Palestine polytechnic university

College of administrative sciences and informatics

Department of information technology

Student Portal using SMS

SPSMS

Prepared by

Hana Mujahed **Ibtesam Iqdimat**

Shaheera AL-Aamayreh

Suna Abu Hamdya

Supervised by

Dr.Mohammed Aldasht

4/2/2008

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

Jan, 2008

Dedication

To our parents.....

To our family.....

To our Supervisor Dr.Mohammed Aldasht.....

To our friends.....

And to all our teachers and lecturers.....

Ibtesam

Suna

Shaheera

Hana

Acknowledgment

Many thanks to our supervisor Dr.Mohammed Aldasht.

Many thanks for the head of the Information Technology Department in our university

Mr. Akram Ihshayesh

Much thanks for lecturers Muhammad Amro and Hamde Tahbob.

A lot of thanks for our friends who help us .

Thanks for every one who help us to accomplish this work.

Abstract

The main aim of our project is to exploit a widely used technology to let students to get their university information from any where and any time they want using SMS, and introduce more efficient and effective academic life for ppu students by convert the traditional way into web based work and SMS to save time and effort of student and make response to their requests information more easy.

In addition to meeting student transactions, our system support registered administrator functions to make the process consistent; the administrator can edit the student information, and can provide them with their information such as financial status, marks, academic status.

We conclude that this system should focus on the student requirements for his information more easily.

Project problem

There are many problems face students when they want to get marks academic status such as PC or internet connection to them when they want to get there information, and there are many problems with using PC's and internet like speed of computer, network traffic.

.

So we suggest building this system to give solution for these problems.

Table of contents

List of figures

System Specification

Chapter one

- *Introduction*
- *Application Objectives*
- *Functional Descriptions.*
- *Non- Functional Descriptions.*
- *System Constraints.*
- *Feasibility Study*
- *Resources and Cost.*
- *Time Scheduling.*

1.1 Introduction :

In this chapter we will explore and explain system objectives, functional and non functional description, constraints, and feasibility study that include risks that face the project, solution of the risks, project alternatives, project resources, cost and time scheduling.

1.2 System Objectives:

This project aims to achieve the following objectives:

- Provide easy access for students to get their marks, grades, academic status and financial status through SMS and web site.
- Provide way for Students to get their required information any where and any time using the mobile.
- Save time and efforts.
- Students do not want using PC's or internet connection to get his information.

1.3 Functional Descriptions:

Student functional description:

1. Provide authorized access for each student by his username and password.
2. Provide academic status report for each student that include (accumulate average, grade, major average and warning).
3. Provide financial status report for each student that includes (debit, credit, balance, and assistance).
4. Provide marks report for each student that include (courses name and it's marks).
5. Allow the student to modify his password by sending SMS or by using the student page.
6. Provide report about registered courses that include (courses name and credit hour).

Administrator functional description:

1. Provide authorized access for administrator by his username and password.
2. Add new student information: the administrator can insert new student information.
3. Delete information: the administrator can delete student information.
4. Update information: the administrator can update student information.
5. Student account: allow administrator to display all authenticated students.
6. Change password: allow administrator to change his password by entering new password and confirm it.

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:

a. product requirements :

1. Maintainability and scalability which enforce the system to coexist with any new changes.
2. Security: The system will have the ability to prevent any student from access information for other student through authentication. Also the portal has ability to prevent illegal operations by using validation controls.
3. Understandability that make it easy for students to use and deal with, such as student use abbreviations for queries.
4. Reliability: the processes are designed to prevent the error occurrence before causing large problem.
5. Must operate in windows XP environment.
6. The student portal using SMS system developed for the purpose of supporting integration between existing and future system.

b. Process requirements :

The student portal using SMS project and its document must be delivered by ... Jan 2008.

1.5 System Constraints :

1. The time is limited by the end of the course (4 month).
2. Limited cost.
3. Building the system with simple interface.

1.6 Feasibility Study:

1.6.1 Project Alternatives:

a. Using PC and internet connection to get information.

This way has many advantages and disadvantages:

Advantages:

1. Provide authentication information by allowing the student log in using his own username and password..
2. Provide online interaction with the supervisor.
3. Support security on the database by authorizing student to manipulate his profile by adds, delete and update data and changing password.

Disadvantages:

Inflexibility to force the students to be in a specific place and time to get services.

b. Process requirements :

The student portal using SMS project and its document must be delivered by ... Jan 2008.

1.5 System Constraints :

1. The time is limited by the end of the course (4 month).
2. Limited cost.
3. Building the system with simple interface.

1.6 Feasibility Study:

1.6.1 Project Alternatives:

a. Using PC and internet connection to get information.

This way has many advantages and disadvantages:

Advantages:

1. Provide authentication information by allowing the student log in using his own username and password..
2. Provide online interaction with the supervisor.
3. Support security on the database by authorizing student to manipulate his profile by adds, delete and update data and changing password.

Disadvantages:

Inflexibility to force the students to be in a specific place and time to get services.

b. Using SMS:

In this way students can get there information through mobile by sending SMS to other mobile phone then to PC , but also this way have some advantages and disadvantages :

Advantages:

1. Save time and efforts.
2. Low cost of SMS for students.
3. Availability of mobile.
4. Can be Sent and Read at Any Time.
5. Easy to use.
6. Provide mobility.
7. Provide Security and privacy.

Disadvantages:

1. No interaction between student and the supervisor.
2. Mobile limitation.

1.6.2 Cost-Benefits Analysis:

The system includes the following aspects for cost benefit analyses:

1. Economic aspect :

All required programs and software that needed in our system are available for certain cost:

- Visual studio .NET 2005.
- SQL server 2000.
- Microsoft office 2003.
- HyperTerminal.

2. 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:

✓ **Risks :**

1. Bad analysis leads to new requirements during the construction of the system or after implementation.
2. Services shortages such as no credit in the intermediate mobile, or no charge in it.
3. PC suite software required updates each period to continue in operation.
4. Possibility of increasing the cost for building the system.

✓ **Solution of risks :**

1. Study the system deeply to determine all requirements needed to build the system.
2. Use devices with high capabilities.
3. PC suite must be available in any time.

1.6 Economic study:

1) For development :

Hardware:

Item	Number of unit	Cost per unit	Cost of units
PC p4, 504 MB RAM, HD 20 GB, 3 GHz CPU with 2.78 GHz speed.	2	$450	$900
Flash memory 1G	4	$20	$80
Connection cable USB port	1	$7	$7
Nokia mobile 6100	1	$130	$130
Nokia mobile	1	$50	$50
printer	1	$120	$120
Total			**$1287**

Table (1,1) Development hardware resources and costs .

Software:

Item	Number of unit	Cost per unit	Cost of units
Microsoft windows XP	2	$100	$200
Visual studio.NET 2005	1	$100	$100
Microsoft office 2003	2	$250	$500
SQL server 2000	1	$120	$120
Nokia pc suite for Nokia 6100	1	$70	$70
Visio	1	$50	$50
HyperTerminal	1	$10	$10
Photoshop	1	$70	$70
Macromedia Flash	1	$60	$60
Total			**$1180**

Table (1,2) Development software resources and costs .

Human :

Persons	No	Cost/month
Programmer and Web Designer	2	$500
Database developer and documentation.	2	$500
Total		**$1000**

Table (1, 3) Development human resources and costs

2) For operation:

Hardware:

Item	Number of unit	Cost per unit	Cost of units
PC p4 , 504 MB RAM , HD 20 GB , 3 GHz CPU with 2.78 GHz speed .	1	$450	$450
Connection cable USB port	1	$7	$7
Nokia mobile 6100	1	$130	$130
Nokia mobile	1	$50	$50
Total			**$637**

Table (1,4) Operation hardware Resources and costs .

Software:

Item	Number of unit	Cost per unit	Cost of units
Microsoft windows XP	1	$100	$100
Visual Basic.net 2005	1	$100	$100
SQL server 2000	1	$120	$120
Nokia pc suite for Nokia 6100	1	$70	$70
Total			**$390**

Table (1,5) Operation software Resources and costs .

Human :

Persons	No	Cost/month
Administrator	1	$500
Assistance	1	$500
Total		**$1000**

Table (1,6) Operation human Resources and costs .

2) **For operation:**

Hardware:

Item	Number of unit	Cost per unit	Cost of units
PC p4 , 504 MB RAM , HD 20 GB , 3 GHz CPU with 2.78 GHz speed .	1	$450	$450
Connection cable USB port	1	$7	$7
Nokia mobile 6100	1	$130	$130
Nokia mobile	1	$50	$50
Total			**$637**

Table (1,4) Operation hardware Resources and costs .

Software:

Item	Number of unit	Cost per unit	Cost of units
Microsoft windows XP	1	$100	$100
Visual Basic.net 2005	1	$100	$100
SQL server 2000	1	$120	$120
Nokia pc suite for Nokia 6100	1	$70	$70
Total			**$390**

Table (1,5) Operation software Resources and costs .

Human :

Persons	No	Cost/month
Administrator	1	$500
Assistance	1	$500
Total		**$1000**

Table (1,6) Operation human Resources and costs .

The overall costs of our project summarized in the following table :

For development	
hardware	$1287
software	$1180
human	$4000
Total	**$7467**

Table (1,7) overall cost for development .

For Operation	
hardware	$637
software	$390
Total	**$1027**

Table (1,8) overall cost for operation .

1.8 Time scheduling:

In this part we will show all tasks that should be finished within the specified period in fifteen weeks. These tasks shown below:

Number of Task	Tasks	Time in weeks
T1	Information collection	3
T2	System Planning	2
T3	Software Requirement	3
T4	System design	2
T5	Coding and Implementation	5
T6	System testing	1
T7	Error corection	1
T8	Documentation	15

Table (1,9) Time schedule in weeks.

Task allocation vs. Time:

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure (1,1) Gantt chart for time schedule .

Software Requirements Specification

Chapter Two

- *Introduction*
- *Requirements Specification*
- *Functional Description*
- *Validation Criteria*
- *Information Description*
  - ✓ *Data Flow Diagram.*
  - ✓ *Data Dictionary.*
  - ✓ *System Interface Description.*
  - ✓ *Database Requirements.*
  - ✓ *Database Model.*

2.1 Introduction:

In this phase we will indicate the requirements specifications, functional description, validation criteria, for the system and analyses it, and we will describe all of the system data and clarify the requirements of database.
Also this chapter will include:

- Context diagram
- Data Flow Diagram

2.2 Requirements specifications:

A. Student requirements specifications

1. Authentication :
   - The student requests his information by using the mobile or by using the portal.
   - The student insert his username , password , mobile number and query respectively when he used SMS , or insert his username and password and select the query from the web form when he use the portal.
   - If the student inserts a valid input he will see his information according to the query, such as marks, academic status, financial status and registered courses.

2. Student check marks:

   ☒ By using SMS:

   Sending SMS that contain student number, password, and mobile number, (M) shortcut for mark query, semester and year. The student will get the following information:

   - Courses name
   - Courses marks

☒ By using portal :

By clicking the marks link, the student will see his marks for all semester since he starting study until the last semester, this can display by choosing the year and the semester from dropdown list, so the student can see the marks for each semester, the result of this request will be displayed in the list as :

- Course number.
- Course name.
- Mark.
- Credit hours

3. Check academic status:

☒ By using SMS:

Sending SMS that contain student number, password, mobile number, and (AS) shortcut for academic situation query. The student will get the following:

- Accumulative average.
- Major average.
- Grade.

☒ By using portal :

The student will show his academic information from the first semester until the current semester. When the student chooses the academic status he will see:

- Student full name.
- Student number.
- College name and major name.
- Current level.
- Averages: Accumulative average, major average and semester average.

- Honors: for the graduated student if he has an honor or not.
- Adjourning: specifies if the student regular or not.
- Scientific degree :(BA or Diploma).
- Student status: graduated or not.
- Warning: specifies the warning type and date that student has gained.

4. Check financial status:

☒ By using SMS :

Sending SMS that contains student number, password, mobile number, (FS) shortcut for financial situation query, semester, and year. The student will get the following:

- Balance (credit and debit).
- Financial assistances: aids, scholarship and other.

☒ By using portal :

In this choice the student will see the financial status for each year that contain the the following information:

- Student full name.
- Student number.
- Assistances.
- Semester balance.
- Total balance.
- Credit and debit for each semester.

5. Registered courses:

➢ By using SMS:

Sending SMS that contain student number, password, mobile number, and (RC) shortcut for registered courses query. The student will get the following:

- Courses name.
- Courses credit hours.

➢ By using portal:

When the student selects this choice, he will show the courses that he registered for the current semester. This will contain the following information:

- Student name and number.
- The courses name
- Credit hours.
- Courses number.
- Courses type.

6. Modify user information:

In this case the student can change his password, by entering his new password and confirm it, by using portal and SMS.

B. Administrator requirements specifications:

1. Create new student :

Provide administrator the ability to insert all information for new student such as student personal information like student name, student number, student major, tawjihi average and telephone number.

2. modify user information:

This process includes:

a) Update student information :

Administrator can update student personal information, academic situation, financial situation, registered courses and conversion.

b) Delete students :

Administrator can delete student account.

c) Students accounts:

In this form administrator will show all the authenticated students with their ID's and their password.

d) Change password :

Administrator 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.

3. Authentication :

Administrator can log in to his web form by using user name and password.

2.3 Functional descriptions:

1. Log in as student.

Function: log in as student.

Description: identify the student by using valid username and password.

Inputs: student user name and password.

Source: student and log in form.

Outputs: student form.

Destination: SQL database server.

Requires: valid log in.

Precondition: no thing.

Post condition: the student form display.

Figure (2.1) student log in

2. Student check marks using SMS.

Function: student check marks.

Description: provide ability to browse the student's marks from first semester until the current semester.

Inputs: student number, password, mobile number, M, semester, year.

Source: Student and SMS.

Outputs: student marks.

Destination: SQL database server.

Requires: valid student number, password, mobile number, M, semester, year then click send button in mobile.

Precondition: valid inputs.

Post condition: The student can get his courses marks.

Figure (2, 2) student check marks by SMS.

3. Student checks academic status using SMS.

Function: student checks academic status.

Description: provide ability to browse the student academic status (Accumulative average, major average and grade).

Inputs: student number, password, mobile number, AS, semester, year.

Source: Students and SMS.

Outputs: student academic status.

Destination: SQL database server.

Requires: valid student number, password, mobile number, AS, semester, year.

Precondition: valid inputs.

Post condition: The student can get his academic status.

Figure (2.3) student checks academic status using SMS

4. Student checks financial status using SMS.

Function: student checks financial situation.

Description: provide ability to browse the student financial status (balance And assistances).

Inputs: student number, password, mobile number, FS, semester, year.

Source: Students and SMS.

Outputs: student financial status.

Destination: SQL database server.

Requires: valid student number, password, mobile number, FS, semester, year.

Precondition: valid inputs.

Post condition: The student can get his financial status.

Figure (2.4) student checks financial status using SMS

5. Student registered courses using SMS.

Function: student registered courses.

Description: this function display the courses student register for the current Semester.

Inputs: student number, password, mobile number, RC.

Source: Students and SMS.

Outputs: student registered courses.

Destination: SQL database server.

Requires: valid student number, password, mobile number, RC.

Precondition: valid inputs.

Post condition: the student can see the registered courses

Figure (2.5) student registered courses using SMS.

6. Student checks marks using portal

Function: student checks marks.

Description: provide ability to browse the student's marks for each semester From when he start until the current semester.

Inputs: select the semester and the year from the drop down list.

Source: Student and the student web form.

Outputs: student marks.

Destination: SQL database server.

Requires: valid student log in and select marks link from student form.

Precondition: student form and no marks displayed.

Post condition: The student marks displayed.

Figure (2.6) student checks marks using portal

7. Student checks academic status using portal

Function: student academic status.

Description: provide ability to stunt to browse his academic status.

Inputs: select the academic status link from the student web form.

Source: Student and the student web form.

Outputs: student academic status.

Destination: SQL database server.

Requires: valid student log in and select academic status link from student form.

Precondition: student form and no academic status displayed.

Post condition: The student academic status displayed.

Figure (2.7) student checks academic status using portal

8. Student checks financial status using portal

Function: student financial status.

Description: provide ability to student to browse his financial status.

Inputs: select the financial status link from the student web form.

Source: Student and the student web form.

Outputs: student financial status.

Destination: SQL database server.

Requires: valid student log in and select financial status link from student form.

Precondition: student form and no financial status displayed.

Post condition: The student financial status information displayed.

Figure (2.8) student checks financial status using portal

9. Student registered courses using portal

Function: student registered courses.

Description: provide ability to student to browse his registered courses for the Current semester.

Inputs: select the registered courses link from the student web form.

Source: Student and the student web form.

Outputs: student registered courses for the current semester.

Destination: SQL database server.

Requires: valid student log in and select registered courses link from student form.

Precondition: student form and no registered courses displayed.

Post condition: The student registered courses information displayed.

Figure (2.9) students registered courses using portal

10. Student changes his password using portal.

Function: student change his password.

Description: this function allows the student to change his password.

Inputs: User Name, old password and new password and clicking change Button.

Source: student and change the password form.

Outputs: new password saved in database.

Destination: SQL database server.

Requires: inserting old password correctly.

Precondition: insert old password correctly and new password.

Post condition: new password saved in database.

Figure (2.10) student change his password using portal

11. Student log out

Function: student log out.

Description: This function allows the student to log out from his form.

Inputs: click log out link at student form.

Source: student.

Outputs: the student log out from his form.

Destination: SQL database server.

Requires: Nothing.

Precondition: click on log out link.

Post condition: The student log out from the form.

Figure (2.11) student log out operation

12. Administrator updates the student's information.

Function: administrator updates the student's information.

Description: this function allow the administrator to update the students Information.

Inputs: student number and click on update button.

Source: administrator form.

Outputs: student new information.

Destination: SQL database server.

Requires: valid student indicated number.

Precondition: login.

Post condition: the new information saved in database.

Figure (2.12) administrator updates the student's information

13. Log in as administrator.

Function: log in as administrator.

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

Inputs: User Name and password.

Source: administrator and log in form.

Outputs: administrator form and menu.

Destination: SQL database server.

Requires: valid username and password.

Precondition: login.

Post condition: the administrator form display.

Figure (2.13) log in as administrator

14. Administrator changes his password.

Function: administrator changes his password.

Description: this function allows the administrator to change his password.

Inputs: User name, old password and new password and clicking change button.

Source: administrator and change the password form.

Outputs: new password saved in database.

Destination: SQL database server.

Requires: inserting old password correctly.

Precondition: insert old password correctly and new password then confirm it.

Post condition: new password saved in database.

Figure (2.14) administrator change his password

15. Administrator show students accounts.

Function: show student accounts.

Description: this function allow the administrator to identify the student using Username and password.

Inputs: administrator User Name and password and clicking student account Button.

Source: administrator and his form.

Outputs: the student account form.

Destination: SQL database server.

Requires: inserting student number correctly.

Precondition: login.

Post condition: The student account form display.

Figure (2.15) administrator show students account

16. Administrator add new student

Function: Add new student.

Description: this function allows the administrator to add new Student.

Inputs: student number, major, level and clicking new button.

Source: administrator and insert information form.

Outputs: the new student information saved in database.

Destination: SQL database server.

Requires: student number, major and level.

Precondition: click on insert new student button.

Post condition: The student information saved in database by clicking save Button.

Figure (2.16) Administrator add new student

17. Administrator delete student

Function: Administrator delete student.

Description: This function allows the administrator to delete student.

Inputs: student number and clicking delete button.

Source: administrator and delete information form.

Outputs: the information for the student is deleted from the database.

Destination: SQL database server.

Requires: student number.

Precondition: click on delete button.

Post condition: The student information deleted from database by clicking Delete button.

Figure (2.17) Administrator delete student

2.4 validation Criteria:

To avoid user errors we have to meet all required validation to guide the student to obtain maximum benefit while using student portal using SMS system, these validation criteria as show below:

1) For the administrator :

- Password must be at least six character lengths for all accounts.
- Password should not contain any space or special character.
- Password may be the same for many users but the username must different.
- Each user input should be checked against all constraint.

2) For the student to write the SMS :

- Student should use the indicated SMS format to send the message as follow:

✓ To request marks(M) the format should like

Username.....
Password.....
Mobile number.....
M Semester Year.....

