses from the moment of entering the university until the latest semester.

Input: click on passed courses link from student main menu.

Source: student and his main menu web form.

Output: report of passed courses and some other information.

Destination: SQL database server.

Require: valid log in and single click on passed courses item.

Pre-condition: Home Page and no passed courses information displayed.

Post-condition: table of passed courses displayed.

9.1 Filtering criteria.

Function: select filtering criteria.

Description: this function used to show specific courses according some condition; such as filtering courses according study year. So only the courses studied at that year displayed by choosing the year from dropdown list that shown automatically after clicking the year checkbox.

Input: select filtering method in year, semester or course type and pick the value of selected method from dropdown list that becomes visible automatically.

Source: student and his passed courses web form.

Output: table of passed courses and some other information.

Destination: SQL database server.

Require: selecting filtering criteria.

Pre-condition: student menu web form.

Post-condition: table of passed courses displayed.

10. My boards.

Function: displaying student boards.

Description: this function specialized to offer special area that display each student boards only, so no other students can see these boards.

Input: click on my boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on my boards item in the students main menu.

Pre-condition: single click on my board's item in student's main menu.

Post-condition: table containing student boards

11. Course boards.

Function: displaying course boards.

Description: In this section the student can show all of his courses boards and he can filter these boards by their date and he can specify what the course board he want. Consequently, he can show it or show all the boards.

Input: click on course boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, author, subject, course name and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on course boards item in the students main menu.

Pre-condition: single click on course boards item in student's main menu.

Post-condition: table containing course boards

12. Department boards.

Function: display department boards.

Description: In this section the student can show all of his department boards and he can filter these boards by their date and he can specify what the department board he wants.

Input: click on department boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, subject, department name, author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on department boards item in the students main menu.

Pre-condition: single click on department boards item in student's main menu.

Post-condition: table containing department boards

13. College boards.

Function: display college boards.

Description: In this section the student can show all of his college boards and he can filter these boards by their date and he can specify what the college boards he wants.

Input: click on college boards in student's main menu.

Source: student and his main menu web form.

Output: display table with board name, subject, college name, subject, college name author and link button to read the board or to save it locally.

Destination: SQL database server.

Require: valid log in and single click on college boards item in the students main menu.

Pre-condition: single click on college boards item in student's main menu.

Post-condition: table containing college boards.

14. Change password.

Function: change password.

Description: In this function the student has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

Input: click on change password in student's main menu.

Source: student and his main menu web form.

Output: change my password web form that contains a small frame including textboxes to insert old password, new password and then confirm it.

Destination: change my password web form.

Require: valid log in and single click on change password item in the students main menu.

Pre-condition: single click on change password item in student's main menu.

Post-condition: change password web form.

14.1 Update changed password process.

Function: Update change password process.

Description: this sub-function enables the student to confirm the changing process by clicking update button to save new password in database.

Input: click on update password button in change my password web form.

Source: student and change my password web form.

Output: new changed password saved in database.

Destination: SQL database server.

Require: inserting old password correctly.

Pre-condition: filling old password, new password textboxes correctly.

Post-condition: new password saved in database.

15. General voting.

Function: General voting.

Description: From the student main menu web form, student can show the question to be answered and he can show the questionnaire results and he can answer in yes, no or maybe.

Input: select answer by click on some radio button.

Source: student and his main menu web form.

Output: save answer in the database.

Destination: SQL database server.

Require: choose answer.

Pre-condition: entering question for voting and answer it.

Post-condition: save the result in database.

16. New boards.

Function: new boards.

Description: In this function, the student can see all recent boards types related to him. By default these boards since three days ago and the student can insert the number of days ago to check the issued boards within this time period.

Input: number of days and click on generate button.

Source: student and his main menu web form.

Output: issued boards within the specified time period.

Destination: SQL database server.

Require: filling since number of days textbox.

Pre-condition: student menu web form.

Post-condition: student menu web form with intended boards.

18. Documents.

Function: this function enables the student to print some documents.

Description: From here the student can print his academic status with finished hours and marks, and he can print paper that insures that this student is regular students in the university.

Input: single click on the document name link.

Source: Administrator.

Output: document body.

Destination: SQL database server.

Require: choosing document name.

Pre-condition: student menu web form.

Post-condition: document displayed.

19. Insert boards.

Function: Insert boards.

Description: this function Provides administrator the ability to insert boards. In this section the administrator can insert new boards by providing the primary board data.

Input: click on insert board link from administrator main menu and fill Date, Due-to date, Board title, Board number and the author textboxes and the board text.

Source: Administrator and insert new board web form.

Output: save the board and its information in database in the database.

Destination: SQL database server.

Require: choose board body location type new one in board body text.

Pre-condition: fill all textboxes and check board body radio button, then type the board body and click insert button.

Post-condition: saved board in database.

20. Delete boards.

Function: Delete boards.

Description: this function Provides administrator the ability to delete boards. This form specialized to enable the administrator to delete expired boards that exceeds the specified due-to date.

Input click on delete board link from administrator main menu and check the radio button to define the condition that the board will deleted according it.

Source: Administrator and delete boards web form.

Output: deleted board and its information from database.

Destination: SQL database server.

Require: select board type.

Pre-condition: get boards in target board's drop down list and then specify which expired boards to delete.

Post-condition: deleted board from database.

20.1. Get boards.

Function: Get boards.

Description: this function attaches expired boards to be displayed in target boards drop down list.

Input: check on expired boards, board name or board type radio buttons.

Source: Administrator and delete boards web form.

Output: display all expired boards.

Destination: boards to delete drop down list

Require: select board type.

Pre-condition: click on delete board link from administrator main menu.

Post-condition: select boards to delete and click delete button to complete the process.

20.2. Brows boards.

Function: Brows boards.

Description: this function used if the board body is saved on the local server, so administrator can brows the board instead of typing it again.

Input click on brows button to specify path of board location.

Source: Administrator and delete boards web form.

Output: the path of the board filled in path textbox.

Destination: path textbox.

Require: select board type.

Pre-condition: get boards in target board's drop down list and then specify which expired boards to delete.

Post-condition: deleted board from database.

21. Update boards.

Function: Update boards.

Description: Provides the administrator the ability to update boards; this form used to make some editing to existing boards.

Input: click on update board link from administrator menu.

Source: administrator and update boards form.

Output: updated board.

Destination: SQL database server.

Require: administrator should identify what he going to update and specify the board location if it exist.

Pre-condition: select board name from all available boards.

Post-condition: confirm the process by pressing on change button.

22. Insert Marks.

Function: Insert marks.

Description: this function Provides administrator the ability to insert marks of courses that offered at the end of current semester.

Input: click on insert marks link from administrator menu and choose the course number and student number then insert marks.

Source: administrator and insert marks web form.

Output: save courses marks for each student registered in the previous specified course.

Destination: SQL database server.

Require: select course number.

Pre-condition: select course number and student number.

Post-condition: displaying students and insert marks then submit the process.

23. Add new college.

Function: Add new college.

Description: Here administrator can add a brief description for the new college and the college number determined automatically and he can insert the college name and click insert button to save the information in the database.

Input: click Add college link from administrator menu and college name and description but the college number determined automatically.

Source: administrator and add new college web form

Output: save college information in colleges table in the database.

Destination: SQL database server.

Require: college number and name.

Pre-condition: click on add college from administrator menu.

Post-condition: save college information in database by clicking insert button.

24. Insert new major.

Function: Insert new major.

Description: This form enables the administrator to insert new major at any college or department, a brief description can be added and the primary major information can be added too.

Input: major number, name, number of credit hours, outline year and department name.

Source: administrator and insert new major web form

Output: save major information in majors table in the database.

Destination: SQL database server.

Require major number, name, number of credit hours outline year and department name.

Pre-condition: click on insert new major from administrator menu.

Post-condition: save major information in database by clicking insert button.

25. Questionnaire analysis.

Function: Questionnaire analysis.

Description: Administrator can provide students with the questionnaire results and voting analysis that often come from database and calculated automatically, the student can vote one time only.

Input: click questionnaire analysis link from administrator main menu.

Source: administrator and questionnaire analysis web form.

Output: display questionnaire result.

Destination: SQL database server.

Require: None.

Pre-condition: entering question for voting.

Post-condition: upload the question.

26. Control panel.

Function: Control panel.

Description: From this link the administrator gain control over all the portal processes, links and forms, he can activate or deactivate any link or control in the students accounts.

Input: student, instructor or portal settings.

Source: administrator and control panel web form.

Output: activate or deactivate student, instructor settings and other portal links and processes.

Destination: student and instructor main menus and d other portal settings.

Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

26.1. Apply student settings.

Function: Apply student settings.

Description: this function used to apply settings that selected for manage student account to allow or deny him from performing some activities.

Input: select setting.

Source: administrator and control panel web form.

Output: activate or deactivate student settings.

Destination: student main menu.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

26.2. Apply instructor settings.

Function: Apply instructor settings.

Description: this function used to apply settings that selected for manage instructor account to allow or deny him from performing some activities.

Input: select setting.

Source: administrator and control panel web form.

Output: activate or deactivate instructor settings.

Destination: instructor main menu.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

26.3. Apply Portal settings.

Function: Apply Portal settings.

Description: this function used to apply settings that selected for manage portal such as delete expired boards automatically or allow browsing the Portal inside the university only.

Source: administrator and control panel web form.

Output: activate or deactivate Portal settings.

Destination: Portal settings.

Require: Require: choose setting by select checkboxes.

Pre-condition: select settings.

Post-condition: apply selected settings.

27. User accounts.

Function: User accounts.

Description: In this function administrator should select the PAP user type and then he will show all the authenticated users with their ID's and their passwords.

Input: user type.

Source: Administrator and user accounts web form.

Output: Table with usernames and passwords.

Destination: SQL database server.

Require: user type.

Pre-condition: select user type.

Post-condition: display accounts.

28. Change password.

Function: change password.

Description: In this function the administrator has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

Input: click on change password in administrator main menu.

Source: student and his main menu web form.

Output: change my password web form that contains a small frame including textboxes to insert old password, new password and then confirm it.

Destination: Change password web form.

Require: valid log in and single click on change password item in the administrator main menu.

Pre-condition: single click on change password item in administrator's main menu.

Post-condition: change password web form.

29. Insert boards by instructor.

Function: Insert boards.

Description: this function Provides instructor the ability to insert boards. In this section the instructor can insert new boards by providing the primary board data, the board type might be student or course only.

Input: click on insert board link from instructor main menu and fill Date, Due-to date, Board title, Board number and the author textboxes and the board text.

Source: Administrator and insert new board web form.

Output: save the board and its information in the database.

Destination: SQL database server.

Require: choose board body location type or write new one in board body text.

Pre-condition: fill all textboxes and check board body radio button, then type the board body and click insert button.

Post-condition: saved board in database by clicking insert button.

30. Insert Marks by instructor.

Function: Insert marks.

Description: this function Provides instructor the ability to insert marks by himself of courses that offered by him only.

Input: click on insert marks link from instructor menu and choose the course number and student number then insert marks.

Source: instructor and his insert marks web form.

Output: save courses marks for each student registered in the previous specified course.

Destination: SQL database server.

Require: select course number.

Pre-condition: select course number and student number.

Post-condition: displaying students and insert marks then submit the process.

31. Change password.

Function: change password.

Description: In this function instructor has the ability to change his password by inserting the old password and replace it by a new one and confirm it. Then click update button to complete the changing process.

Input: click on change password in instructor's main menu.

Source: instructor and his main menu web form.

Output: change my password web form that contains a small frame including textboxes to insert old password, new password and then confirm it.

Destination: change my password web form.

Require: valid log in and single click on change password item in the instructor main menu.

Pre-condition: single click on change password item in instructor's main menu.

Post-condition: change password web form.

32. My regular students.

Function: My regular students.

Description: In this function instructor has the ability to generate list of his regular student in one of his courses.

Input: course number.

Source: instructor and his main menu web form.

Output: list of regular students.

Destination: SQL database server.

Require: valid log in and single click on my regular students item in the instructor main menu.

Pre-condition: single click on change password item in instructor's main menu.

Post-condition: list of regular students.

33. Log in as instructor.

Function: log in as instructor.

Description: identifying the instructor by using valid username and password.

Input: instructor username and password

Source: instructor and log in form.

Output: instructor main menu.

Destination: SQL database server.

Require: valid log in and valid username and password.

Pre-condition: Home Page and disconnected instructor.

Post-condition: connected instructor and his main menu displayed.

34. Help.

Function: Help.

Description: this link used to conduct the portal visitors how to benefit form services and facilities provided by the portal.

Input: single click on help link on top of home page.

Source: Any PAP visitor.

Output: help content.

Destination: SQL database server.

Require: None.

Pre-condition: single click on help link.

Post-condition: help content.

35. Search engine.

Function: search engine.

Description: This search engine limited to courses, colleges, majors, employees and boards, this search limited on portal database.