Figure (2.18) student request marks SMS format.

✓ To request academic status(As) the format should like

Username.....
Password.....
Mobile number.....
AS Semester Year.....

Figure (2, 19) check academic status SMS format

✓ To request financial status(Fs) the format should like

Username.....
Password.....
Mobile number.....
FS Semester Year.....

Figure (2, 20) check financial status SMS format

✓ To request registered courses(Rc) the format should like

Username.....
Password.....
Mobile number.....
RC.....

Figure (2, 21) check registered courses SMS format

2.5 Information Description:

2.5.1 Context diagram:

This diagram shows the relationship between the student portal using SMS system and external entities:

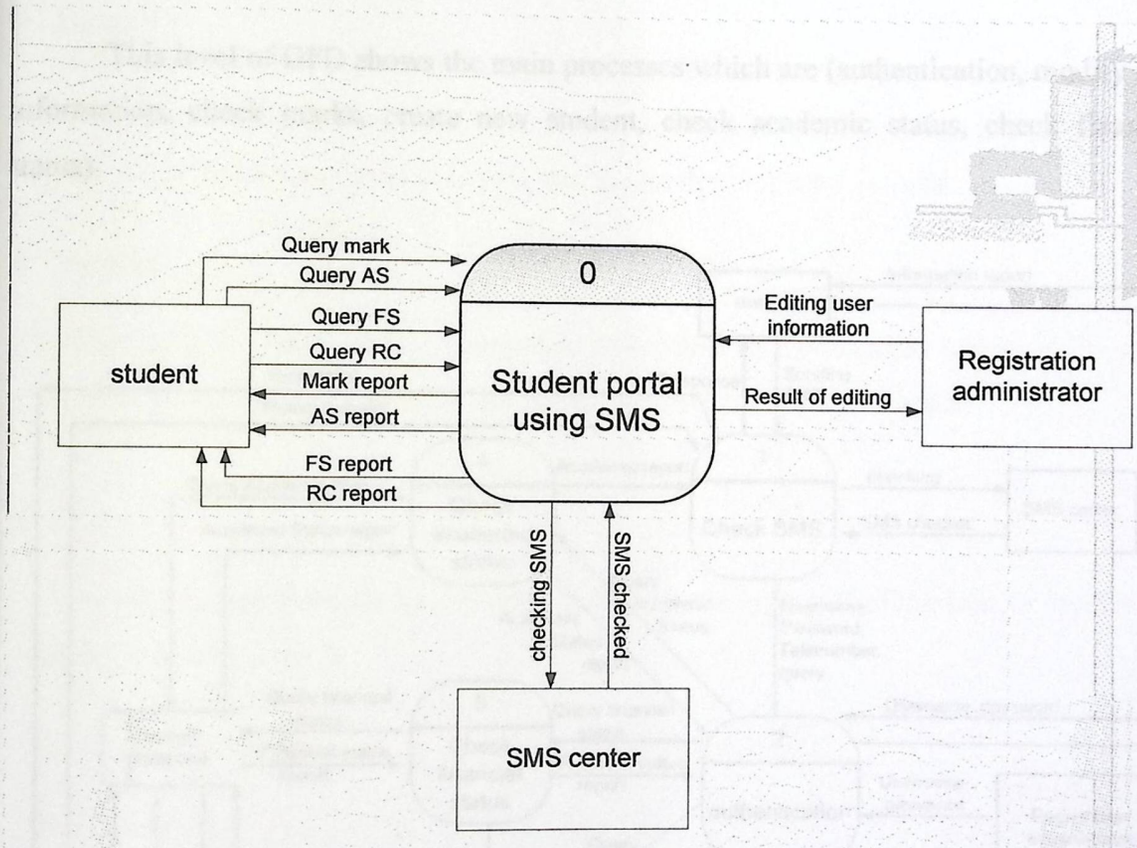


Figure (2, 22) Context Diagram

data flow diagram:

DFD in this system shows the main processes in the system, which we describe in two level of DFD (level0 and level 1).

level 0:

This level of DFD shows the main processes which are (authentication, modify user information, check marks, create new student, check academic status, check financial status).

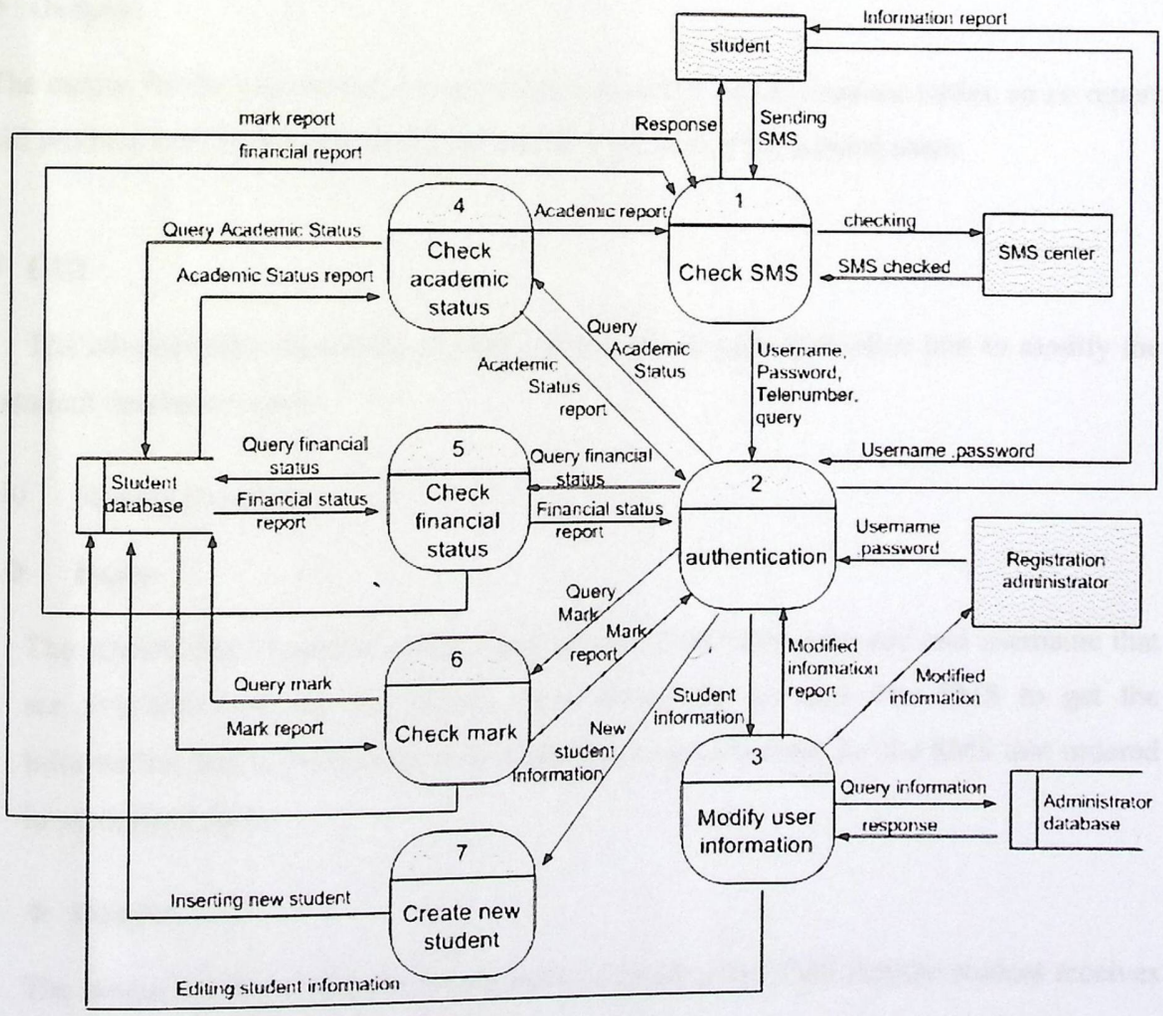


Figure (2, 23) Data Flow Diagram Level (0)

2.5.3 System interface description:

This system will have an interface with two user categories (administrator and student) this section will describe the system interface for each user

a) Administrator interface:

❖ **Input :**

All pages must be secured, the administrator will have accessibility to the database, and this will be done through input tool to facilitate his work to modify records in easy way. Validation techniques are used to ensure that the predefined input is valid.

❖ **Output:**

The output for the administrator transaction will reflect on the database tables, so no report will produce except the reports that should be available for the administrator.

❖ **GUI :**

The administrator should be provided with a log in page that allow him to modify the student database records.

b) Student interface:

❖ **Input**

The student SMS must be secure, each student must have password and username that are available only to the student, they allow him to send The SMS to get the information that he want, and each query has a specific form for the SMS that ordered in predefined form.

❖ **Output**

The output for the student is the response of sending the SMS that the student receives it in the same order that he sends.

2.5.4 Database requirements:

Tables:

1. **student**
   - Student number.
   - First name
   - Last name
   - Password.
   - Major number.
   - Telephone number.
   - City.
   - Gender.
   - Tawjehi average.
   - Tawjehi no.
   - Entry year.
   - Scientific degree.
2. **registered courses**
   - Student number.
   - Course number.
   - Mark.
   - Semester.
   - Year
3. **course**
   - Course number.
   - Course name.
   - Credit hour.
   - Type number.

4. **course type**

   - Type description.
   - Type number.

5. **academic status**

   - student number
   - semester
   - year
   - accumulative average
   - major average
   - grade
   - current level
   - honors
   - adjourning
   - student status

6. **financial status**

   - student number
   - semester
   - year
   - balance
   - credit
   - debit
   - assistances

7. **warnings**

- student number
- Warning number.
- Warning type.
- semester
- year

8. **college**

- college number
- college name

9. **department**

- Department number.
- Department name.
- College number.

10. **major**

- Major number.
- Major name.
- Credit hours.
- Department number.

11. **conversion**

- Student number.
- Semester.
- Year.
- New student number.
- New major number.

12. Tawjehi:

- Tawjehi no.
- Tawjehi type.

13. Administrator:

- Username.
- password

2.5.5 data dictionary :

Entity Name	Description
Log in	This procedure allows the administrator or student to log in to his form.
Log out	This procedure allows the administrator or student to log out from his forms.
Change password	This procedure allows the administrator or student to change his password and saved new password in database.
Create new student	This procedure allows the administrator to insert new student information to database.
edit student information	This procedure allows the administrator to edit student information.
User account	This procedure allows the administrator or to show student account.
Academic status	This procedure allows the student to show his Academic status for all semesters.
Financial status	This procedure allows the student to show his Financial status for all semesters.
Registered courses	This procedure allows the student to show his Registered courses for the current semester and his marks for previous registered courses.

AT command	AT commands are instructions used in hyper terminal to control an intermediate mobile. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That's why are called AT commands.
AT+CMGF=1	This AT command used to ensure that the mobile support text mode.
AT+CMGR	This AT command used to read SMS from mobile.
AT	This AT command used to test connection with mobile phone.
AT+CMGS	This AT command used to send SMS to mobile.

Table (2, 1) Data Dictionary

System Design

Chapter Three

- *Introduction*
- *Functional Design*
  - ✓ *Description.*
  - ✓ *Interface.*
  - ✓ *Constraints*
  - ✓ *Flowchart*
  - ✓ *User interface Design*
- *I/O Design*
- *Database Design*
- *Test Plan*
- *Program Language and Coding*

3.1 Introduction:

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

3.2 Functional design:

Functional design for each module should described in the software system, it describe the interface, the constraints, and the user interface design.

The function design in our system is:

- ✓ Student functions design.
- ✓ Registration administrator function design.

A. Student functions design :

1) Student Log in

✓ ***Description***: this function provides ability for the student to log in to his web form.

✓ ***Interface*** :

- **Input:** student username and password.
- **Output**: student web form if the username and password true, error messages if he enters invalid username or password.

✓ ***Constraints*** :

- Username must number only (6 character at least).
- Password must be 6 characters at least.

- Only the authenticated student can log in to his page.
- The password and username must be entered then checked.
- Username and password must not have any special character (* / # &?).

✓ ***Flowchart:***

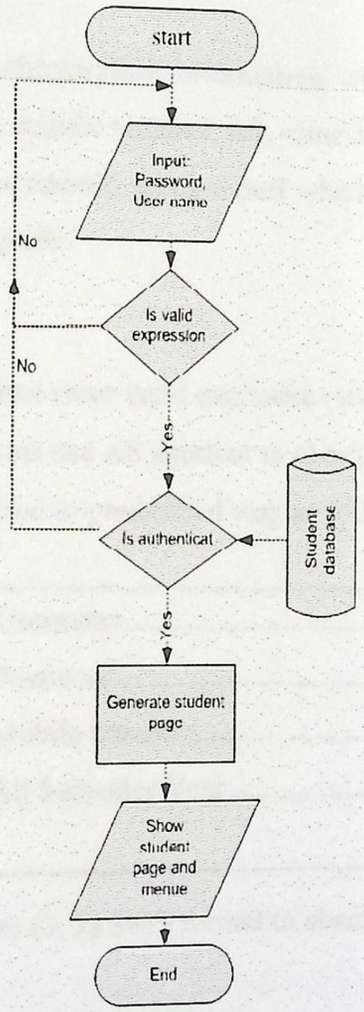


Figure (3, 1) Student log in operation

2) Check Academic Status Using SMS:

✓ ***Description***: this function provides ability to browse the student academic situation (accumulative average, major average, and grade).

✓ ***Interface*** :

- **Input**: authentication information which include student number, student password, mobile number, AS, semester and year.
- **Output**: academic status report which include accumulative average, major average, grade.

✓ ***Constraints*** :

- Student must enter valid username (student number) and Password.
- Student must use AS shortcut to check his academic status.
- Must ordered in predefined way as follow:

Username.....
Password.....
Mobile number.....
AS Semester Year.....

Figure (3, 2) SMS format to check academic status.

✓ ***Flowchart***

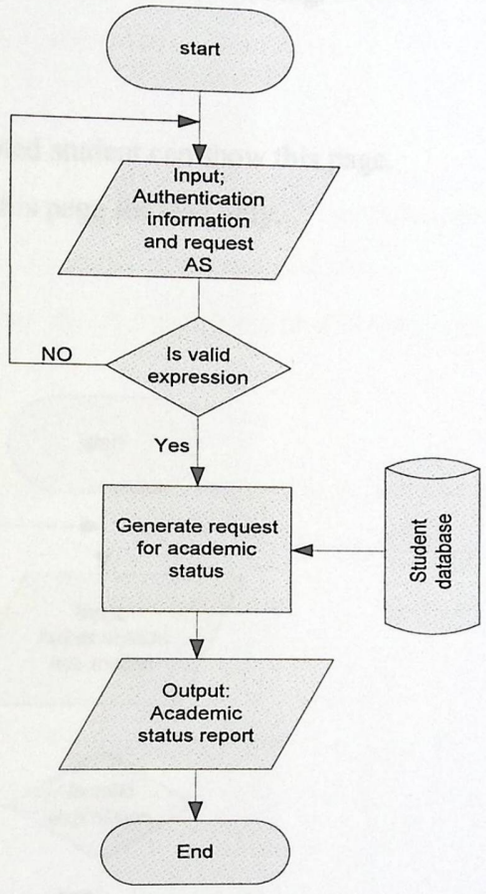


Figure (3, 3) student check academic status using SMS operation.

3) Check Academic Status Using Portal:

✓ ***Description***: this function provides ability to browse the student academic situation by using the portal (accumulative average, major average, grade, warnings…).

✓ ***Interface*** :

- **Input**: authentication information (username and password).
- **Output** : student academic status details which include student full name, student number, College name, Major name, Semester, current level,

Academic year, Accumulative average, Major average, Honors, Scientific degree, Adjourning, Student status, Warning.

✓ ***Constraints*** :

- Only authenticated student can show this page.
- Information at this page for read only.

✓ ***flowchart:***

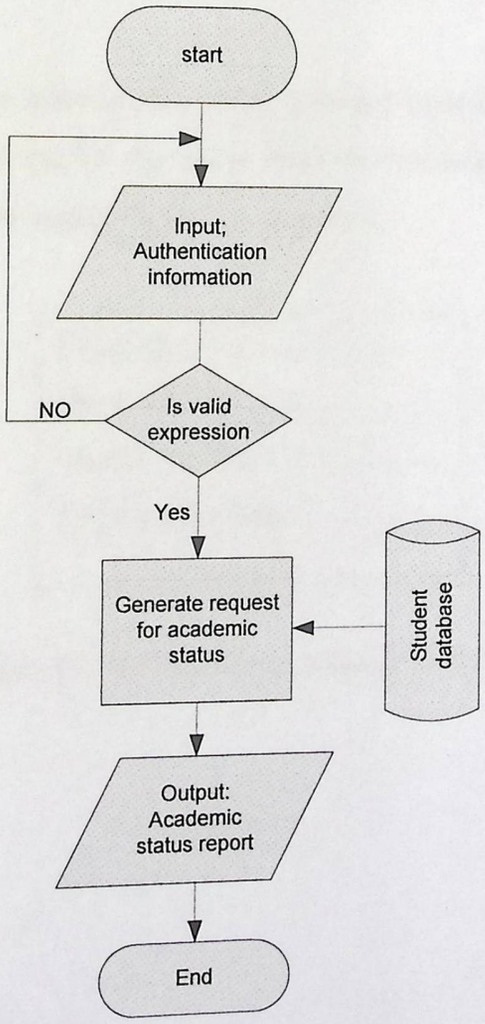


Figure (3, 4) Student checks academic status using portal operation

4) Check Financial Status Using SMS:

- ✓ ***Description***: this function provides ability to student to brows the financial situation (balance and assistances).

- ✓ ***Interface*** :
  - **Input**: authentication information which include student number, student password, mobile number, FS, semester, and year.
  - **Output**: financial status report which include assistances, Semester balance.

- ✓ ***Constraints*** :
  - Student must enter valid username (student number) and Password.
  - Student must use FS shortcut to check his financial status.
  - Must ordered in predefined way as follow

Username.....
Password.....
Mobile number.....
FS Semester Year.....

Figure (3, 5) SMS format to check financial status.

✓ ***flowchart :***

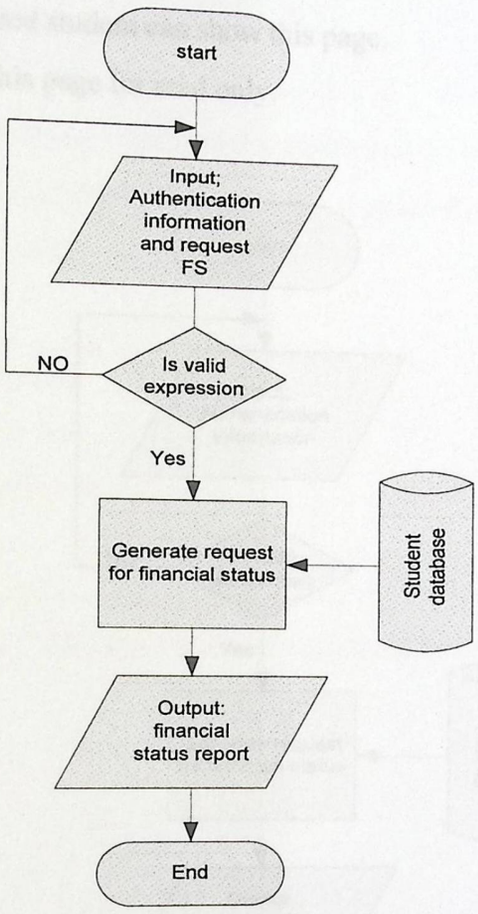


Figure (3, 6) Student checks financial status using SMS operation.

5) Check Financial Status using the portal:

✓ ***Description***: this function provides ability to student to brows the financial status (balance and assistances) by using portal for each semester from he begin until current semester.

✓ ***Interface*** :

- **Input**: authentication information which include student username and student password.
- **Output**: financial status report which include Assistances, Semester balance, Total balance.

✓ ***Constraints*** :

- Only authenticated student can show this page.
- Information at this page for read only.