Input: complete or part of the subject name.

Source: Any PAP visitor.

Output: table of required search.

Destination: SQL database server.

Require: complete or part of the subject name.

Pre-condition: click portal search.

Post-condition: search content.

36. FAQs.

Function: FAQs.

Description: This link used to serve students with dynamic help that provided by portal specialists that including common Frequently Asked Questions and try to answer them..

Input: single click on the question.

Source: Any PAP visitor and FAQs section of home page.

Output: suggested answer.

Destination: SQL database server.

Require: None.

Pre-condition: click the question.

Post-condition: the answer.

37. General activities.

Function: general activities.

Description: This area contains current general activities take place or that expected to occur.

Input: single click on activity name.

Source: Any PAP visitor and general activities section of home page.

Output: the activity illustration.

Destination: SQL database server.

Require: None.

Pre-condition: click the activity name.

Post-condition: detailed activity explanation.

38. Honor board.

Function: honor board.

Description: This corner displays scrolled marquee with student name links of honored students. To show details of each student you have to click on name link.

Input: single click on the honored student name.

Source: Any PAP visitor.

Output: table of student name, major and accumulative average.

Destination: SQL database server.

Require: None.

Pre-condition: click the student name name.

Post-condition: table of student name, major and accumulative average.

39. Boards marquee.

Function: board's marquee.

Description: this function used to show the latest boards for publics since seven days.

Input: single click on the board name.

Source: Any PAP visitor.

Output: board body.

Destination: SQL database server.

Require: None.









Pre-condition: click the board name.

Post-condition: board body.








2.4 Data Dictionary:

Tables



1. academic_status:

-  ac_no.
-  warning.
-  Dismiss.
-  Delay
-  Std_level.
-  Regular.
-  Graduated.
-  Hours_reg.



2. boards.

-  Brd_no.
-  Brd_date.
-  Due_to.
-  Bodyt.
-  Bodyb.
-  Auther.
-  Subject.



3. brd_col.

-  Col_no.
-  Brd_no.



4. brd_crs.

-  Crs_no.
-  Brd_no.




5. brd_dpt.

-  Dpt_no.
-  Brd_no.





6. brd_std.

-  std_no.
-  brd_no.



7. Colleges.

-  Col_no.
-  Coll_name.
-  Descr.

8. Courses.

-  Crs_no.
-  Crs_name.
-  Crd_hours.
-  Descr.

9. courses_type.

-  Co_no.
-  Co_t.

10. crs_mjr.

-  Crs_no.

📌 Mjr_no.

11. departments.

📌 Dpt_no.

📌 Dpt_name.

📌 Descr.

📌 Col_no.

📌 Cwh_r.

📌 Ac_no.

12. emp_class.

📌 Emp_no.

📌 Crs_no.

📌 Sem_no.

📌 Class_no.

13. employee.

📌 Emp_no.

📌 Emp_name.

📌 Dpt_no.

📌 St_no.

📌 Id_no.

14. fin_status.

📌 Fin_no.

📌 Assistant.

📌 Scholarship.

📌 Others.

📌 Credit.

📌 Debit.

📌 Std_no.

15. honor.

📌 Std_no.

📌 Major.

📌 Std_avg.

📌 Std_pic.

📌 Fname.

📌 Sname.

📌 Tname.

📌 Lname.

16. login_admin.

📌 Username.

📌 Pwd.

17. login_ins.

📌 Emp_no.

📌 Pwd.

18. login_std.

📌 std_no.

📌 pwd.

19. majors.

📌 Mjr_no.

📌 Mjr_name.

📌 Hours.

📌 Descr.

📌 Dpt_no.

📌 Outline.

20. offered_courses.

📌 Crs_no.
📌 Class_no.
📌 Time.
📌 Room_no.
📌 Sem_no.
📌 Ac_year.

21. passed_crs.

📌 Std_no.
📌 Crs_no.
📌 Sem_no.
📌 Ac_year.
📌 Class_no.
📌 Mark.
📌 Co_no.

22. setting.

📌 Set_no.
📌 Set_s.
📌 Set_date.

23. students.

📌 Std_no.
📌 Std_fname.
📌 Std_sname.
📌 Std_tname.
📌 Std_lname.
📌 Birth_y.
📌 Birth_m.
📌 Birth_d.

24. study_system.

📌 St_no.
📌 St_name

25. taw_branch.

📌 Ta_no.
📌 Ta_name.

26. vote.

📌 Vote_no.
📌 Vote_body

27. vote_result.

📌 Vote_no.
📌 V_yes.
📌 V_no.
📌 V_maybe.
📌 Who.

Views

1. academic_s

📌 std_fname.
📌 std_sname.
📌 std_tname.
📌 std_lname.
📌 Birth_p
📌 Ta_name.
📌 Tawjehy_avg.

Warning.

2. balance:

assistant.
Scholarship.
Others.
Credit.
Debit.
Stdno.
Semno.
Ac_year.

3. basic_info:

std_no.
std_fname.
std_sname.
std_tname.
std_lname.
Tel_no.
Address.

4. board_view:

date.
author.
title.
Board_no.

5. calcavg:

std_no.
mark.
Crs_no.

Co_no.

Crd_hours.

Std_fname.

Std_sname.

Std_tname.

6. col_brd:

board_date.
Author.
Title.
College_no.

7. col_info:

college.
College_no.

8. colleges_boards:

author.
Subject.
Col_no.
Brd_no.
Coll_name.
Brd_date.
Bodyt.
Bodyb.

9. courses_boards:

author.
Subject.
Crs_no.
Brd_no.

📁 Brd_date.

📁 Crs_name.

📁 Bodyt.

📁 Bodyb.

10. crs_boards:

📁 Crs_name.

📁 Brd_date.

📁 Auther.

📁 Subject.

📁 Std_no.

📁 Crs_no.

📁 Brd_no.

11. crs_info:

📁 Course_no.

📁 Course

📁 Credit_hours.

12. crs_reg:

📁 Crs_name.

📁 Std_no.

📁 Crs_no.

13. current_schedule:

📁 Time.

📁 Room_no.

📁 Crs_name.

📁 Crs_no.

📁 Sem_no.

📁 Std_no.

📁 Class_no.

📁 Ac_year.

14. departments_boards:

📁 Author.

📁 Subject.

📁 Dpt_no.

📁 Dpt_name.

📁 Brd_no.

📁 Brd_date.

📁 Bodyt.

📁 Bodyb.

15. dpt_brd:

📁 Board_date.

📁 Author.

📁 Title.

📁 Dpt_no.