✓ ***flowchart :***

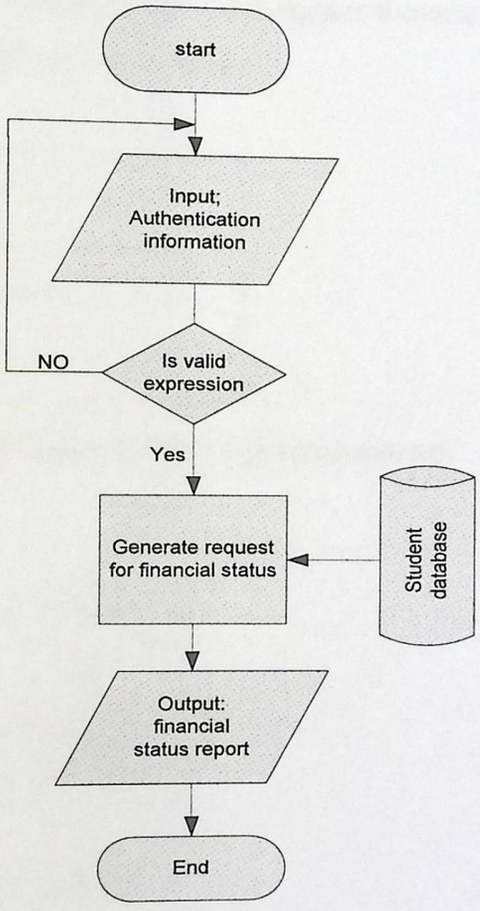


Figure (3, 7) Student financial status using portal operation.

6) Check Registered Courses Using SMS:

✓ ***Description:*** this function allows the student to display the registered courses for the current semester.

✓ ***Interface*** :

- **Input**: authentication information which include (student number), student password, mobile number and RC.

- **Output:** report of registered courses which include registered courses name and credit hours.

✓ ***Constraints*** :

- Student must enter valid username (student number) and Password.
- Student must use RC shortcut to check his registered courses.
- Must ordered in predefined way as follow:

Username.....
Password.....
Mobile number.....
RC.....

Figure (3, 8) SMS format to check registered courses.

✓ ***flowchart :***

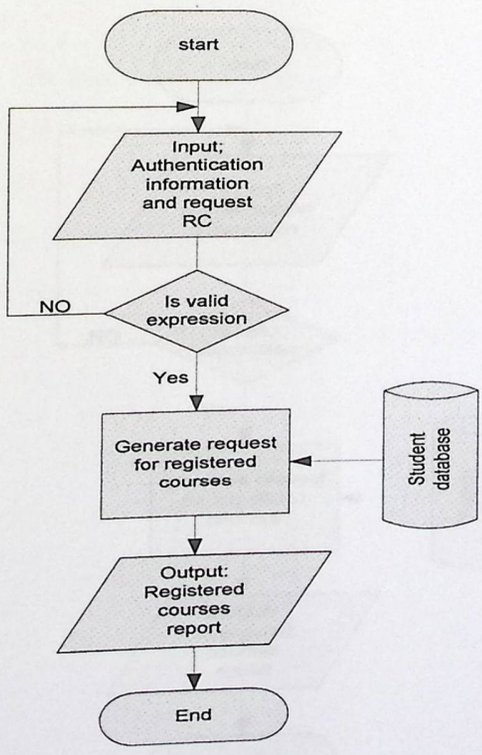


Figure (3, 9) Student check registered courses using SMS operation

7) Check Registered Courses Using the Portal:

✓ ***Description:*** this function allows the student to display the registered courses for the current semester by using the portal.

✓ ***Interface*** :

- **Input** : student number
- **Output**: registered courses report which include Student name, student number, courses name, credit hours, courses number.

✓ ***Constraints*** :

- Only authenticated student, and his administrator can show this page
- Information at this page for read only.

✓ ***Flowchart:***

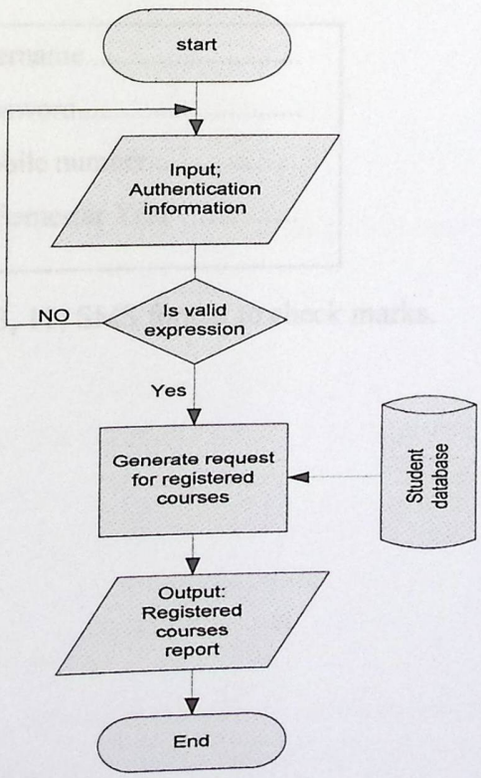


Figure (3, 10) student check registered courses operation using portal

8) Request Marks Using SMS:

✓ ***Description***: this function provides ability for the student to see his marks for each semester.

✓ ***Interface*** :

- **Input:** student username (student number), student password, mobile number and M, semester, year.
- **Output**: marks report.

✓ ***Constraints*** :

- Student must enter valid username (student number) and Password.
- Student must use RC shortcut to check his.
- Must ordered in predefined way as follow:

Username.....
Password.....
Mobile number.....
M Semester Year ……..

Figure (3, 11) SMS format to check marks.

✓ ***Flowchart:***

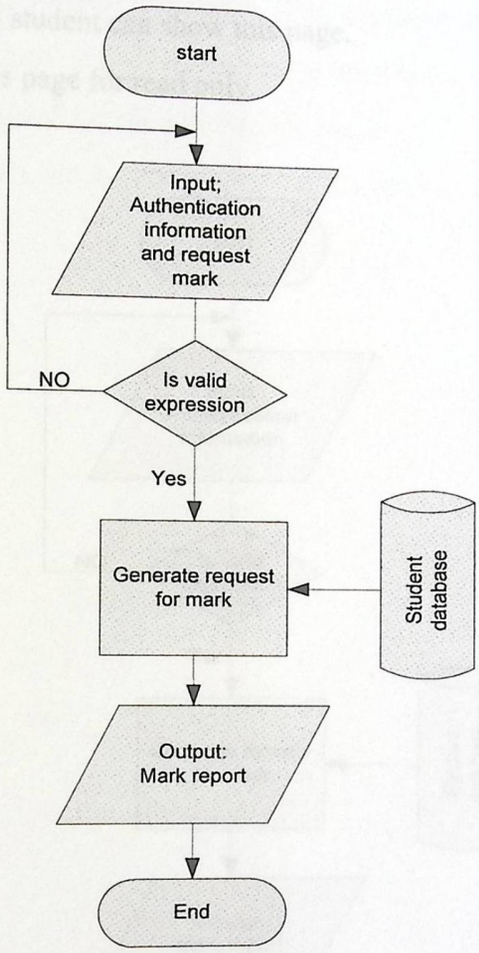


Figure (3, 12) Student request marks using SMS operation

9) Request Marks Using the Portal:

✓ ***Description*** : this function provide ability for the student to see his marks for each semester until last semester.

✓ ***Interface*** :

- **Input:** student number.
- **Output:** marks report which includes Course number, Course name, Marks, Credit hours.

✓ ***Constraints*** :

- Only authenticated student can show this page.
- Information at this page for read only.

✓ ***Flowchart:***

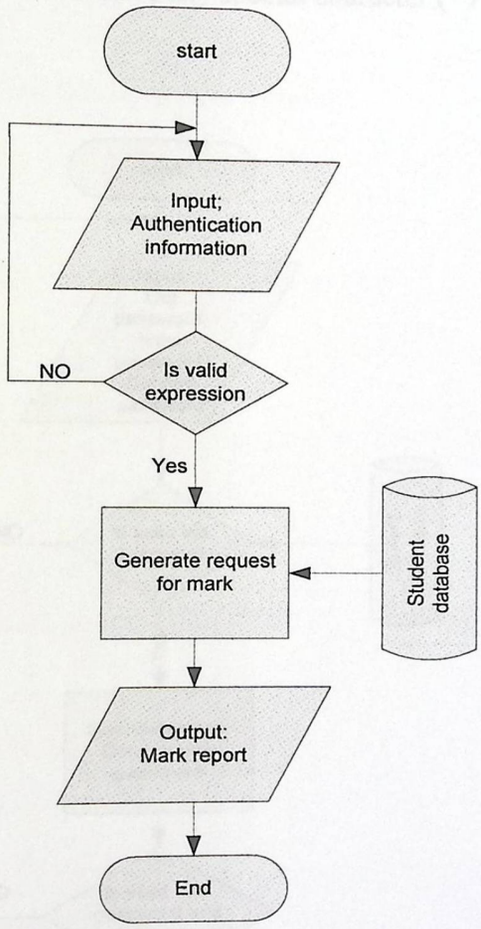


Figure (3, 13) Student request marks using portal operation

10) Change Password:

✓ ***Description***: in this function student has the ability to change his password and replace it by a new one and confirm it.

✓ ***Interface*** :

- **Input:** student old password, new password, and confirm new password.
- **Output**: password updated.

✓ ***Constraints*** :

- New Password must be numbers only (6 character at least).
- The new password and its confirmation must be match.
- New password will be used in the next log in .
- New password must not have any special character (* / # &?).

✓ ***Flowchart :***

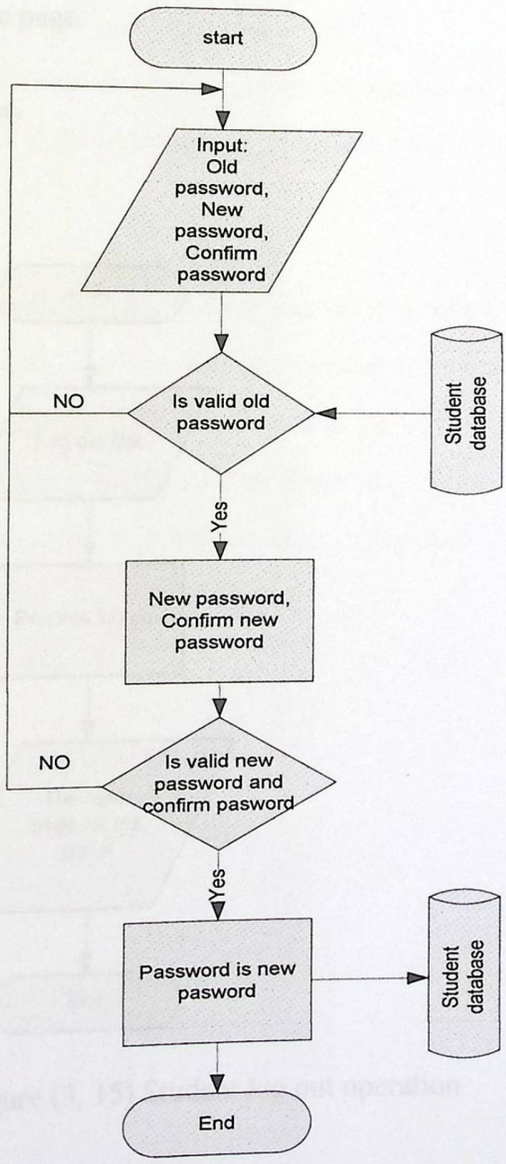


Figure (3, 14) Student change his password using portal

11) Student log out:

- ✓ ***Description***: this function allows the student to log out from his page, and back to the home page.

- ✓ ***Interface*** :
  - **Input:** click at log out link in student forms.
  - **Output**: home page.

- ✓ ***Constraints:*** None.

- ✓ ***Flow chart:***

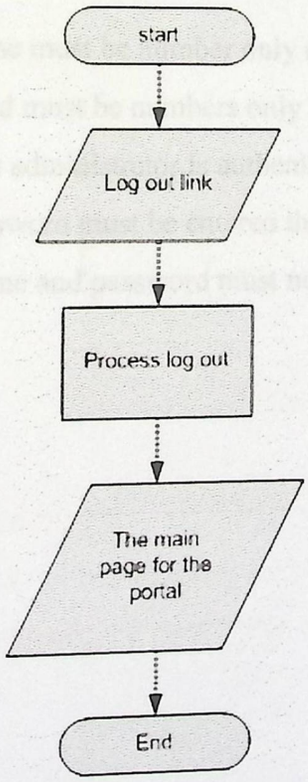


Figure (3, 15) Student log out operation

B. Register administrator function design.

1) Administrator Log in:

✓ ***Description***: this function provides ability for the administrator log in to his web form.

✓ ***Interface*** :

- **Input:** administrator username and password.
- **Output**: administrator web form if the username and password valid, error messages if the username or password invalid.

✓ ***Constraints*** :

- Username must be number only (6 character at least).
- Password must be numbers only (4 character at least).
- Only the administrator is authenticated to log in to this page.
- The password must be entered then checked.
- Username and password must not have any special character (* / # &?).

✓ ***Flowchart:***

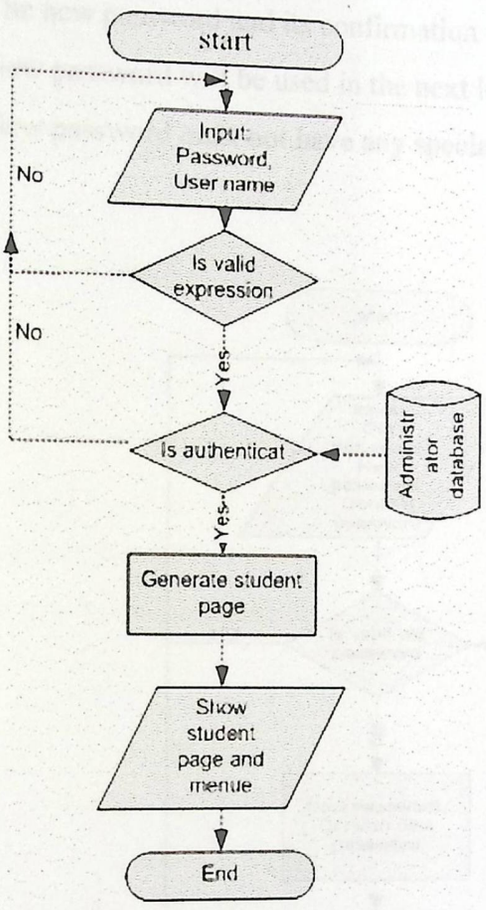


Figure (3, 16) Administrator log in operation

2) Change password :

✓ ***Description*** : in this function Administrator has the ability to change his password and replace it by a new one and confirm it

✓ ***Interface*** :

- **Input:** administrator username old password, and new password, confirm new password.
- **Output**: password updated.

✓ ***Constraints*** :

- New Password must be numbers only (4 character at least).
- The new password and its confirmation must be match.
- New password will be used in the next log in.
- New password must not have any special character (* / # &?).

✓ ***flowchart :***

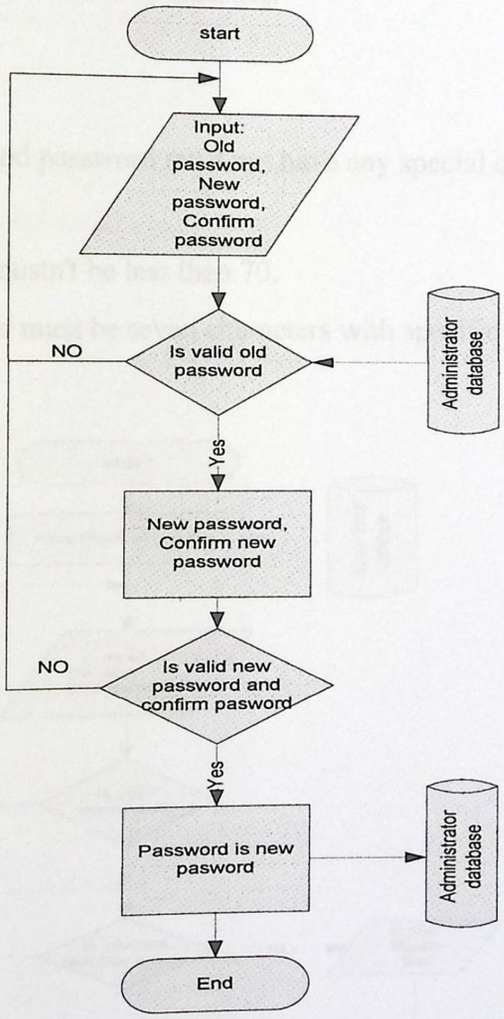


Figure (3, 17) Administrator change password operation.

3) add new student :

✓ ***Description***: in this function Administrator has the ability to add student to the system.

✓ ***Interface*** :

- **Input:** student number, student first name, student last name, password, username, major number, city, telephone number, gender, tawjihi average, Tawjihi type, entry year.

- **Output**: new student added to database.

✓ ***Constraints*** :

- Student number and password must not have any special character (* / # &?).
- Tawjihi average mustn't be less than 70.
- Telephone number must be seven characters with specific format.

✓ ***Flowchart:***

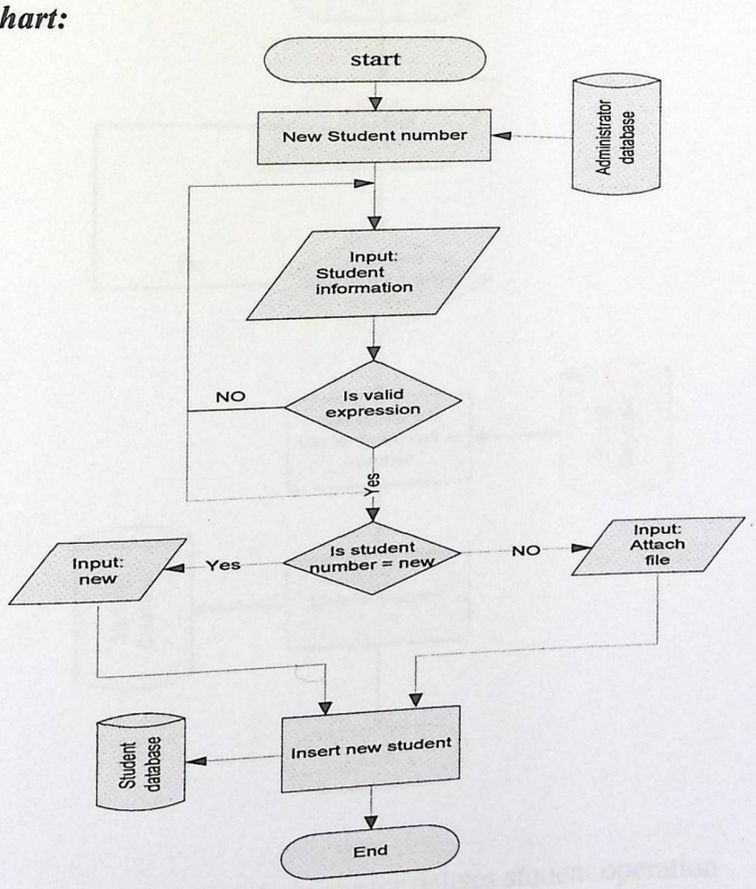


Figure (3, 18) Administrator add new student operation

4) Delete Student :

✓ ***Description***: in this function Administrator has the ability to delete student from any college in the system.

✓ ***Interface*** :

- **Input: student** number.
- **Output**: student deleted from database.

✓ ***Constraints*** :

- All fields of the student information should delete when the student number entered and click at delete button.