16. emp_account:

📁 Emp_no.

📁 Emp_name.

📁 Dpt_name.

📁 St_name.

📁 Pwd.

📁 Id_no.

17. emp_info:

📁 Employees.

📁 Sientific degree.

📁 Department.

📌 Employees_no.

18. emp_view:

📌 Emp_no.

📌 Crs_no.

📌 Class_no.

📌 Std_no.

19. login_view:

📌 std_sname.

📌 Std_sname.

📌 Std_tname.

📌 Std_lname.

📌 Pwd.

📌 Std_no.

📌 Dpt_no.

📌 Col_no.

20. loginins:

📌 emp_no.

📌 pwd.

📌 Emp_name.

21. main_boards:

📌 due_to.

📌 Subject.

📌 Brd_no.

📌 Brd_date.

📌 Bodyt.

📌 Bodyb.

📌 Author.

22. mark_hours:

📌 crd_hours.

📌 Std_no.

📌 Mark.

23. mjr_info:

📌 mahor.

📌 Credit_hours.

📌 Outline.

📌 Department.

📌 Major_no.

24. my_crsboards:

📌 Crs_no.

📌 Brd_no.

📌 Bodyt.

📌 Bodyb.

📌 Author.

📌 Subject.

📌 Brd_date.

📌 Std_no.

25. passed:

📌 Crs_no.

📌 Std_no.

📌 Co_t.

📌 Co_no.

📌 Crs_name.

📌 Crd_hours.

📌 Mark.

📁 Sem_no.

📁 Std_no.

26. profile:

📁 std_fname.

📁 Std_sname.

📁 Std_tname.

📁 Std_lname.

📁 Address.

📁 Telno.

📁 Mjr_name.

📁 Outline.

30. stdpaper:

📁 Std_fname.

📁 Std_sname.

📁 Std_tname.

📁 Std_lname.

📁 St_name.

📁 Mjr_name.

📁 Std_no.

27. sem_hours:

📁 Std_no.

📁 Hours_reg.

📁 Hours_p.

31. student_boards:

📁 Author.

📁 Subject.

📁 Std_no.

📁 Std_sname.

📁 Std_fname.

📁 Std_tname.

📁 Std_lname.

📁 Brd_no.

28. semester_marks:

📁 Crs_name.

📁 Crd_hours.

📁 Crs_no.

📁 Std_no.

📁 Mark.

📁 Ac_year.

📁 Sem_no.

📁 Co_no.

29. std_brd:

📁 Board_date.

📁 Author.

📁 Title.



5.1 Introduction

Before we deliver the final system copy to the customer we must test the system to ensure that the system meet its specification and is worked properly as we expected.

Testing process is divided into number of successive stages or levels that will be implemented to ensure form system validation.

This phase from software development is a most important phase since it consumes a lot of time, and require a great effort to get system to be reliable, this doesn't mean that the system must be completely free of errors and faults, but it means that the system should be good enough for its intended use.

The testing process include five successive levels as below:

- System units testing.
- Module testing.
- Subsystems testing.
- Integration testing.
- System and acceptance testing.

Testing process will take place in time space that was assigned for the testing process, table (5.1) illustrate testing process schedule:



5.2 System unit testing

at this stage of testing process we separate or divided the system into subsystems and components that will be tested separately to ensure that they operate correctly and meet its specification.

In our system the administrator operations is the most operations that affect the system and database , so we will test login, delete boards, and change password process, and for the students operations we will test change password, and voting process.

We will insert valid inputs one time, and invalid another time, and we will see the system responding result.

This table describes the unit testing process for administrator login:

Unit testing process case	Input data testing		Expected result	Actual result
	Username	Password		
Valid login username and password	111	456	Administrator page loaded	Administrator page loaded
Wrong username or password expression	Adk14	\$hy1/dj	Invalid inputs format	Invalid inputs format
Invalid login username or password	111	654	Error alert appear, return to login page	Error alert appear, return to login page

Table (5.2) Administrator login process testing



Now we will testing change password process, so the following table illustrate it:

Unit testing process case	Input data testing			Expected result	Actual result
	Username	New Password	Confirm it		
Valid login username and password and confirm it correctly	111	678952	678952	Password updated successfully	Password updated successfully
Wrong username or password expression	Gfr342	Efr&4\$	Efr&4\$	Invalid inputs format	Invalid inputs format
Invalid login username	Frd324	678952	678952	Error alert appear, return to change pwd page	Error alert appear, return to change pwd page
Wrong in password confirmation	111	678952	678925	Error alert, "Confirm password correctly"	Error alert, "Confirm password correctly"

Table (5.3) Change password process testing



Now we will testing delete boards process, the following table show the process:

Unit testing process case	Input data testing		Expected result	Actual result
	Select type of deleted board	Select board you want to delete		
Select deleted board type, then select the board from dropdown list to delete it, then click delete button	Expired boards	Board title	The board was deleted successfully	The board was deleted successfully
Doesn't select board type to be deleted, and click on delete button	Not selected	Empty list	Error alert," Please select the board type first"	Error alert," Please select the board type first"
Select board type, doesn't select the board to delete, and click on delete button	Board Name	Not selected	Error alert ," Please select the board you want to delete from list"	Error alert ," Please select the board you want to delete from list"

Table (5.4) Delete boards process testing



The table below show the testing of change password for students process:

Unit testing process case	Input data testing			Expected result	Actual result
	Username	New Password	Confirm it		
Valid login username and password and confirm it correctly	Session("b")	111222	111222	Password updated successfully	Password updated successfully
Wrong password expression	Session("b")	Efr&4\$	Efr&4\$	Invalid inputs format	Invalid inputs format
Invalid login password	Session("b")	\$j\7@	\$j\7@	Error alert appear, return to change pwd page	Error alert appear, return to change pwd page
Wrong in password confirmation	Session("b")	111222	222111	Error alert, "Confirm password correctly"	Error alert, "Confirm password correctly"

Table (5.5) Change student password process testing.



The following table illustrate the student voting process testing:

Unit testing process case	Input data testing			Expected result	Actual result
	Yes	No	Maybe		
Doesn't select any choice, then click submit button	Not selected	Not selected	Not selected	Error alert, "select at least one choice from list"	Error alert, "select at least one choice from list"
Select yes, no, or maybe " first time"	✓	Or ✓	Or ✓	Thanks, process done successfully.	Thanks, process done successfully.
Select yes, no, or maybe at another time for the same question	✓	Or ✓	Or ✓	Error alert , " Sorry, you were voting"	Error alert , " Sorry, you were voting"

Table (5.6) Answer question (voting) process testing.

5.3 System module testing

At this level we test the modules which encapsulated related components, so it can be tested without other system modules. Procedures, functions, and system interface are tested.



5.4 *Sub-system testing*

Our system include mainly three subsystems or clusters there are:

1. *Administrator subsystem.*

We tested all functions and procedures that operate through administrator pages to ensure that they are work correctly and meet its specification.

And we tested the data flow and transfer between procedures and functions, and interfaces and how data and information layout on pages and how they are appearance.

And we tested security side, to ensure that the system is secure and is safe from hacking.

2. *Instructor subsystem.*

We tested instructor subsystem as we do at administrator subsystem, we tested all functions and procedures to ensure that they work correctly and meet its specification and its behave as we expected.

And we tested the interface , and how the information are layout and appearance on screen, in addition we tested security to prevent any attempt to harmful or destroy the system, and we tested data transfer between functions and procedures to ensure that all data are correctly flowed and used.

3. *Student subsystem.*

As the previous subsystems we tested all functions and procedures, and in every page we tested the data flow between the user and database to ensure that they are transfer correctly and in correct way.



And we tested user interface, and how information layout on pages, and we take in our consideration the interface colors that affect user, so we tested it and we use suitable color for user, and we tested security to ensure that the system is secure and safely.

5.5 Integration testing

After we tested the subsystems and then we tested modules as encapsulated components, at this level we integrated all modules together and tested them as a whole system to ensure that the system works properly and meet its specifications.

The testing of the system integration indicates that the system works properly and meet its specification.

5.6 System and acceptance testing

1. System testing.

To make up the system we integrate administrator , instructors, and students subsystems, so after this integration some errors or faults will appear, since miss interaction between subsystems and problems in the interface between them, this process is concerned with finding this errors which are unanticipated.

So we validate the system to ensure from its functional and non-functional requirements .

2. Acceptance testing

This process test if the system is consistent with its environment or not, so we use this process to ensure from consistency between system and environment.



And we test at this process tow issues:

- ✓ Functional requirements.

We test the system if he meet the functional requirements, such as allow student to read data from database, and allow administrator to read , update, delete data from database.

- ✓ Non-functional requirements.

If the system meet the non-functional requirements such as security, reliability, flexibility.

5.7 Snapshots

in this section we will select some snapshots from the web site to display they here to show how will the real system behave under certain situations and conditions.

5.7.1 Some of the administrator pages.

This snapshot for insert boards page display the system handle invalid input formats:

4.3 Input Output design :

After specification of requirements and describing functional system, we must declare these function throw. The designing forms shows who function works and shows who user can interact with it . and these steps of designing I/O consider as a first step for implementing the software and developing it .

In this chapter we design the I/O user interface design , these are student I/O design and Administrator I/O design.

1- Student interface design :

a. Student Input design:

Student can't entered to portal without ID and Password , and he constrained with your page , and not make any think in this portal only authenticated with sum works.

➤ Login student :

The login student interface is an input user interface . and from it the user can entered to the university portal to saw your information .

These interface have tow textbox and hyperlink :

- textbox for user Id in the university
- textbox for student Password
- hyperlink (forgot my password) these important if the student forgot password and he want to retrieve it.

Login

Please Insert Your ID And Password

Student Name :

Password :

Login

[Forgot My Password?](#)

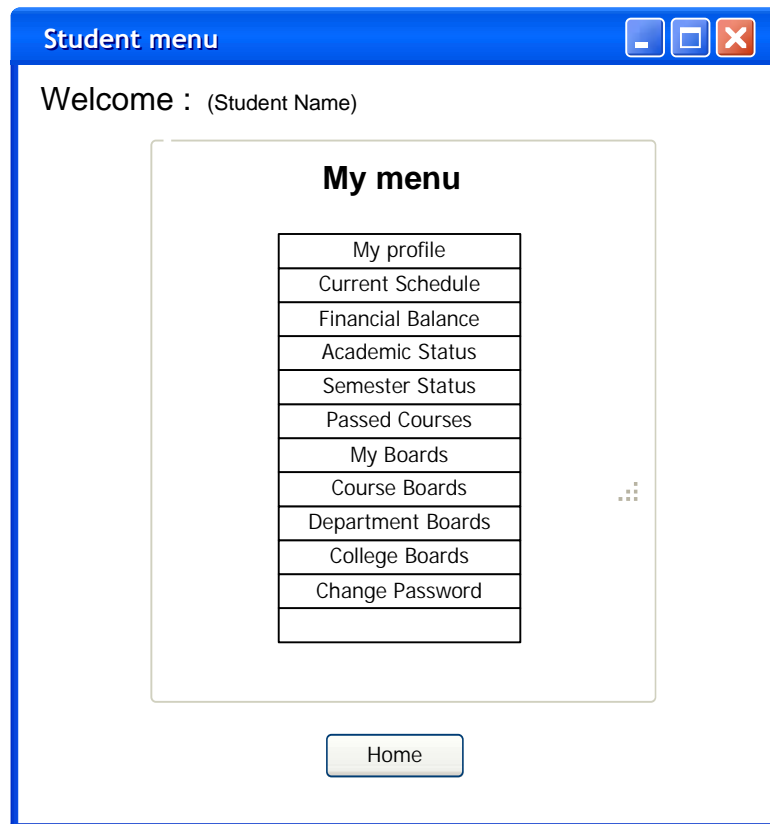
➤ **Student Menu :**

this menu gives student all task that can make it .

this menu have many links to many page and these links are :

- My profile hyperlink : this link go with student to your profile to sure for yourself information.
- Current Schedule link: this show current courses for current semester and instructor for it.
- Financial balance : this appeared finance information for student and provide with credit, dept balance .
- Academic status : this provide with academic information in registered unite .
- Semester Status : this show status to current semester .
- Passed Courses : shows the passed course in the university and the hours not passed it.
- My Boards : show boards for you.

- Course Board : show board for registered course in current semester.
- Department Board : show board for your department.
- College board : show board for the college .
- Change Password : this give student flexibility to change your password



➤ **Chang My Password:**

The student can change password from this link by inserting old password ,new password ,and confirm new password.

This have three input text box and three command button to:

- old password textbox
- new password textbox
- confirm new password textbox
- back command button
- home page command button
- update password command button.