✓ ***Flowchart:***

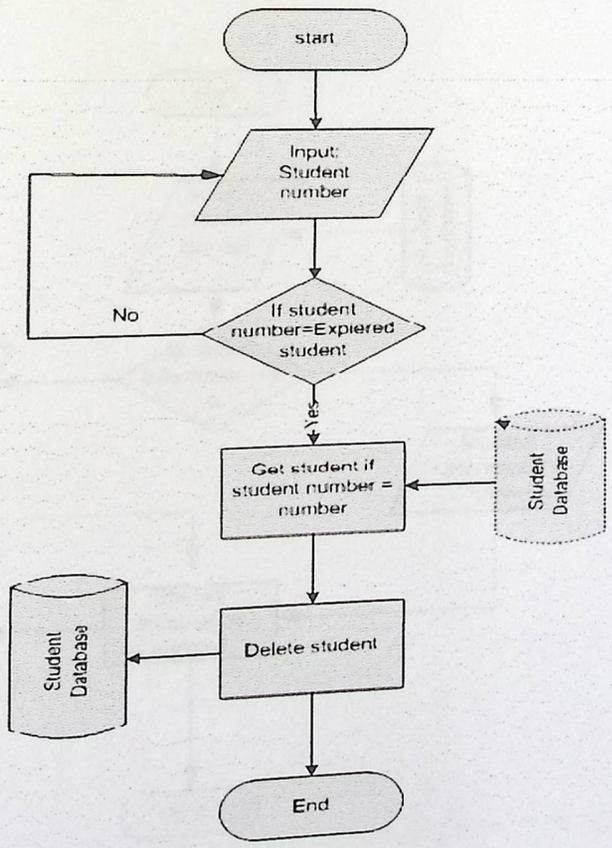


Figure (3, 19) Administrator deletes student operation

5) Update Student Information :

✓ ***Description***: this function allows the Administrator to update all student information.

✓ ***Interface*** :

- **Input:** student number.
- **Output:** student information updated and saved in database.

✓ ***Constraints*** :

- The student updated information must use in the next time to show student information.

✓ ***Flowchart:***

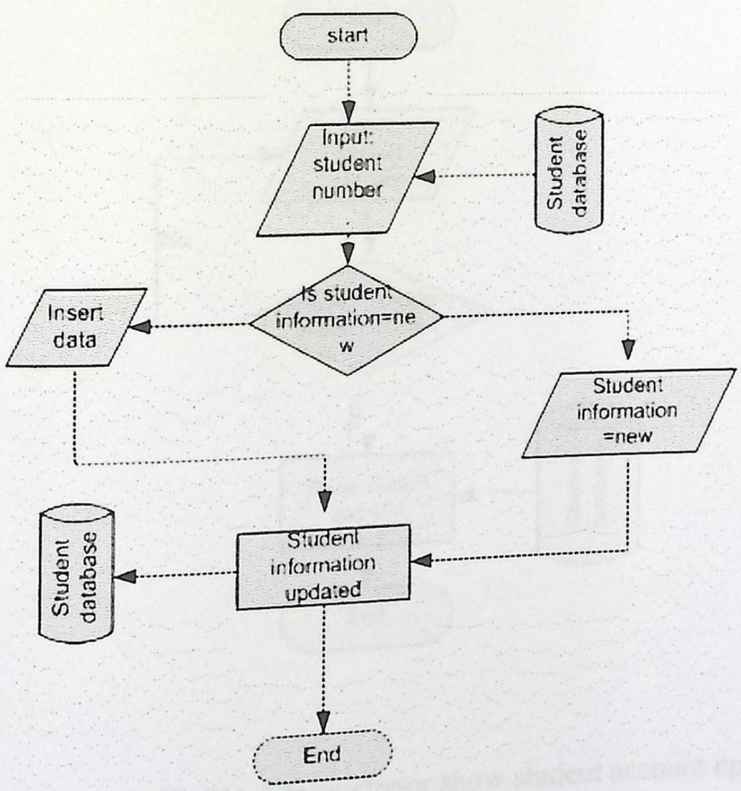


Figure (3, 20) Administrator updates student operation

6) Show Student Account:

✓ ***Description***: in this function Administrator has the ability to show the student account.

✓ ***Interface*** :

- **Input:** student number.
- **Output**: student number (username), password.

✓ ***Constraints*** :

- Only the administrator can show student account.

✓ ***Flowchart:***

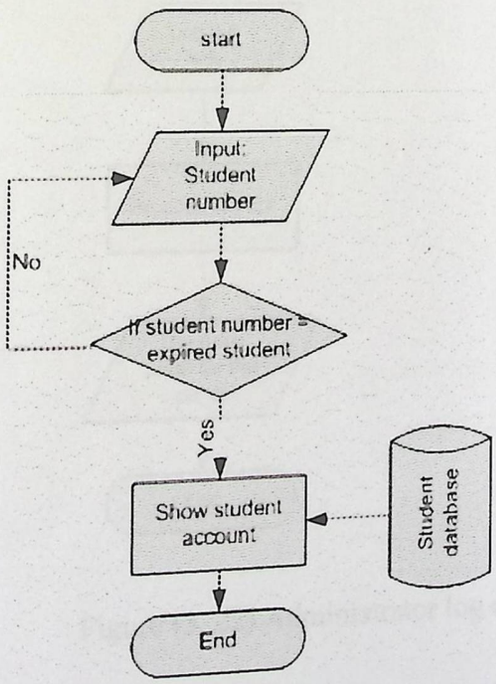


Figure (3, 21) Administrator show student account operation

7) Administrator log out :

- ✓ ***Description***: this function allows the Administrator to log out from his page, and back to the home page.

- ✓ ***Interface*** :
  - **Input:** click at log out link in Administrator forms.
  - **Output**: home page.

- ✓ ***Constraints:*** None.

- ✓ ***flowchart :***

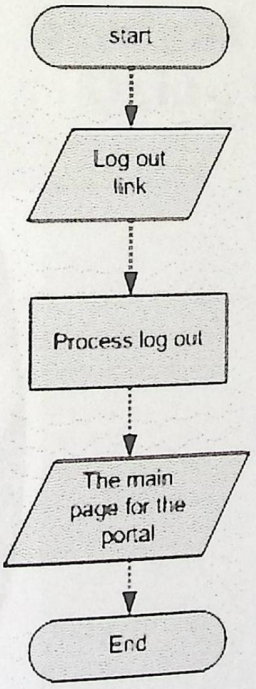


Figure (3, 22) Administrator log out

3.3 *System Interface Design :*

In this section we show the input/output screens. The screens (the designed form) show how the functions work, and how the user can interact with it.

In this system we have two interface designs (student and administrator interface design) in addition to interface for sending and receiving SMS as below:

- **Receiving SMS that sent by student:**

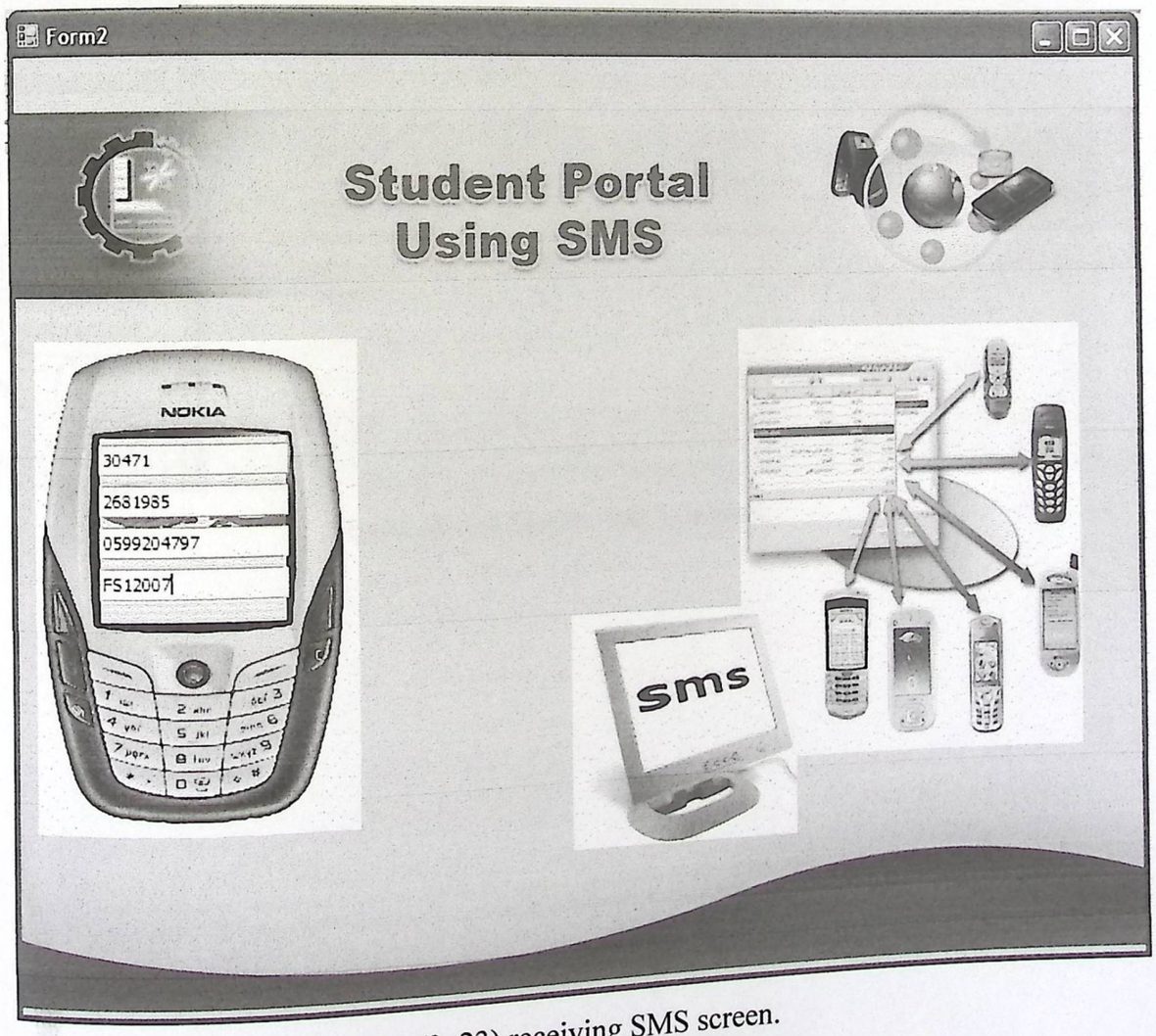


Figure (3, 23) receiving SMS screen.

2) Student Interface Design:

- *Student log in:*

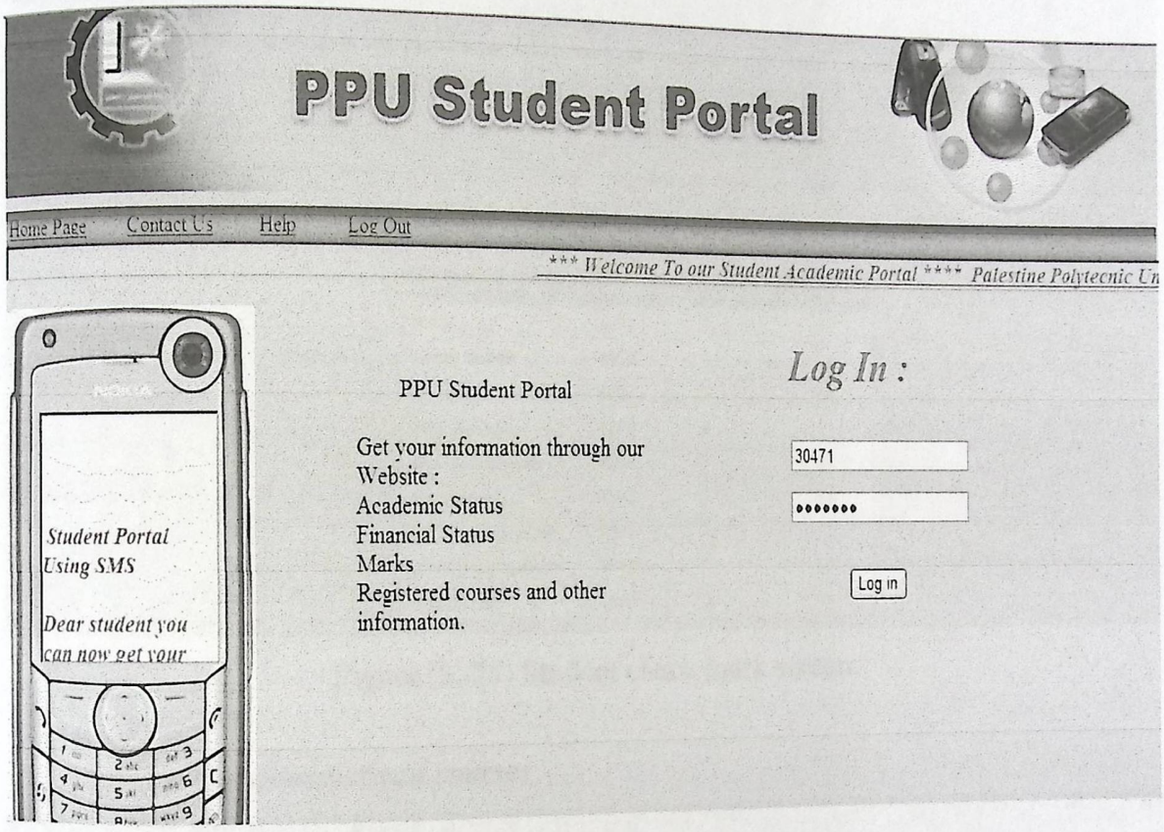


Figure (3, 24) Student log in screen.

- *Check marks*

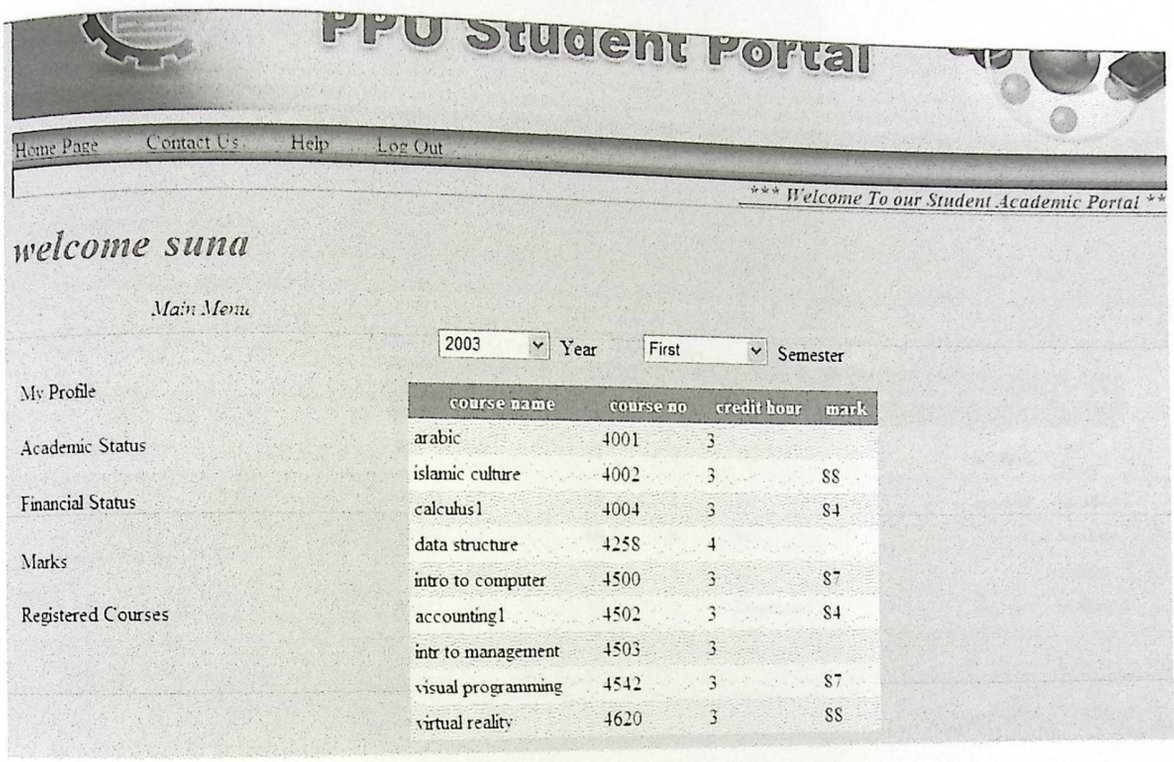


Figure (3, 25) Student check mark screen.

- *Check registered courses*

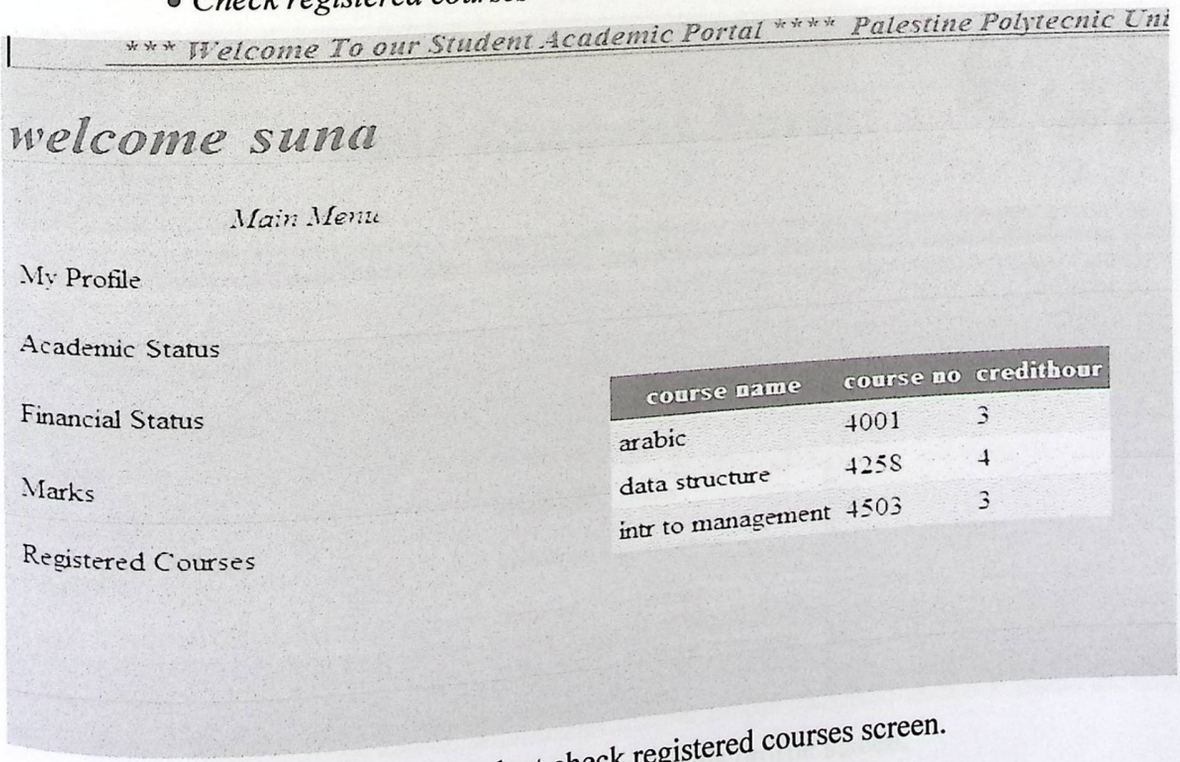


Figure (3, 26) Student check registered courses screen.

- *Check academic status*

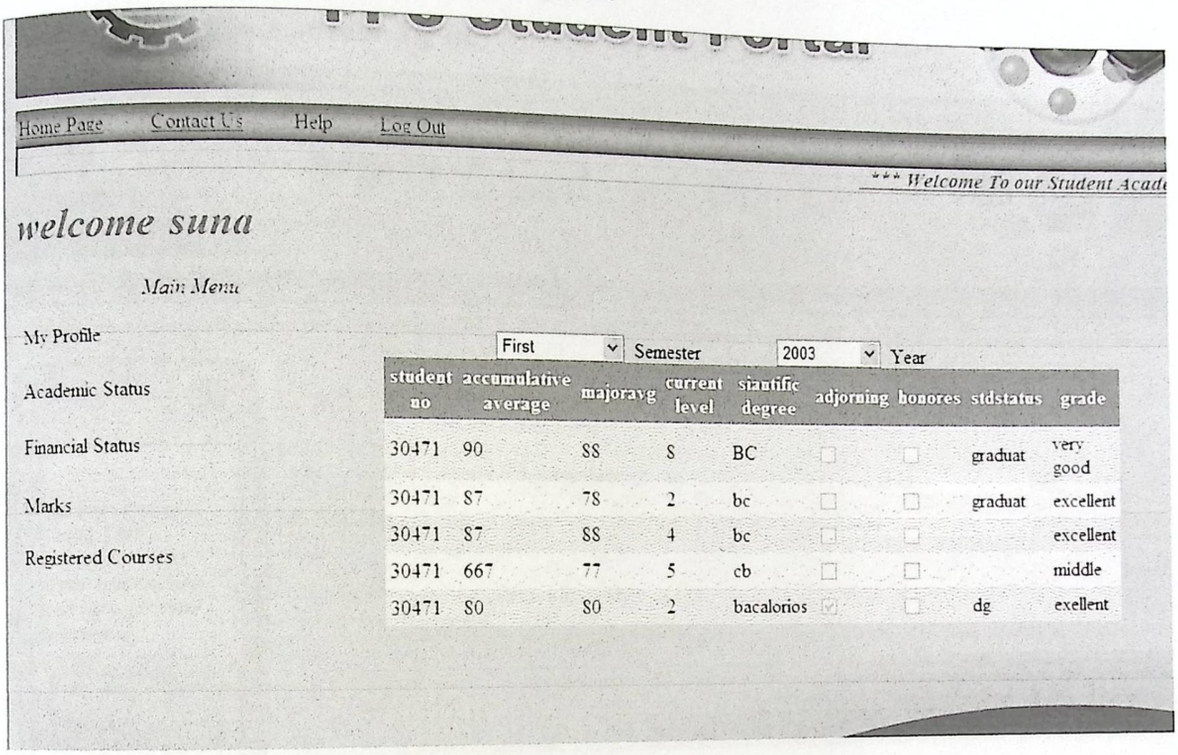


Figure (3, 27) Student check academic status screen.

- *Student profile*

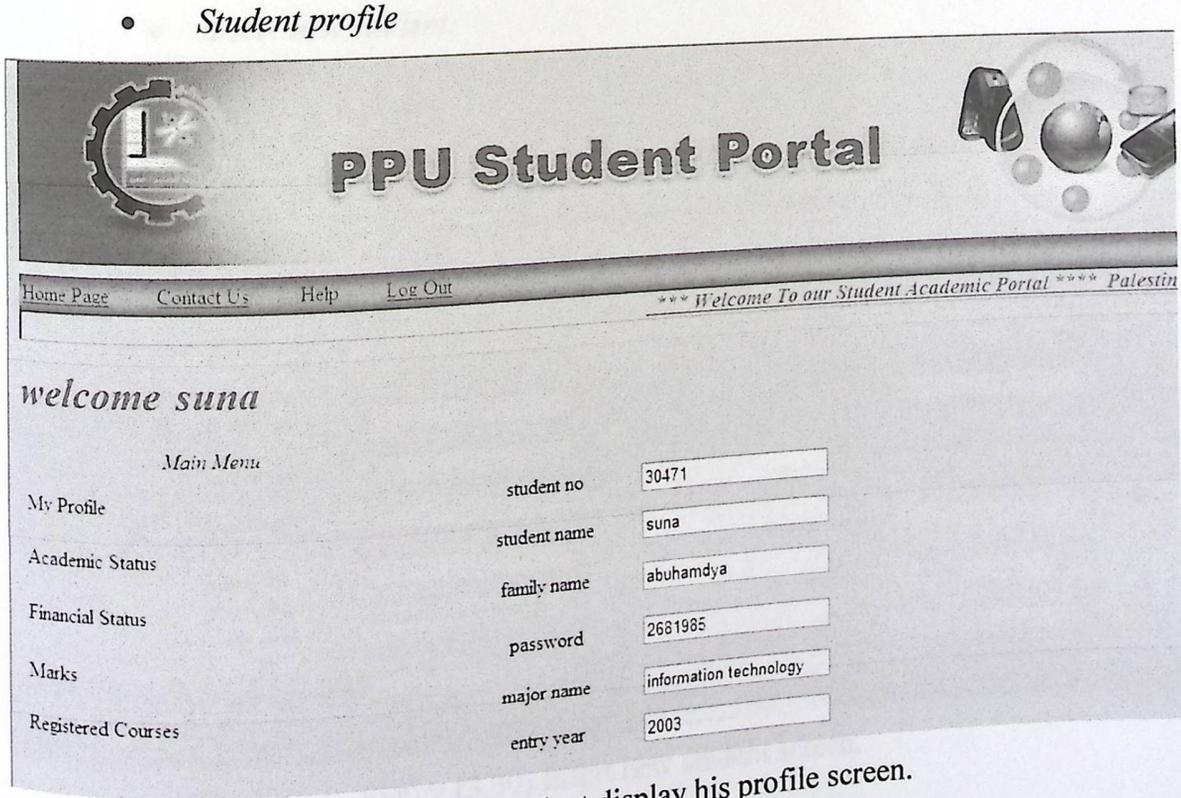


Figure (3, 28) Student display his profile screen.

3) Administrator Interface Design:

- *Administrator log in:*

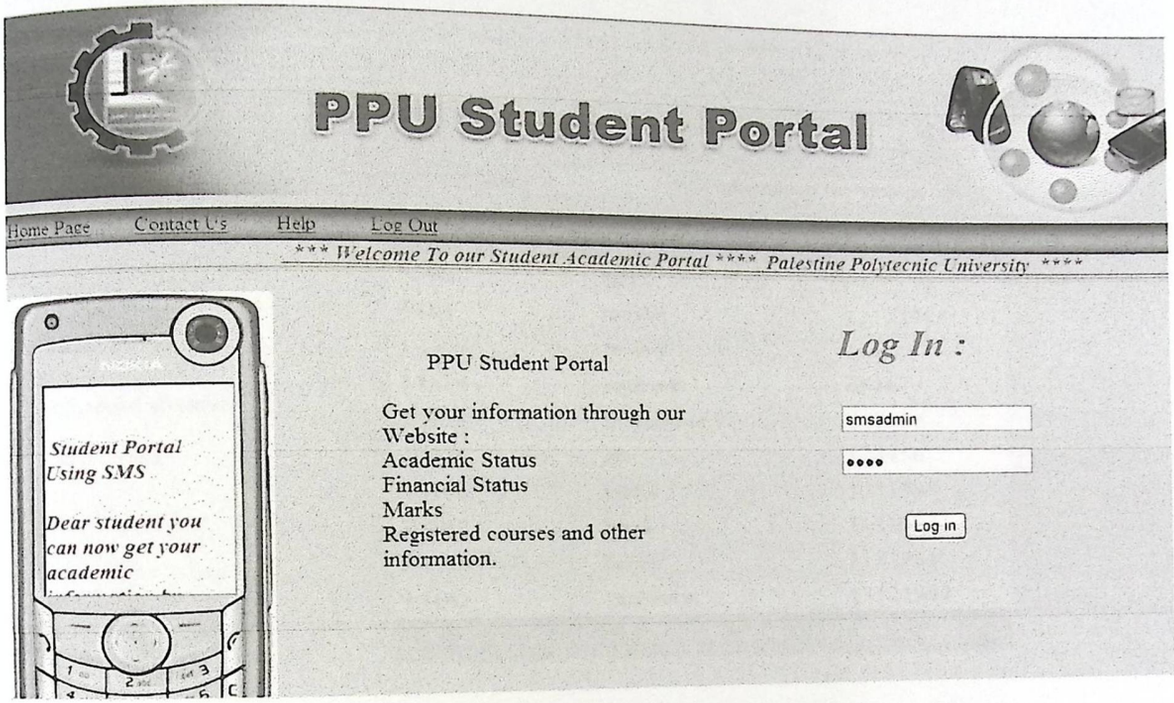


Figure (3, 29) administrator log in screen.

- *Insert new student:*

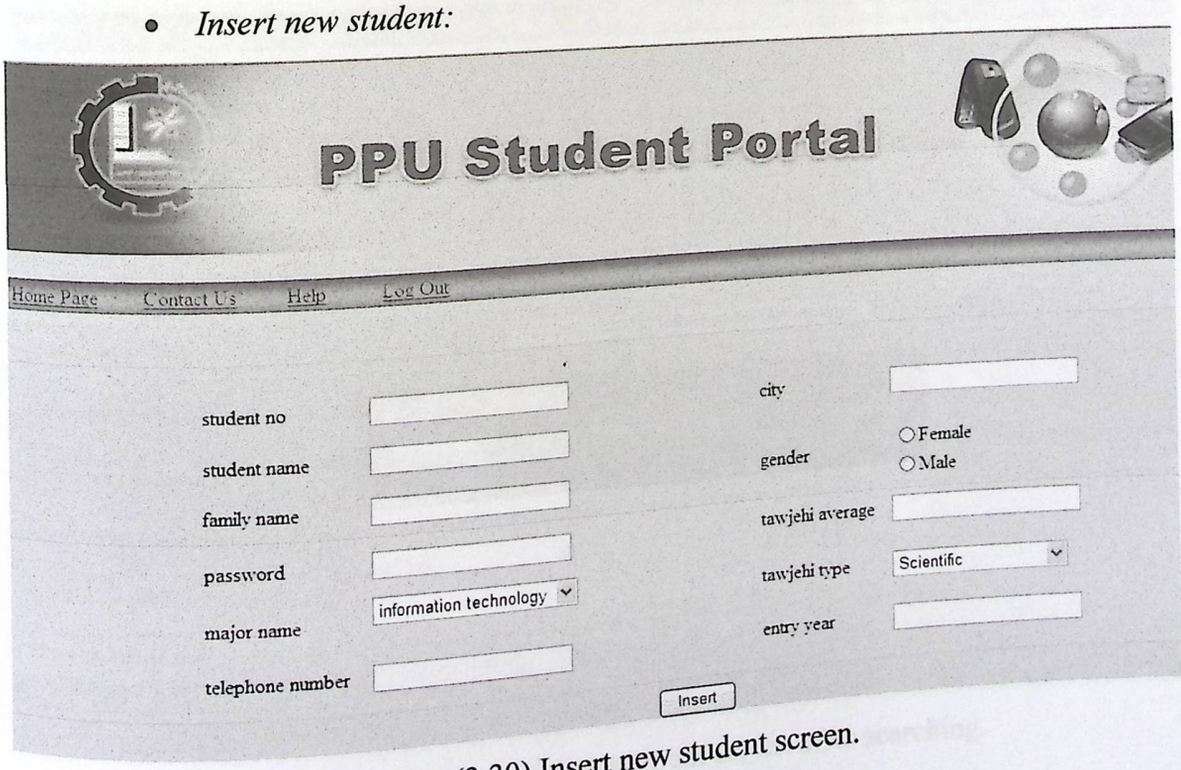


Figure (3,30) Insert new student screen.

- *Show student account:*

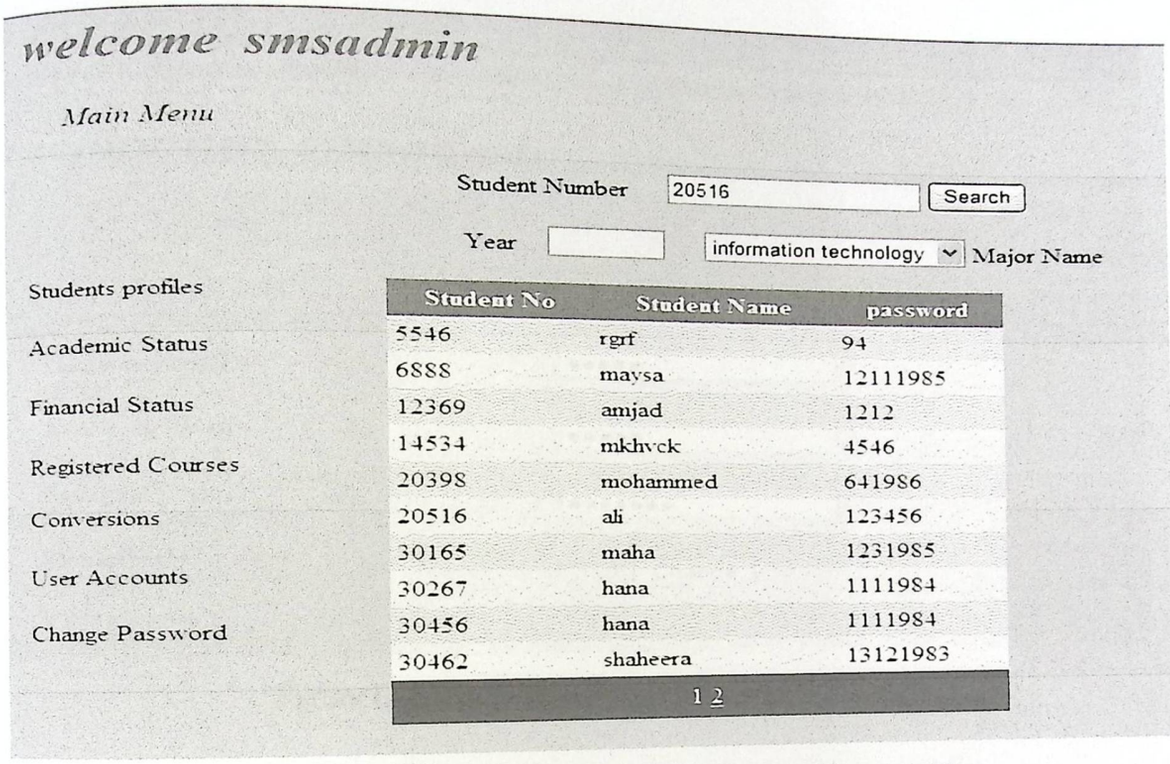


Figure (3,31) display student account screen before searching.

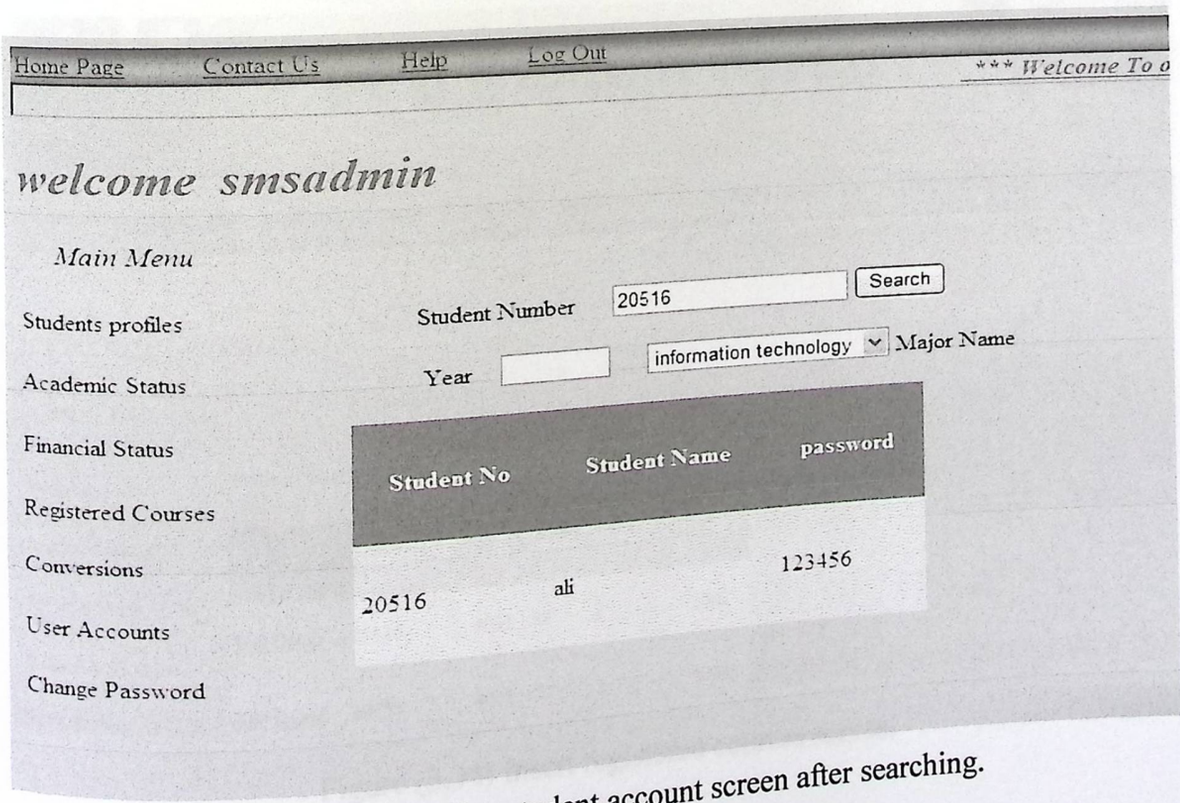


Figure (3,32) display student account screen after searching.

- *Change password :*

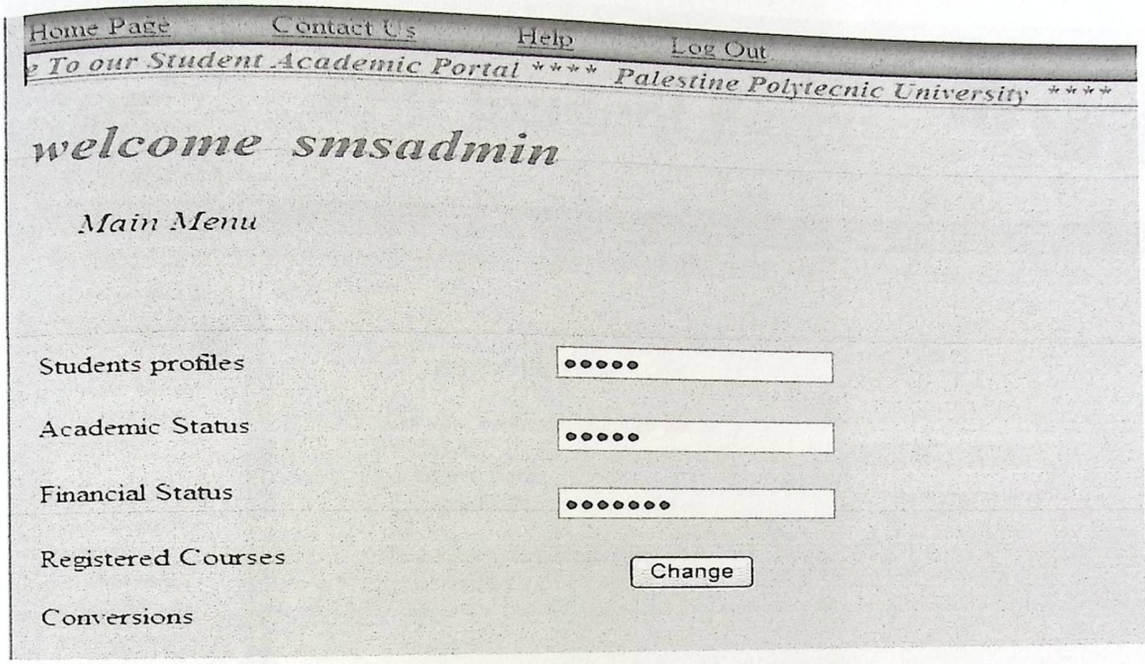
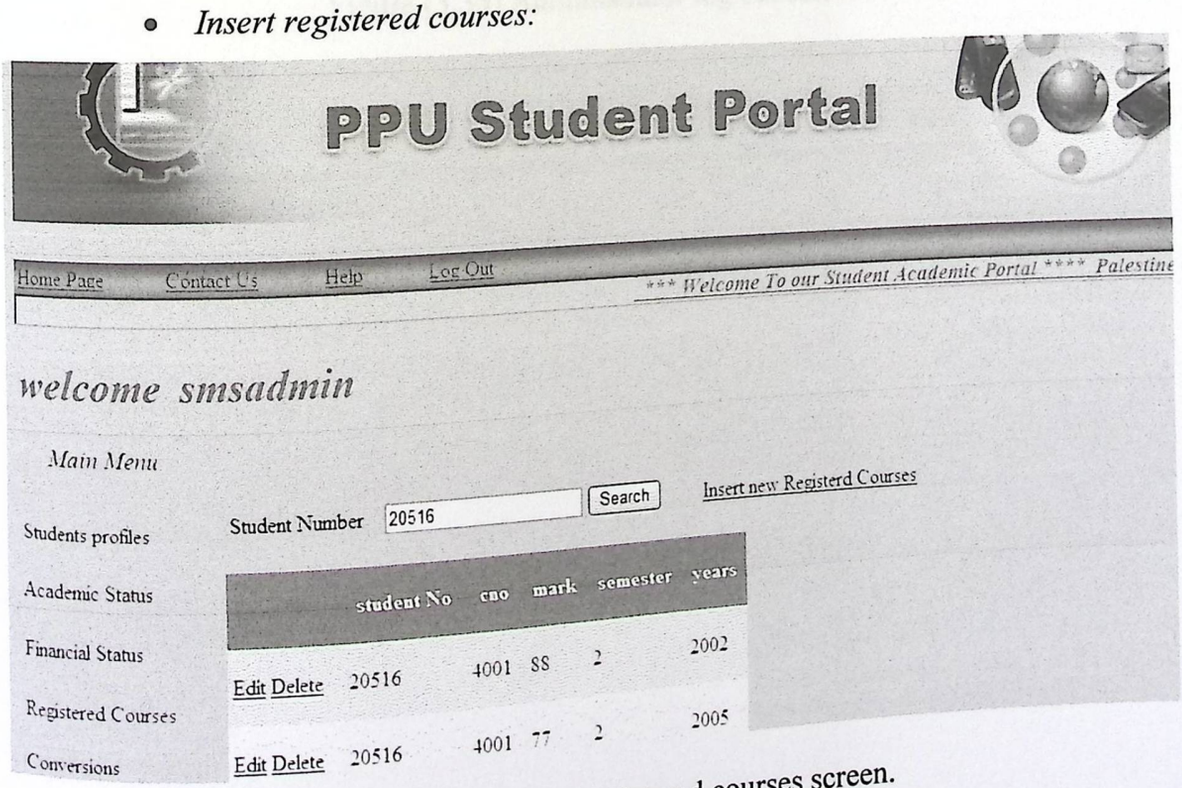


Figure (3,33) administrator change password screen

- *Insert registered courses:*



. Figure (3,34) Insert registered courses screen.