The image shows a screenshot of a web browser window titled "Change Password". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a white background with a blue border. At the top center, the text "Change Password" is displayed in bold. Below this, there is a light gray rounded rectangle containing three input fields. The first field is labeled "Old Password", the second "New Password", and the third "Confirm it". Each label is positioned to the left of its corresponding input box. Below the input fields, there are three buttons: "Update", "Back", and "Home", arranged horizontally. The "Update" button is on the left, "Back" is in the middle, and "Home" is on the right. All buttons have a light gray background and a thin border.

b. Student Output design :

This show the result of inputs in forms and may be show some of input in this forms.

These output design:

➤ My profile :

The output in this form are :

- student name (first name, second, third, and last name) these show in the form and can't change of it.
- Address : the address of student
- Telephone no: the telephone of student.
- Academic : academic no, science degree, Major, Outline.

The screenshot shows a web browser window titled "My Profile". The main content area is titled "My Profile [stdncj]". It contains three sections, each with a blue header and a light yellow border:

- My Name:** This section contains four input fields. The first two are labeled "First Name:" and "Family Name:". There are two unlabeled input fields between them.
- My Address:** This section contains two input fields. The first is labeled "Address:" and the second is labeled "Telephone no:".
- My Academic Info.:** This section contains four input fields. The first is labeled "Academic no:", the second "Science Degree:", the third "Major:", and the fourth "Outline:".

At the bottom of the form, there are two buttons: "Back" and "Home".

➤ **Financial Balance :**

The output in this form are :

- Assistance : show the assistance for student in university
- Scholarship : show the scholarship for student in university
- Other : show other aid for student
- Credit Balance
- Debit balance
- Semester balance
- Total Balance

The screenshot shows a web browser window titled "Financial balance" with standard window controls. The main content area contains a form with the following structure:

- Financial balance** [Std.No]
- Financial Aids** section containing:
 - assistance** [input field]
 - Scholarship** [input field]
 - Other** [input field]
- Balance** section containing:
 - Credit** [input field]
 - Debit** [input field]
- Balance** section containing:
 - Semester Balance** [input field]
 - Total Balance** [input field]

At the bottom of the form, there are two buttons: "Back" and "Home".

➤ **Academic Status :**

The output in this form are :

- major degree : show the degree of major ex: diploma, bachelor, master .
- tawjihi branch : science, ...
- tawjihi average : shows average in tawjihi .
- hours of passed courses .
- hours of remains courses .
- department name and college name : department and college name for student .
- average accumulate : shows average accumulative in passed courses .
- average major : the average in required major courses .
- academic alert : shows if student have alert or no
- Dismiss : show if student are dismiss in university or no
- Delay : is student dealy in major or no
- Level : show the level of study in university.
- Regular : shows if student is regular or no
- Graduated : show if student is graduated or no in this semester .

Academic status

Academic Status [stdno]

First Name **Birth Place**

Major Degree : **Major Name :**

Department Name : **College Name :**

Tawjithi Branch : **Tawjithi Average :**

Community Work Hours

Passed : **Remains :**

Average

Average Accumulate :

Average Major :

Academic Alert : **Dismiss :** **Delay :** **Level :** **Regular :** **Graduated :**

➤ **Semester Status :**

The output in this form are :

- course no : show the current courses in current semester
- course name:
- semester : current semester
- year
- Mark : mark for each course in current semester

- Honored : if the student take honored in study
- Alert : if student take alert in this semester.
- semester average : shows the average in this course
- level : level of study

Semester Status

Semester Status [stdno]

Semester : _____ Year : _____

Course No	Course Name	Mark

Credit Hours : _____

Registered : Passed :

Honored : Semester Avg:

➤ **Passed Courses :**

The output in this form are :

- semester no : show current semester
- year : show current year
- course no
- course name
- course type : the type of courses

- credit hours
- mark
- accumulative average
- passed courses
- number of courses remains

Passed Courses
_ □ ×

Finished Hours [Std No]

Semester :
Year :

Filter Criteria

Year

Semester

Course Type

Pick Value

Year Semester

2004 1

Course Type

Filter Criteria

Acc. Avg. Pass CH.No. CH. Remains

Course No	Course Name	Cours Type	Credit H.	Mark	semester	Year

➤ **My Boards:**

The output in this form are :

- Board date : show the date of board
- Subject : the subject for board
- Author : whose send board
- Board Body : the body of the board

My Board [stdno]

Board Since Days

Semester : _____ Year : _____

Date	Author	Subject	Body

➤ Course Boards

This interface consider input and output interface because have input box called :

- Course Name : is a compo box used to filtered boards as current courses

The output in this form are :

- Board date : show the date of board
- Subject : the subject for board
- Author : whose send board
- Board Body : the body of the board

Course Boards [stdno]

Board Since Days

Semester : Year :

Date	Author	Subject	Course No	Course Name	Body

➤ **Department Boards :**

The output in this form are :

- Department Name : is a compo box used to filtered boards as student department
- Board date : show the date of board
- Subject : the subject for board
- Author : whose send board
- Board Body : the body of the board

Department Board

Department Boards [stdno]

Board Since Days

Semester : Year :

Date	Author	Subject	Department Name	Body

➤ **College Boards :**

The output in this form are :

- College Name : is a compo box used to filtered boards as student College
- Board date : show the date of board
- Subject : the subject for board
- Author : whose send board
- Board Body : the body of the board

College Board

College Boards [stdno]

Board Since Days

Semester : Year :

Date	Author	Subject	College Name	Body

2- Administrator interface design :

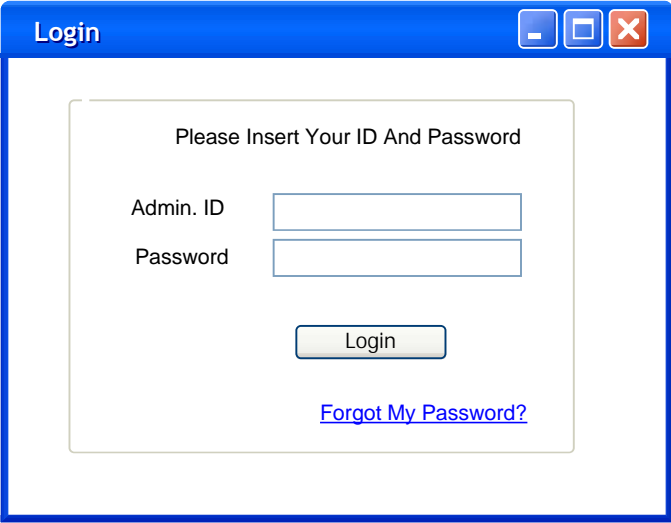
a. Administrator Input design:

Administrator can't entered to portal without ID and Password , and he have all permission on portal .