- *Log out:*

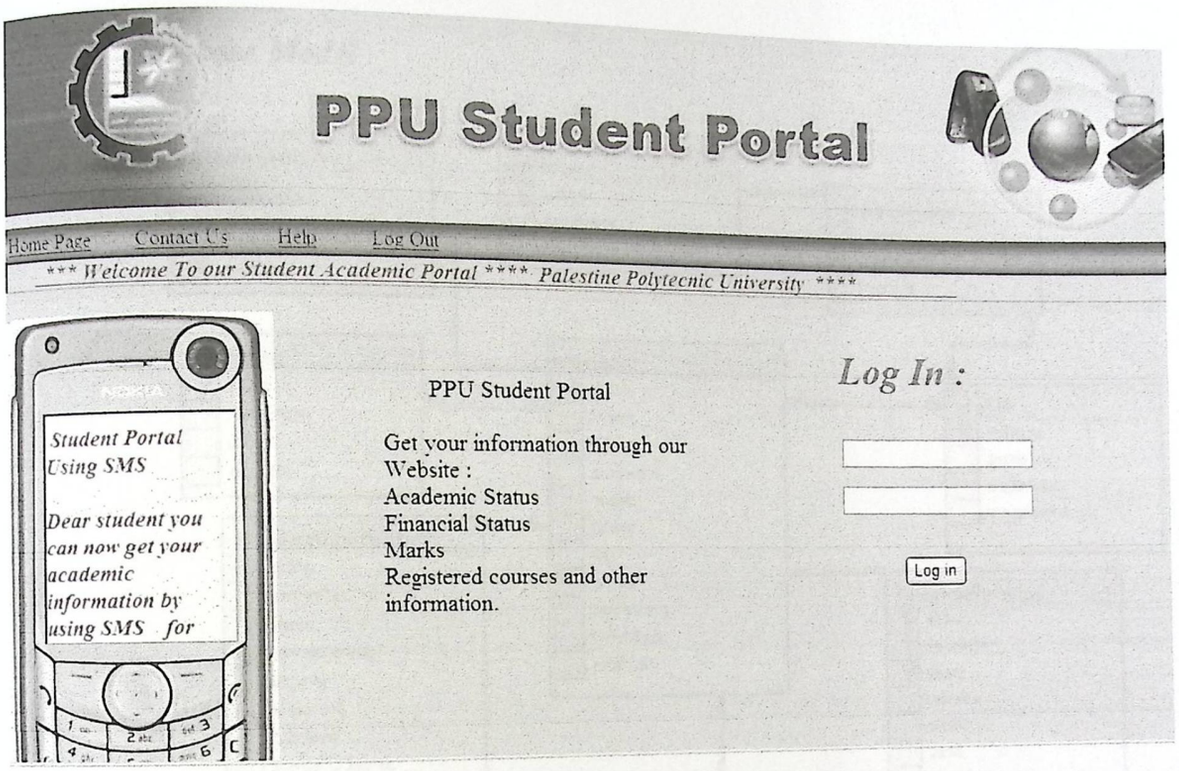


Figure (3,35) Administrator log out screen.

3.4 Database design

3.4.1 Database Model

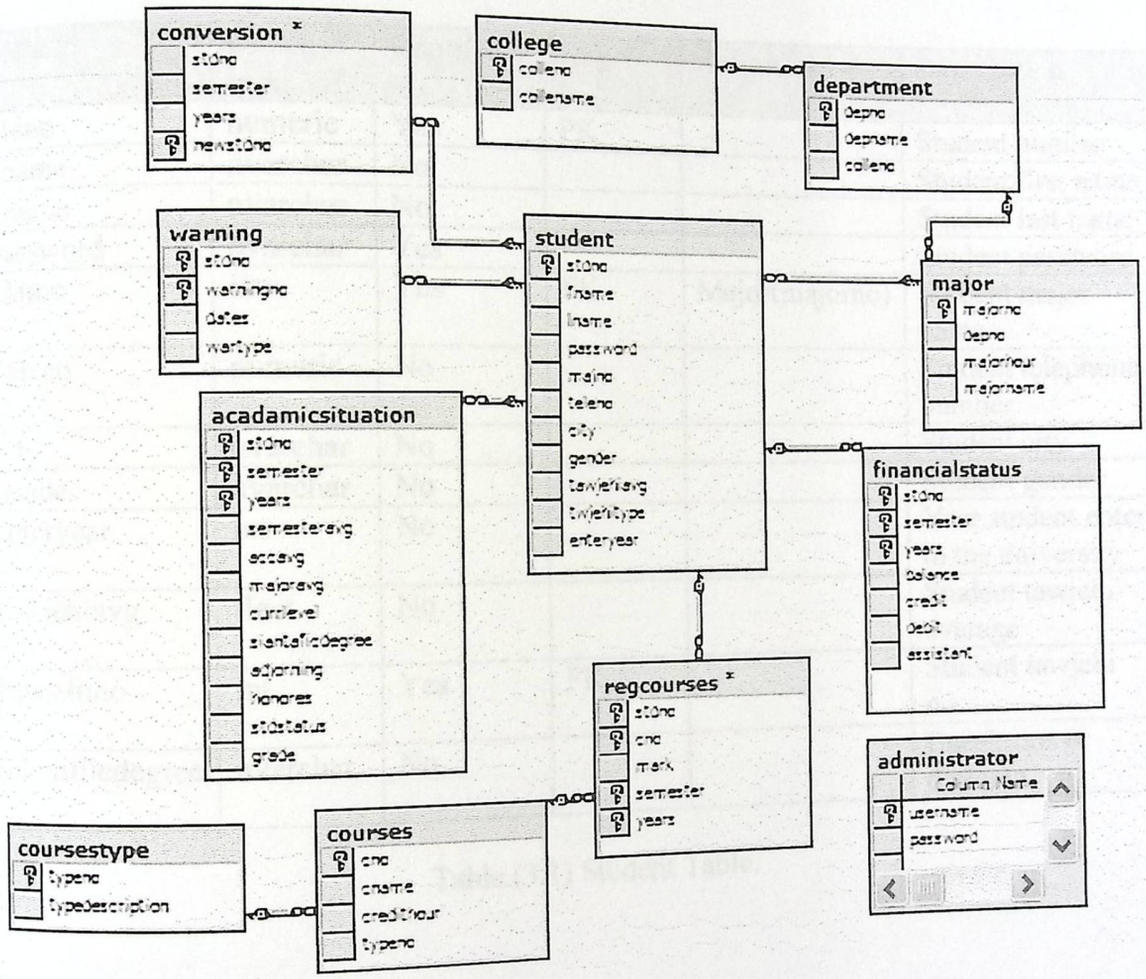


Figure (3,36) Database Model

3.4.2 database Tables

1. Student :

Field	Data type	Required	Key	References	Description
Stdno	numeric	Yes	PK		Student number
Fname	nvarchar	No			Student first name
Lname	nvarchar	No			Student last name
Password	nvarchar	Yes			Student password
Majno	int	Yes	FK	Major(majorno)	Student major name
Teleno	numeric	No			Student telephone number
City	nvarchar	No			Student city
Gender	nvarchar	No			Student gender
Enteryear	int	No			Year student enter to the university
Tawjehiavg	float	No			Student tawjehi average
Tawjehino	int	Yes	FK		Student tawjehi no
Scientificdegree	nvarchar	No			Bacalorios or diploma

Table (3.1) Student Table.

2. Academic Status :

Field	Data type	Required	Key	References	Description
stdno	numeric	Yes	PK,FK	Student(stdno)	Student number
semester	nvarchar	Yes	PK		First ,second or summer
years	numeric	Yes	PK		Academic year
Accavg	float	No			Student accumulative average
Majoravg	float	No			Student major average
cuurlevel	int	No			Student current level
adjourning	bit	No			Adjourning or not
honors	bit	No			Have honors or not
stdstatus	nvarchar	No			Student graduate or not
grade	nvarchar	No			Excellent ,very good or good

Table (3.2) Academic Status Table.

3. Financial Status:

Field	Data type	Required	Key	References	Description
stdno	numeric	Yes	PK,FK	Student(stdno)	Student number
semester	nvarchar	Yes	PK		First ,second or summer
years	int	Yes	PK		Academic year
balance	float	No			Balance value
credit	float	No			Credit value
debit	float	No			Debit value
assistance	float	No			Assistance value

Table (3.3) Financial Status Table.

4. **College :**

Field	Data type	Required	Key	References	Description
colleno	int	Yes	PK		College number
collename	nvarchar	No			Student College name

Table (3.4) College Table.

5. **Department :**

Field	Data type	Required	Key	References	Description
depno	int	Yes	PK		Department number
depname	nvarchar	No			Department name
Colleno	int	Yes	FK	College(colleno)	College number

Table (3.5) Department Table.

6. **Courses :**

Field	Data type	Required	Key	References	Description
Cno	int	Yes	PK		Course number
Cname	nvarchar	No			Course name
credithours	int	No			Course hour
typeno	int	Yes	FK	Coursestype(typeno)	Course type number

Table (3.6) Courses Table.

7. Major:

Field	Data type	Required	Key	References	Description
majorno	int	Yes	PK		Major number
depno	int	Yes	FK	Department(depno)	Department number
majorhour	int	No			Major total hours
majorname	nvarchar	No			Major name

Table (3.7) Major Table.

8. Administrator :

Field	Data type	Required	Key	References	Description
username	nvarchar	Yes	PK		Administrator username
password	nvarchar	Yes			Administrator password

Table (3.8) Administrator Table.

9. Warning:

Field	Data type	Required	Key	References	Description
stdno	int	Yes	PK		Student number
warningno	int	Yes	PK		Warning number
dates	datetime	Yes			The date
wartype	nvarchar	No			Warning type

Table (3.9) Warning Table.

10. Course type :

Field	Data type	Required	Key	References	Description
typeno	int	Yes	PK		Course type number
typedescription	nvarchar	No			Department name

Table (3.10) Course type Table.

11. Registered courses :

Field	Data type	Required	Key	References	Description
Cno	int	Yes	PK,FK	Courses(cno)	Course number
stdno	nvarchar	Yes	PK,FK	Student(stdno)	Student number
mark	float	No			Course mark
semester	nvarchar	Yes	PK		First ,second or summer
years	int	Yes	PK		Academic year

Table (3.11) Registered courses Table.

12. Conversion:

Field	Data type	Required	Key	References	Description
stdno	numeric	yes	FK	Student(stdno)	Student number
semester	nvarchar	no			First ,second or summer
years	numeric				Academic year
newstdno	numeric	yes	PK		New students number
newmajorno	int	yes			New major number

Table (3.12) Conversion Table.

13. Tawjehi:

Field	Data type	Required	Key	References	Description
Tawjehino	int	Yes	PK		Course type number
Tawjehitype	nvarchar	No			Department name

Table (3.13) Tawjehi Table.

3.5 *Test plan :*

Test plan include many steps that used to test the system to ensure that the system perform its operation perfectly, now we will describe the testing steps.

Testing steps:

1. **Unit testing:**

   Here we test each unit separately in the system to ensure that each One meet its requirement in the system and operate correctly.

2. **Sub-system testing:**

   Subsystem testing that depends on testing the related system components, so it can be tested individually.

3. **Integration Testing:**

   The integration testing which depends on testing all components together as a whole system to ensure that the system meets it is requirements.

4. **System testing:**

   System testing is one of the most important stages at all software projects to ensure that it functions and is working as properly as expected, and that it avoids any problem.

3.6 *Programming language and coding*

Here we describe advantages of Visual Studio.NET 2005 and why our selection was on it.

- Support dealing with serial ports so we can use AT command to send and receive SMS in it.
- Security: have high security in transferred data, and support validation that ensure the user's inputs before doing any operation.

Chapter Four

Code and Implementation

- *Introduction*
- *Establishment of development environment*
- *Database implementation*
- *supporting software*
- *operating the system*
- *coding*
- *development process implementation*

4.1 Introduction:

In this chapter we will describe the coding and implementation processes, and the environment that we use in this system

The development of our system needs a set of software and hardware to meet the predefined requirement found in a platform configured to be suitable for the deployment process.

This system is web site and SMS application that depends on a number of software needs to be installed to accomplish this system,

These programs are Microsoft SQL server 2005 visual basic 2005, IIS, Microsoft windows XP, also we have to use some programs to test the connection between the mobile and computer such as nokia pc suite and hyper terminal, and also we use the Macromedia Flash MX and Adobe Photoshop 8.0 to design the system interface.

This chapter describes these 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) Two PC Pentium 4:
     a) 2.78 GHz speed.
     b) 504 MB RAM.
     c) 20 GB H.D.D
     d) Monitor, mouse, keyboard.
  2) One printer.
  3) Four Flash memories 1G.
  4) Connection cable USB port.
  5) Nokia mobile 6100.
  6) Nokia mobile.

- ***Software environment:***

For the system development we use the following software:

1) Microsoft windows XP professional with IIS web server.

We use this operating system because our system is built on Microsoft Visual Studio .NET 2005 programming and this language depends on this type of operating system.

when we want to install IIS (internet information services) , we follow these steps:

- Open control panel.
- Select add /remove programs icon.
- Choose add/remove windows component icon.
- Check on IIS option.
- Click on details button.
- Check on options.
- Insert windows XP professional CD.
- Click next.
- After installed click finished.

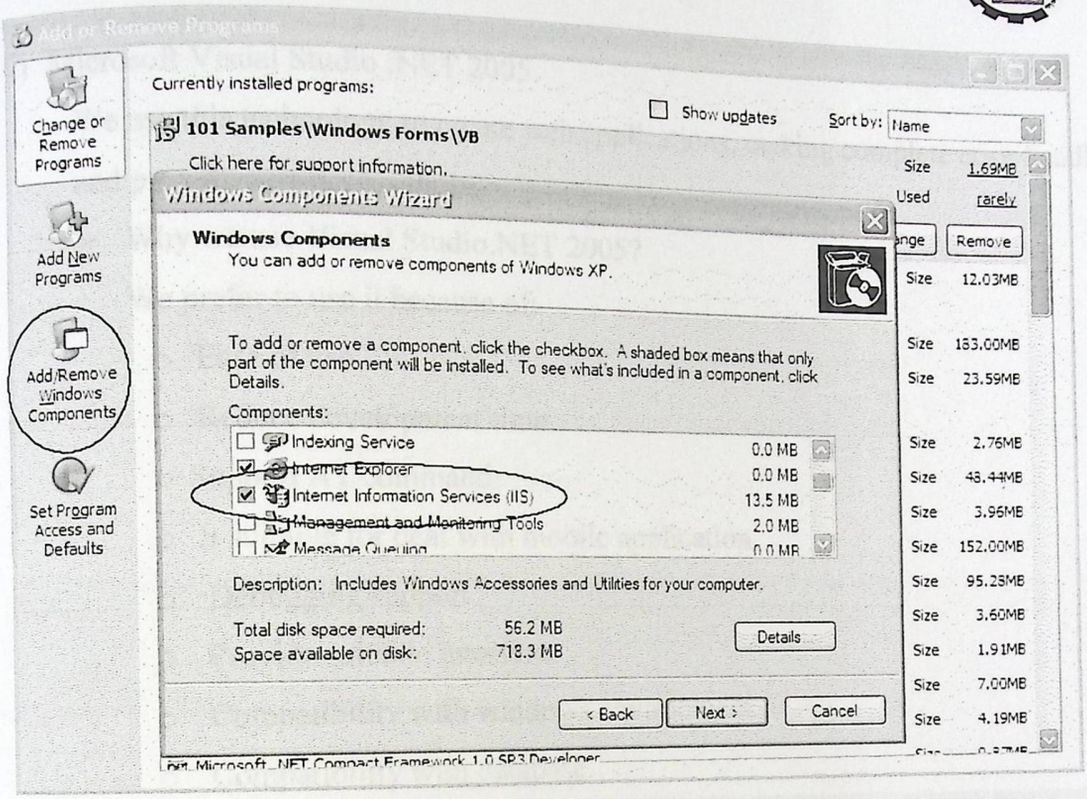


Figure (4.1) IIS (Internet Information Services)

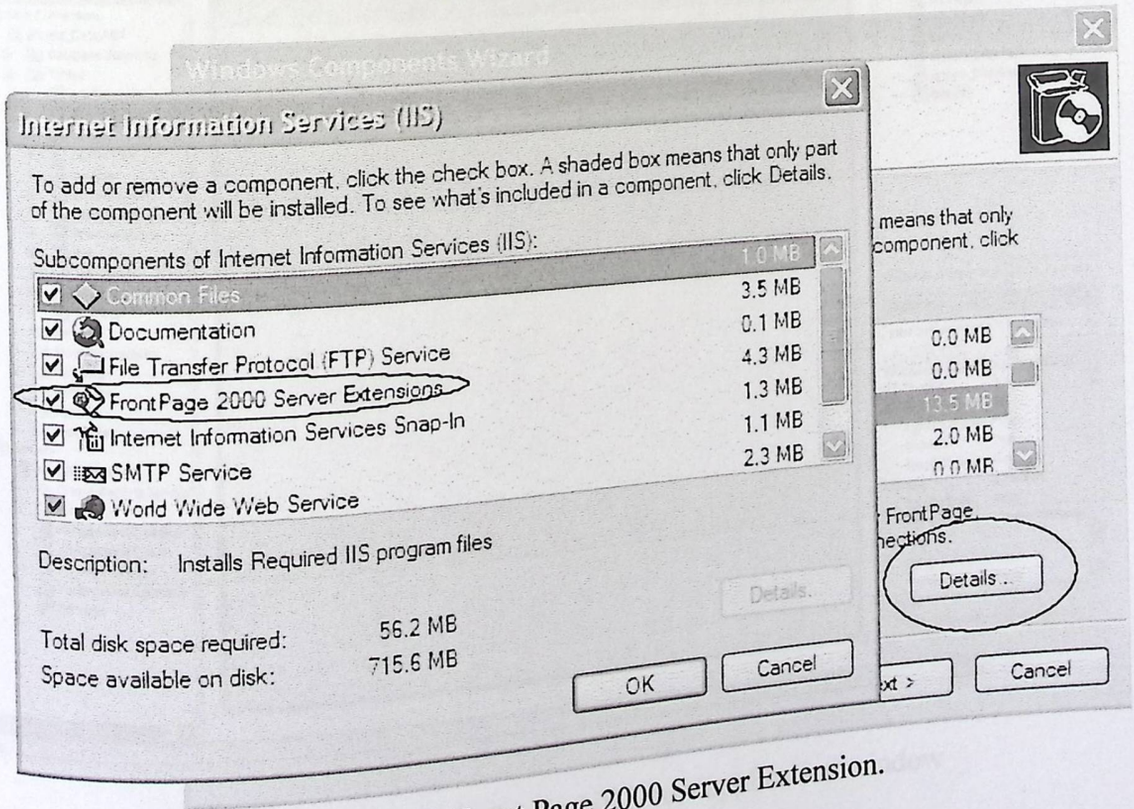


Figure (4.2) Front Page 2000 Server Extension.