➤ Login administrator :

These interface have tow textbox :

- textbox for Administrator Id
- textbox for Administrator Password



The image shows a screenshot of a web application's login interface. The window has a blue title bar with the text "Login" and standard window control buttons (minimize, maximize, close). The main content area is white and contains a light gray rounded rectangle with the following elements:

- Text: "Please Insert Your ID And Password"
- Label: "Admin. ID" followed by a text input field.
- Label: "Password" followed by a text input field.
- Button: "Login" (a rectangular button with rounded corners).
- Link: "[Forgot My Password?](#)" (a blue underlined text link).

➤ **Insert boards :**

The inputs in this form are :

- Date : the date of the inserted board
- Due to : specify the date that the board will finish or deleted.
- Title : the title of the board
- Author
- Board body
- Board type

The screenshot shows a web application window titled "Insert Boards" with a blue header. The main content area is titled "Insert New Boards Form". At the top, there are two labels: "Semester :" and "Year :". Below these is a section titled "Board Data" containing five input fields: "Date:", "Due To:", "Title:", "Board no:", and "Author:". Below the "Board Data" section is a "Board body" section. It contains two radio buttons: "Exit" and "New". Below these is a "Select board type" section with five radio buttons: "Bulletin board", "College board", "Department board", "Course board", and "Student board". To the right of the "Board body" section is a "Board text" section. It contains a large text area with a vertical scrollbar, a horizontal scrollbar, and a "Browse" button. At the bottom of the window, there are two buttons: "Back" and "Home".

➤ **Delete boards :**

Delete boards

Delete boards Form

Delete boards According

<input type="radio"/> Expired boards	Get Boards	Target Board	>	Boards to delete
<input type="radio"/> Board name		<input type="text"/>	<	<input type="text"/>
<input type="radio"/> Board type		<input type="text"/>	>>	<input type="text"/>

Empty

Delete

Home Back

➤ **Update boards :**

Update boards

Update boards Form

Available Boards Titles

Update

Board & deadline

Deadline only

Board Type

Exit

New

Board Deadline

Change

Browse

Update

Home

Back

➤ **Insert college :**

Insert College

Insert college form

Description type

Exit

New one

College no .

College Name .

College description

Browse

Insert

Home

Back

➤ **Insert Major :**

Insert major

Insert major form

Description type

Exit
 New one

Major no.

Major Name.

Credit H.

Outline year

Department name

College description

Table of contents

	:	
Dedication		ii
Acknowledgment		iii
Abstract		iv
Project problem		v
Table of contents		vi
List of tables		viii
List of figures		ix

Chapter One: System Specifications

1.1 Introduction	2
1.2 System objectives	2
1.3 Functional Description	5
1.4 Non-Functional description	9
1.5 Project constraints	9
1.6 feasibility study	10
1.6.1 Alternatives	10
1.6.2 Cost benefit analysis	14
1.6.3 Evaluation of risk	15
1.7 Economic study	16
1.8 Time scheduling	19

Chapter two: Software Requirements Specifications.

2.1 Introduction	21
2.2 Requirements specifications	21
2.3 Functional description	32
2.4 Validation criteria	79
2.5 Information description	80
2.5.1 Dataflow diagram	80
2.5.2 Data dictionary	81
2.5.3 System interface description	91
2.5.4 Database requirement	93
2.5.5 System context model	107

Chapter Three: System Design.

3.1 Introduction	109
3.2 Functional design	109
3.3 I/O design	177
3.4 Database design	205
3.4.1 Database model.....	205
3.4.2 database table.....	206
3.5 Test plane	217
3.6 Programming language and coding	218

Chapter four: System Implementation.

4.1 Introduction	220
4.2 Establishment of development environment	221
4.2.1 Hardware environment	221
4.2.2 Software environment	221
4.3 Database implementation	226
4.4 Supporting Software	229
4.5 Operating the system	229
4.6 Coding	232
4.7 Development process implementation	233

Chapter Five: System Testing.

5.1 Introduction	235
5.2 System unit testing	236
5.3 System module testing	241
5.4 Subsystem testing	241
5.5 Integration testing	242
5.6 Snapshots	243
5.6.1 Administrator pages	243
5.6.2 Student pages	254

Chapter six: System Maintenance.

6.1 Introduction	262
6.2 Maintenance plan	263
6.3 Migration	266
6.4 Internet Information Services maintenance	267
6.5 SQL Server 2000 maintenance	268
6.6 The .NET Framework maintenance	270

Chapter Seven: Conclusion and Recommendations.

7.1 Conclusions and Recommendations	272
7.1.1 Conclusions	272
7.1.2 Skills	272
7.2 Recommendation and future work	273
7.2.1 Recommendations	273
7.2.2 Future work	273
7.3 References	274

List of Tables

Table (1.1) Development hardware cost.....	16
Table (1.2) Development hardware cost	16
Table (1.3) Cost of human recourses	17
Table (1.4) Hardware cost for implementation	17
Table (1.5) Software cost implementation	17
Table (1.6) Human recourses cost for implementation	18
Table (1.7) The overall development cost summery	18
Table (1.8) The overall implementation cost summery	18
Table (1.9) Time schedule in weeks	19
Table (2.1) Stored procedures	88
Table (2.2) Sessions	89
Table (2.3) Applications	90
Table (3.1) Student table	207
Table (3.2) Academic status	207
Table (3.3) Boards	208
Table (3.4) College boards	208
Table (3.5) Department boards	208
Table (3.6) Course boards	208
Table (3.7) Student boards	209
Table (3.8) Colleges	209
Table (3.9) Courses	209
Table (3.10) Course type	209
Table (3.11) Passed courses	210
Table (3.12) Offered courses	210
Table (3.13) Major courses	211
Table (3.14) Department	211
Table (3.15) Employee class.....	211
Table (3.16) Employee	212
Table (3.17) Financial status	213
Table (3.17) Administrator login	213
Table (3.18) Student login	213
Table (3.19) instructor login	214
Table (3.20) Majors	214
Table (3.21) Projects	214
Table (3.22) Settings	215
Table (3.23) Student style	215
Table (3.24) Study system	216
Table (3.25) Tawjihi branch	216
Table (3.26) Vote	216
Table (3.27) Vote result	216
Table (5.1) Testing schedule	235
Table (5.2) Administrator login process testing	236
Table (5.3) Change password process testing	237
Table (5.4) Delete board process testing	238
Table (5.5) Change Student Password process testing	239
Table (5.6) Answer question Voting process testing	240