2) Microsoft Visual Studio .NET 2005.

We use this technology to create web applications, making complete error handling, and provide data access tools.

- Why we use Visual Studio.NET 2005?

  We prefer to use it because of:

  - Ease of use.
  - Reduce development time.
  - Support AT command.
  - It suitable for deal with mobile application.
  - Debugging support.
  - Provide suitable interface.
  - Compatibility with windows environment.
  - Compatibility with database.

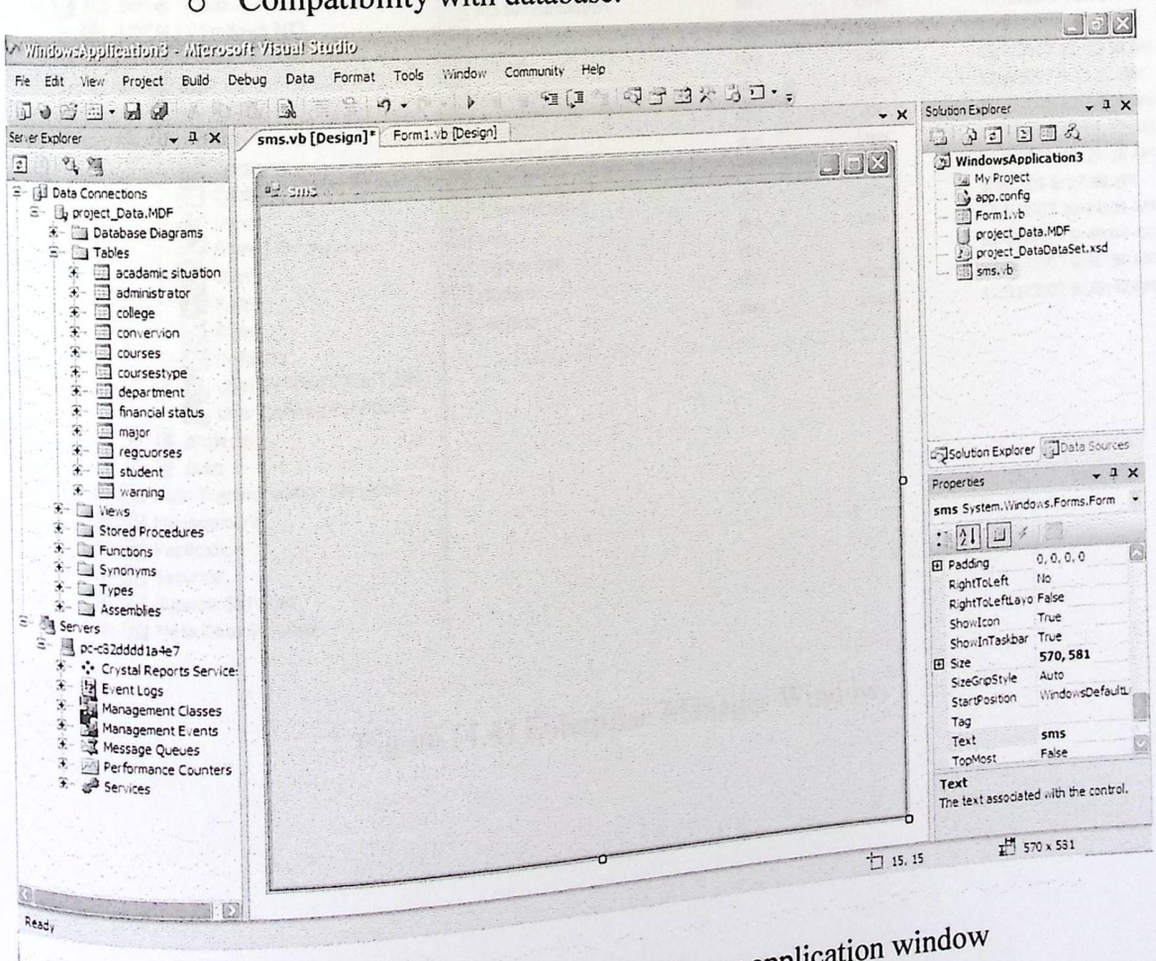


Figure (4.3) Visual Studio.NET new application window

3) Microsoft SQL server 2000.

It a good system to creating, accessing, managing the database for our system.

- Why we use SQL server 2000?
  - Ease of use.
  - Compatibility with Visual Studio .NET 2005.
  - High security.
  - Ease of make a connection, query, and retrieve data.
  - Support stored procedures.

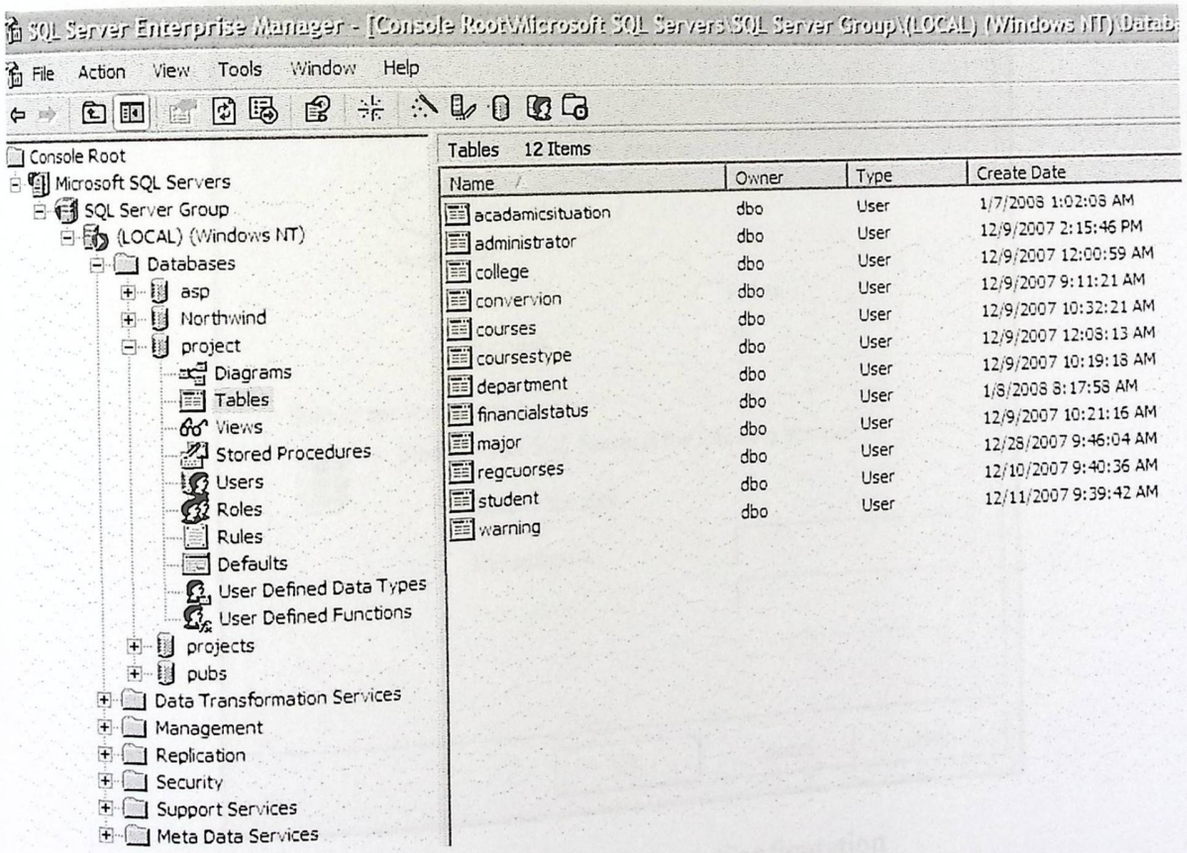


Figure (4.4) Enterprise Manager Window.

➢ Configuration Microsoft SQL server 2000:

Configure SQL server 2000 to the windows only used when connecting application to SQL server 2000 DBMS (database management system) which does not need username or password to be transferred back.

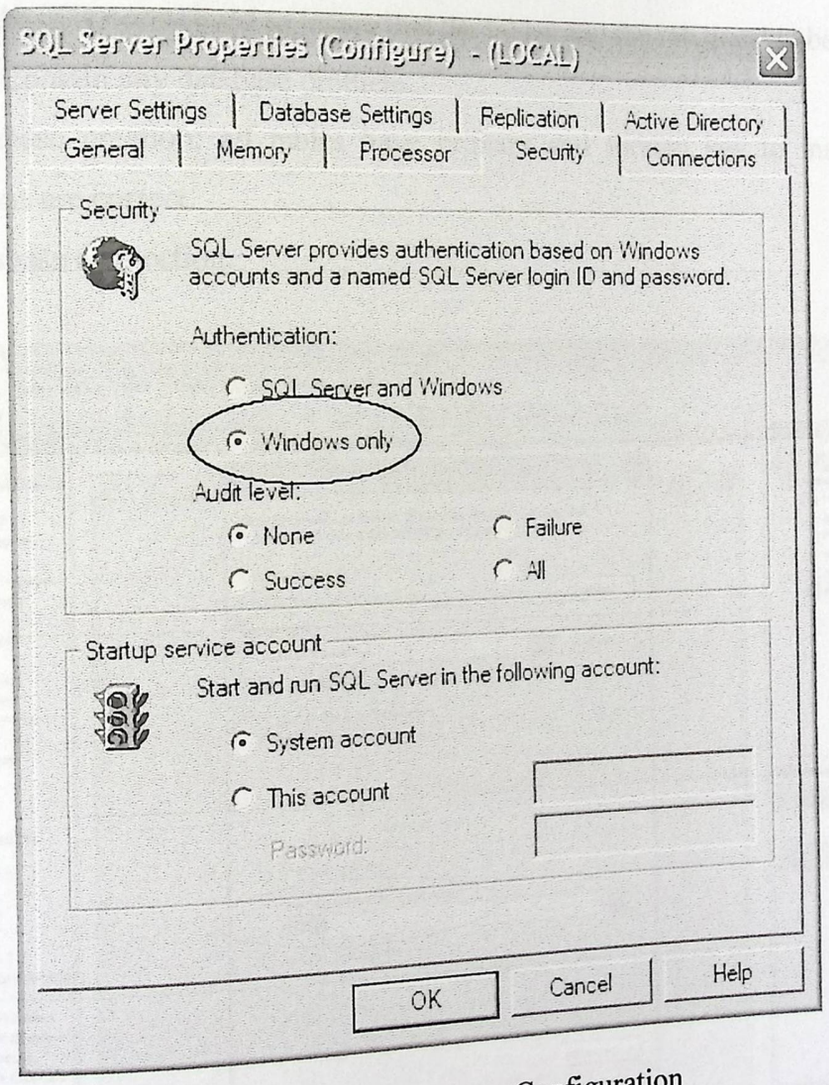


Figure (4.5) SQL Server Configuration

4.3 Database implementation:

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

a) SQL server 2000 database name: project.

b) Database is normalized to insure that the tables and the relationship between them do not contain any database problem.

c) Database creation: all tables have primary and foreign key to insure that the relations are correct.

d) Database connection.

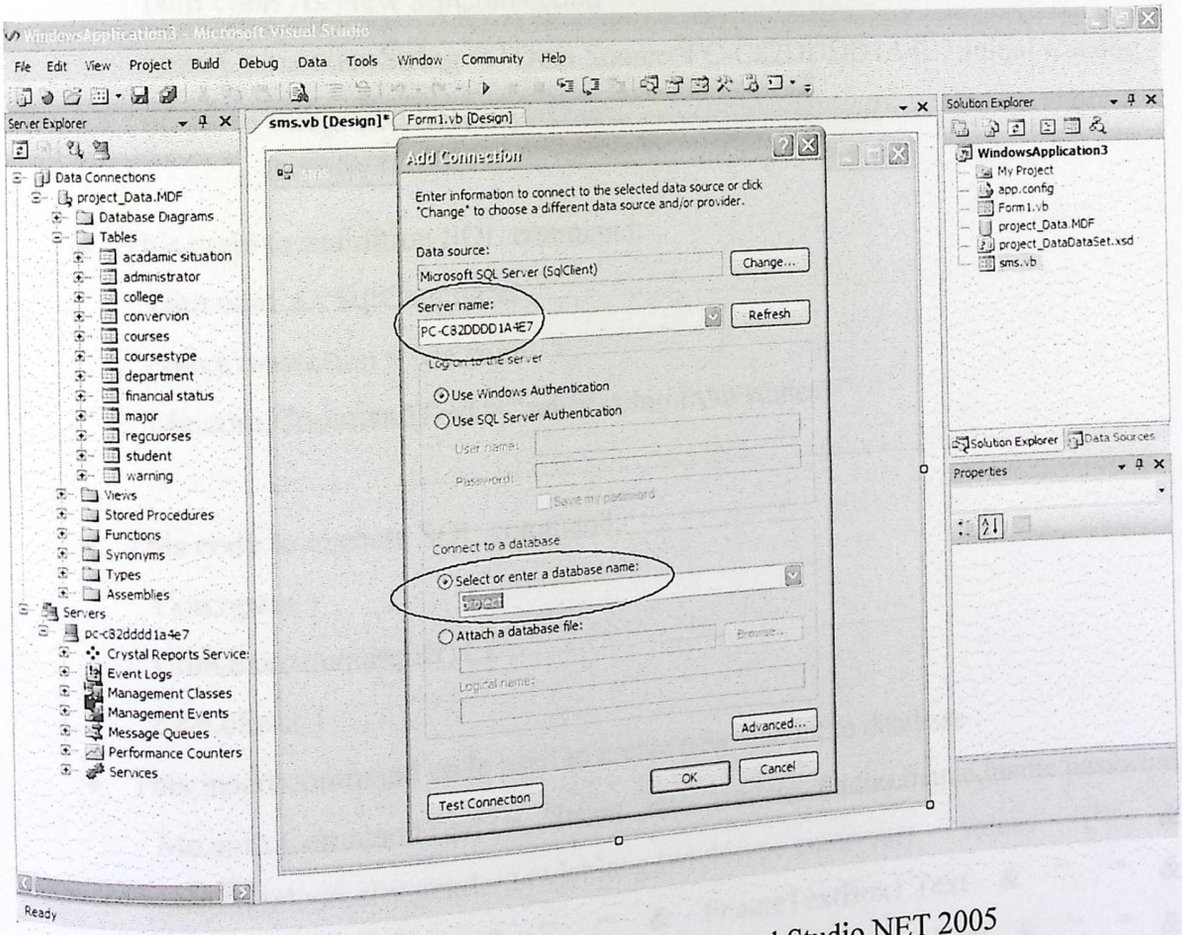


Figure (4.6) data connection in Visual Studio NET 2005

4.4 *Coding:*

In our system we use the Visual Studio.NET 2005 programming language as a programming environment, which provide a good way for programming the functionality, and save time and effort for design the graphical user interface (GUI), because it provide a suitable component (button, label, textbox and table) and support AT command.

Codes writes behind Visual Studio.NET 2005 pages:

- This code to create database connection

```
Dim conn As New SqlConnection
conn.ConnectionString = "Data Source=PC-C82DDDD1A4E7;Initial Catalog =
project ; Integrated Security=True"
```

- This code to definition SQL command:

```
Dim com As SqlCommand
com.Connection = Me.conn
Me.com.CommandText = "select stdno from student "
```

- This code to execute SQL command:

```
Con.open( )
Com.executenonquery( )
Con.close( )
```

- This insert command code used to create new student to database :

```
Me.com.CommandText = "Insert into student (stdno,fname,lname,password
,majno,teleno,city,gender,tawjehiavg,twjehitype,enteryear) values ('" &
StdnoTextBox1.Text & "' ,'" & FnameTextBox1.Text & "', '" &
LnameTextBox1.Text & "' ,'" & PasswordTextBox1.Text & "', '" &
MajnoTextBox.Text & "' ,'" & TelenoTextBox.Text & "','" &
CityTextBox1.Text & "','" & GenderTextBox1.Text & "','" &
```

TawjehiavgTextBox1.Text & "' ,'" & TwjehitypeTextBox1.Text & "','" & EnteryearTextBox1.Text & "')"

- This delete command code used to delete student from database:

  Me.com.CommandText = "delete from student where stdno= '" & StdnoTextBox1.Text & "' "

- This update command code used to update student information at database:

  Me.com.CommandText = "update student set fname= '" & FnameTextBox1.Text & "',lname='" & LnameTextBox1.Text & "',password='" & PasswordTextBox1.Text & "',majno='" & MajnoTextBox.Text & "',teleno='" & TelenoTextBox.Text & "',city='" & CityTextBox1.Text & "',gender='" & GenderTextBox1.Text & "',tawjehiavg='" & TawjehiavgTextBox1.Text & "',twjehitype='" & TwjehitypeTextBox1.Text & "',enteryear='" & EnteryearTextBox1.Text & "' where stdno ='" & StdnoTextBox1.Text & "' "

- This code is used to send SMS from computer to mobile:

  atcCommand = "AT+CMGS=" + cescQuote + sphnPhoneNo + cescQuote + vbCrLf + PasswordTextBox.Text + Char.ConvertFromUtf32(26)

- This code used to read SMS from mobile:

  Dim str As String

  returnStr = "at+cmgr=" + str + vbCrLf

4.5 *supporting software:*

Many of software programs are used to support the system:

1. Photoshop that used to improve the appearance of the application software.

2. Nokia pc suite for Nokia 6100 that is used to make connection between the intermediate mobile and the computer.

3. Microsoft Visio that used to draw the flowchart, context diagram and data flow diagram.
4. HyperTerminal where is used to test if the intermediate mobile support AT command or not, and to enable the intermediate mobile to operate with text mode.
5. Macromedia flash.

4.6 operating the system:

There are many steps should be taken before the system operates:

- Setting up the .NET framework.
- Building the system database.
- Creating the database connection.

To execute the system from the development environment we follow these steps:

- Go to start menu and select Microsoft Visual Studio.NET 2005.
- Then will appear two choices, open link of existing project or new link to create new project. Select open existing project
- Then select project that name is "graduate".