List of Figures

Figure (1.1)	Gantt Chart for time schedule	19
Figure (2.1)	Request home page	32
Figure (2.2)	Student login	33
Figure (2.3)	Administrator login	34
Figure (2.4)	My profile	35
Figure (2.5)	Upload student picture	36
Figure (2.6)	Current schedule for student	37
Figure (2.7)	Student financial status	38
Figure (2.8)	Student academic status	39
Figure (2.9)	Semester status	40
Figure (2.10)	Passed courses	41
Figure (2.11)	Filtering criteria	42
Figure (2.12)	Students boards	43
Figure (2.13)	Course boards	44
Figure (2.14)	Department boards.....	45
Figure (2.15)	College boards	46
Figure (2.16)	Change password	47
Figure (2.17)	Update change password	48
Figure (2.18)	Questionnaire	49
Figure (2.19)	New boards	50
Figure (2.20)	Social guide center	51
Figure (2.21)	Student document	52
Figure (2.22)	Insert boards	53
Figure (2.23)	Delete boards	54
Figure (2.24)	Get boards	55
Figure (2.25)	Browse boards	56
Figure (2.26)	Update boards	57
Figure (2.27)	Insert marks	58
Figure (2.28)	Add new college	59
Figure (2.29)	Insert new major	60
Figure (2.30)	Questionnaire analysis	61
Figure (2.31)	Control panel	62
Figure (2.32)	Applying student setting	63
Figure (2.33)	Applying instructor setting	64
Figure (2.34)	Applying portal setting	65
Figure (2.35)	Change password	66
Figure (2.36)	Insert board by instructor	67
Figure (2.37)	Insert mark by instructor	68
Figure (2.38)	My regular students	69
Figure (2.39)	Login as instructor	70
Figure (2.40)	Help	71
Figure (2.41)	Search engine	72
Figure (2.42)	FAQs	73
Figure (2.43)	General activities	74
Figure (2.44)	Honor board	75
Figure (2.45)	Board marquee	76
Figure (2.46)	My style	77
Figure (2.47)	Graduate projects	78

Figure (2.48) Data flow diagram level : 0	82
Figure (2.49) System context model	107
Figure (3.1) Student login operation.....	111
Figure (3.2) Forget my password operation	114
Figure (3.3) Student profile operation	116
Figure (3.4) Student current schedule operation	118
Figure (3.5) Student financial balance operation	120
Figure (3.6) Student academic status operation	122
Figure (3.7) Semester status operation	125
Figure (3.8) Passed courses operation	128
Figure (3.9) My board operation	130
Figure (3.10) Course boards operation	132
Figure (3.11) Department board operation	134
Figure (3.12) College board operation	136
Figure (3.13) Student voting operation	138
Figure (3.14) Change password operation	140
Figure (3.15) Student logout operation	142
Figure (3.16) My style operation	144
Figure (3.17) Login administrator operation	146
Figure (3.18) Insert board operation	149
Figure (3.19) Delete board operation	151
Figure (3.20) Update board operation	153
Figure (3.21) Insert mark operation	155
Figure (3.22) Insert new college operation	158
Figure (3.23) Insert major operation	160
Figure (3.24) Change administrator password operation.....	162
Figure (3.25) Administrator logout operation	164
Figure (3.26) Instructor insert board operation	166
Figure (3.27) Insert mark operation	168
Figure (3.28) Honored student operation	170
Figure (3.29) Public search operation	172
Figure (3.30) Marquee boards operation	174
Figure (3.31) Graduate project operation	176
Figure (3.32) Student login screen	177
Figure (3.33) Student menu screen	178
Figure (3.34) Change password for student screen	179
Figure (3.35) Student profile screen	180
Figure (3.36) Student current schedule screen	181
Figure (3.37) Student financial balance	182
Figure (3.38) Student academic status	183
Figure (3.39) Student semester status	184
Figure (3.40) Student passed courses	185
Figure (3.41) Student boards	186
Figure (3.42) Courses boards	187
Figure (3.43) Department boards	188
Figure (3.44) Colleges boards	189
Figure (3.45) Administrator login screen	190
Figure (3.46) Insert new boards	191
Figure (3.47) Delete boards	192
Figure (3.48) Update boards	193

Figure (3.49) Insert marks	194
Figure (3.50) Insert new college	195
Figure (3.51) Insert new major	196
Figure (3.52) Questionnaire analysis	197
Figure (3.53) Control panel	198
Figure (3.54) User accounts	199
Figure (3.55) Change password	200
Figure (3.56) Insert new board by instructor	201
Figure (3.57) Insert marks by instructor	202
Figure (3.58) Instructor regular students	203
Figure (3.59) Search engine	204
Figure (3.60) Database model	205
Figure (4.1) ASP.NET new application window	222
Figure (4.2) Enterprise manager window	224
Figure (4.3) Adding ASP.NET account on SQL Server login group	226
Figure (4.4) Creating integrated widows security data link	227
Figure (4.5) Select Microsoft Visual Studio.NET 2003	230
Figure (4.6) Select open project	231
Figure (4.7) Select PAP project	231
Figure (4.8) Development process (create new project)	233
Figure (5.7) Invalid input format testing snapshot for login as administrator	243
Figure (5.8) Valid input format testing snapshot for login as administrator	244
Figure (5.9) Administrator home page	245
Figure (5.10) Invalid input format testing snapshot for insert new board	246
Figure (5.11) Leave title field empty and click upload new board	247
Figure (5.12) Insert all fields and valid inputs foe insert new board	248
Figure (5.12) New course board in marquee and new boards area	249
Figure (5.13) Invalid input testing snapshot for insert mark	250
Figure (5.14) Snapshot after inserting new mark	251
Figure (5.15) Delete expired board from database	252
Figure (5.16) Snapshot after deleting the board successfully	253
Figure (5.17) Invalid data format snapshot for change password	254
Figure (5.18) Invalid confirm new password snapshot	255
Figure (5.19) Voting for the first time from student snapshot	256
Figure (5.20) Voting another time for the same student snapshot	257
Figure (5.21) Current schedule page for student	258
Figure (5.22) Financial balance page	259
Figure (5.23) Finished hours and marks	260
Figure (6.1) Database backup	263
Figure (6.2) Backup options	264
Figure (6.3) IIS window	267
Figure (6.4) SQL Server database	286
Figure (6.5) SQL Server 2000 security section	269

References:

1. Sommerville, Ian, *Software Engineering*, 7th edition, Addison-Wesley, 2001.
2. Microsoft SQL server 2003 Help Topics.
3. Microsoft .NET Framework SDK v1.1.
4. www.asp.net/forums.