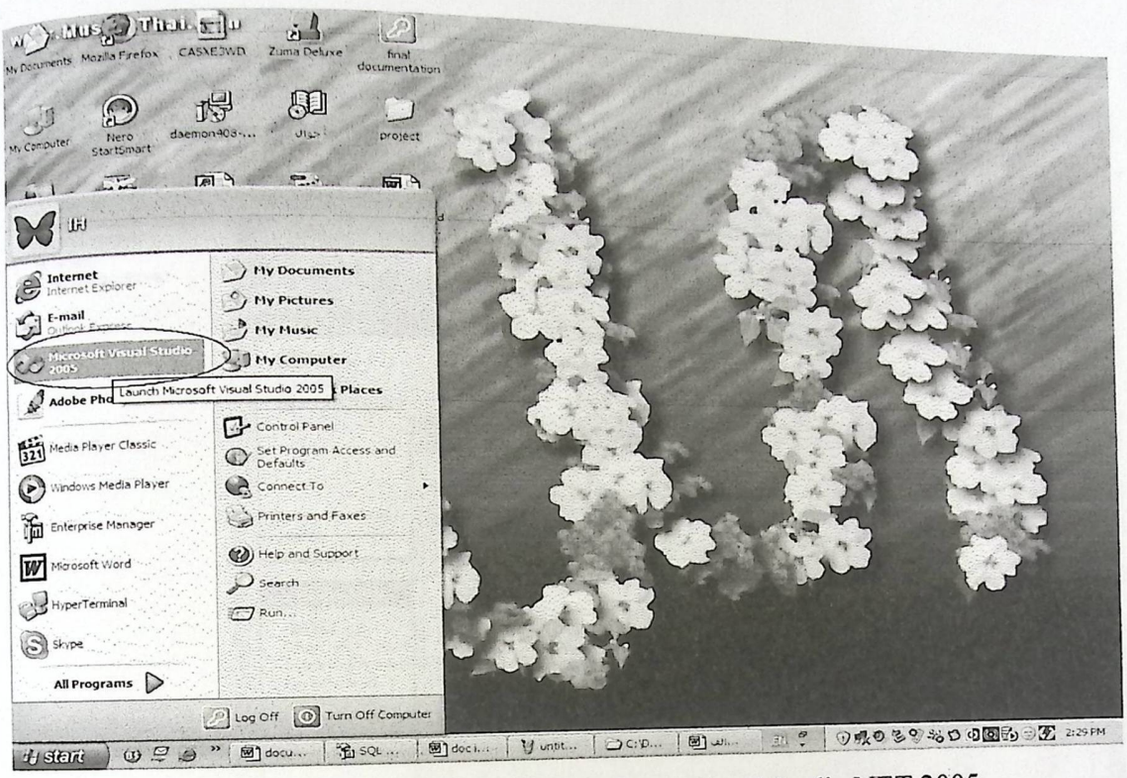


Figure (4.7) select Microsoft Visual Studio.NET 2005

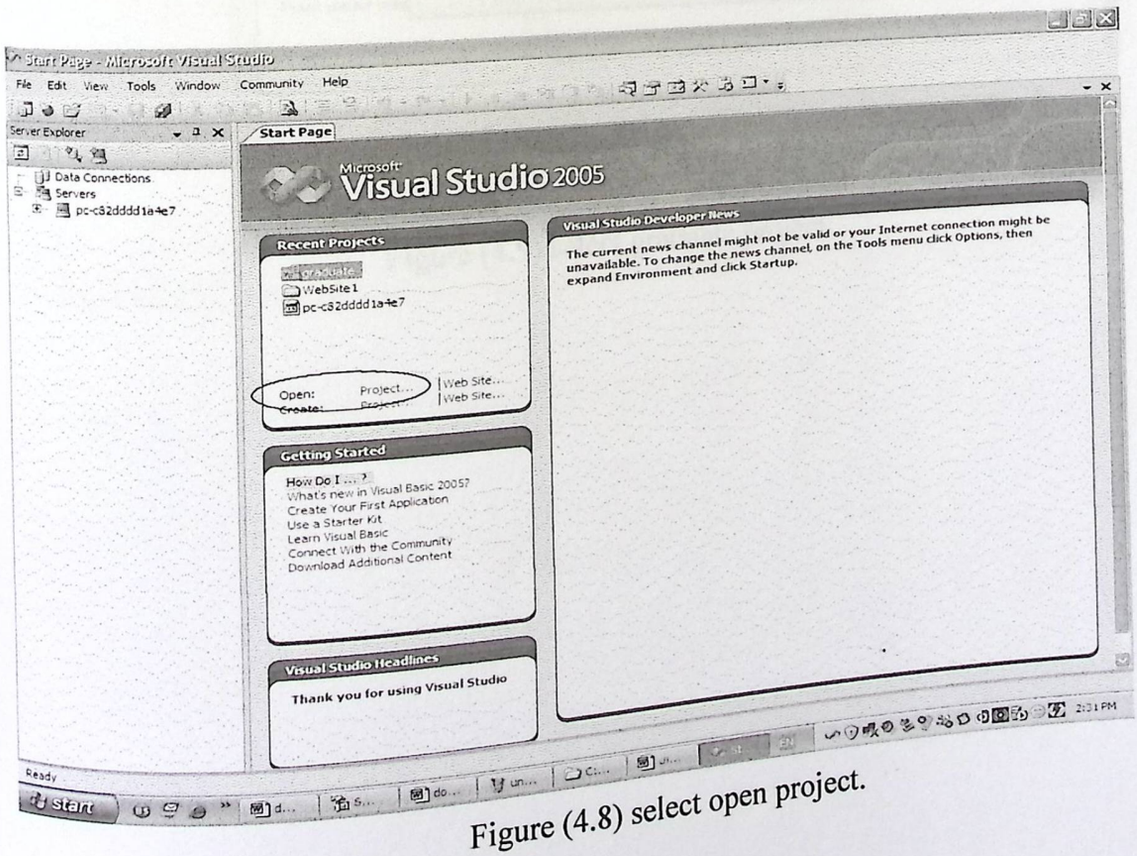


Figure (4.8) select open project.

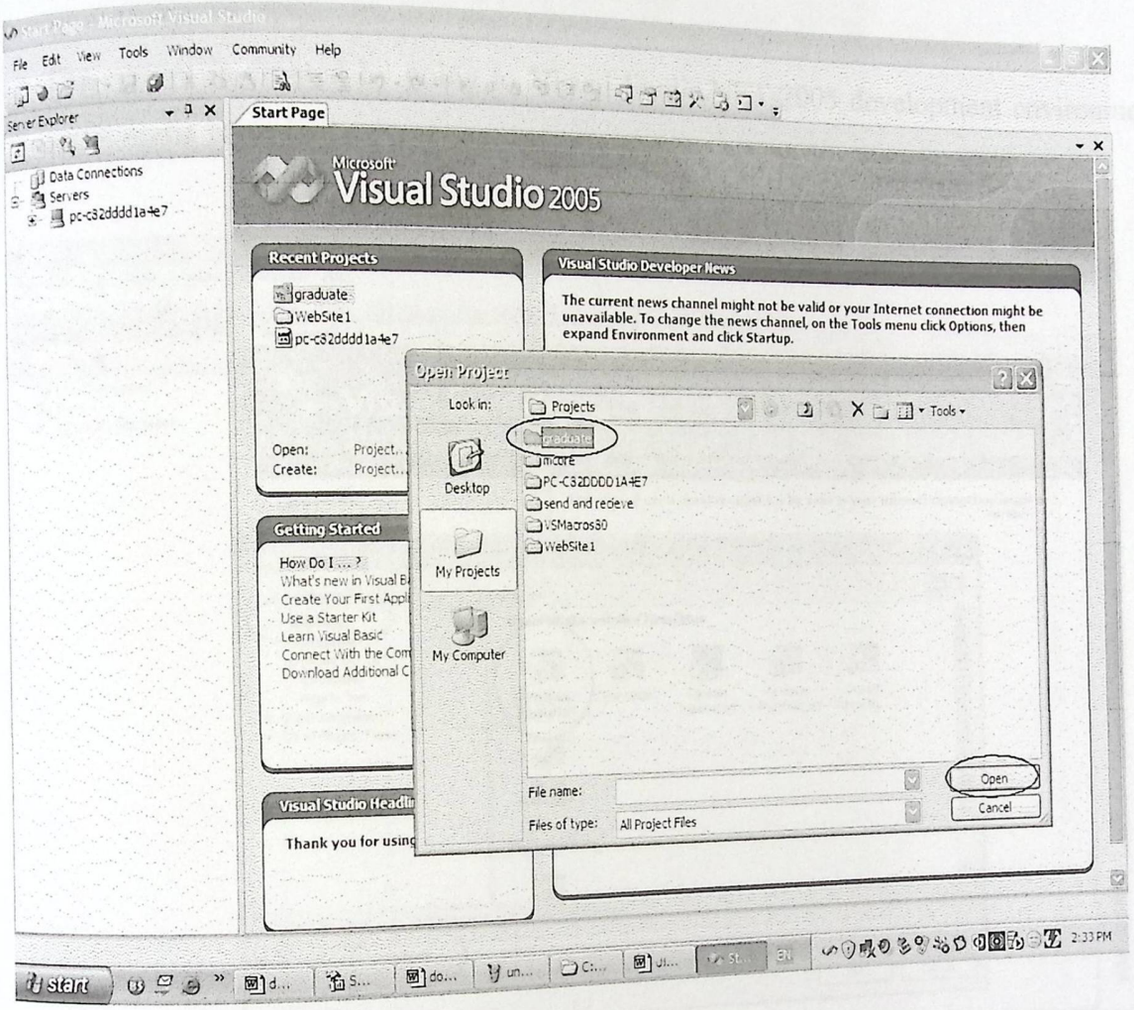


Figure (4.9) select graduate project.

4.7 development process implementation:

The system is created by going to Visual Studio.NET 2005 development environment and it name is project.

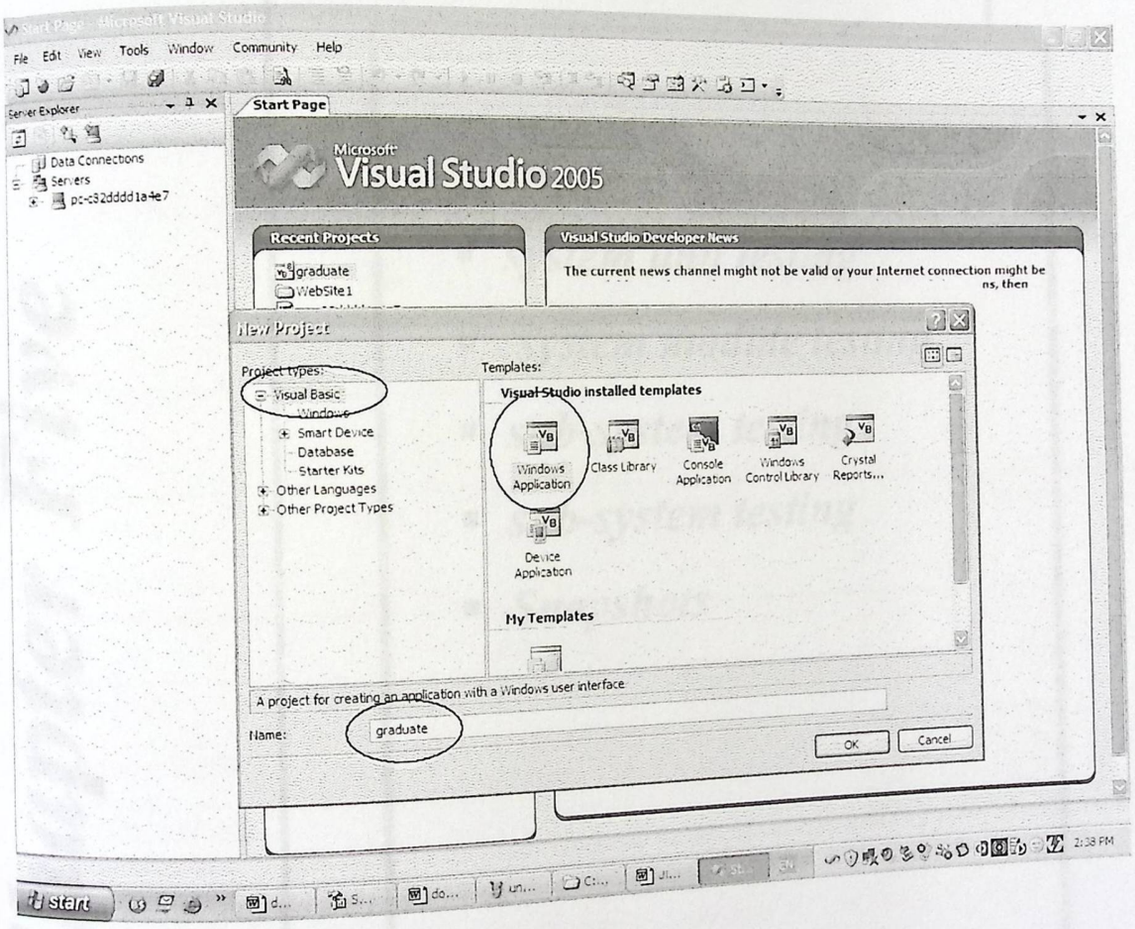


Figure (4.10) create new project.

System Testing

Chapter Five

- *Introduction*
- *system unit testing*
- *system module testing*
- *sub-system testing*
- *sub-system testing*
- *Snapshots*

5.1 *Introduction :*

Most important step we must do before deliver the system is testing the system ,to ensure it worked as we exactly expected , and to ensure that it meet all requirements that we specified before.

The testing process includes four levels:

- System unit testing.
- Module testing.
- Subsystems testing.
- Integration testing.

Testing process will take time as follow:

Time in week / Testing process	First week		Second week
unit testing			
Module testing			
Subsystems testing			
Integration testing			

Table (5.1) Testing schedule

5.2 *System unit testing :*

At this type of testing the system is divided into subsystems that each one of them will be tested separately to ensure that each one is meet its requirement.
In this type of testing we will use white and black box testing to test the administrator log in and student changing his password.

As we will see, first we will use valid inputs and another time we will use invalid inputs, and we will see the result at each time, as the below tables which explains the testing process for administrator log in and student changing his password:

Unit testing process	Username	password	Result	Actual result
Valid username and password	smsadmin	asma	Administrator page load	Administrator page load
Wrong expression for username or password	hghhh	Dfg&df//h	Invalid inputs format	Invalid inputs format
Invalid username or password	smsadmin	1234	Error message appear, invalid username or password	Error message appear, invalid username or password

Table (5.2) Administrator log in testing unit

Unit testing process	Old password	New password	Confirm new password	Expected result	Actual result
Valid old password, new password , and its confirm	2681985	111	111	Password updated successfully	Password updated successfully
Wrong expression for old password or new password and confirm it	hgfjhg	22&2	22&2	Invalid inputs format	Invalid inputs format
Invalid old password	sms	111	111	Error message appear" invalid old password"	Error message appear" invalid old password"
Wrong in password confirmation	2681985	111	112	Error message appear" please confirm password correctly"	Error message appear" please confirm password correctly"

Table (5.3) student change password testing unit

Unit testing process	Old password	New password	Confirm new password	Expected result	Actual result
Valid old password, new password , and its confirm	2681985	111	111	Password updated successfully	Password updated successfully
Wrong expression for old password or new password and confirm it	hgfjhg	22&2	22&2	Invalid inputs format	Invalid inputs format
Invalid old password	sms	111	111	Error message appear" invalid old password"	Error message appear" invalid old password"
Wrong in password confirmation	2681985	111	112	Error message appear" please confirm password correctly"	Error message appear" please confirm password correctly"

Table (5.3) student change password testing unit

5.3 Sub-system testing:

At this type of testing our system has tow subsystems as follow:

1. Administrator subsystem:

Here we tested the administrator functions that operate through the administrator page to ensure they meet their specification, and also we tested the interface how it appears and how the data layout.

2. Student subsystem :

We tested the student functions that operate through the student page to ensure they meet their specification, and also we tested the interface how it appear and how the data layout, also we tested the SMS form that the student should send to request his university information.

5.4 Integration Testing:

At this testing type we tested all modules as a whole system to ensure that the system is meet its requirement and operate as we expected.

5.5 Snapshots:

At this part we will take some snapshots from administrator and student forms.

5.6.1 Some of Administrator forms:

1. Not allowing log in with invalid inputs formats:
   - Snapshot for administrator log in with invalid input formats.

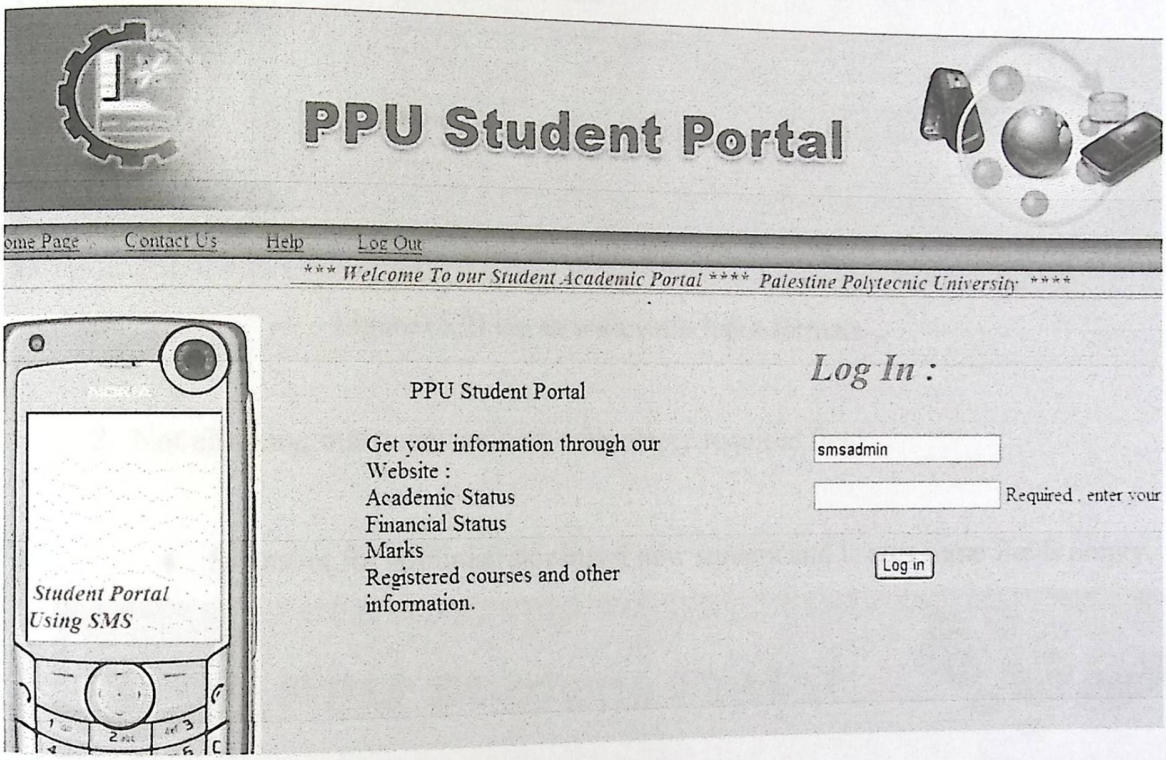


Figure (5.1) log in with invalid input formats.

- Snapshot for administrator log in with valid input formats.

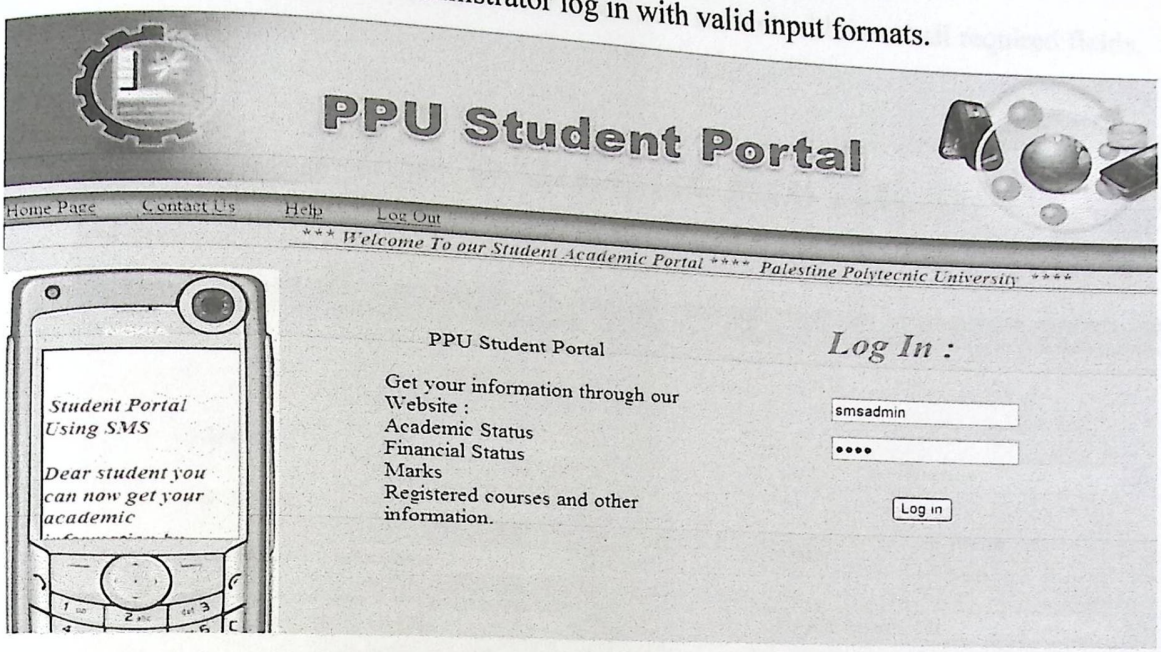


Figure (5.2) log in with valid input formats.

2. Not allowing insert new student with empty required field:

- Snapshot for administrator insert new student and leaves some fields empty.

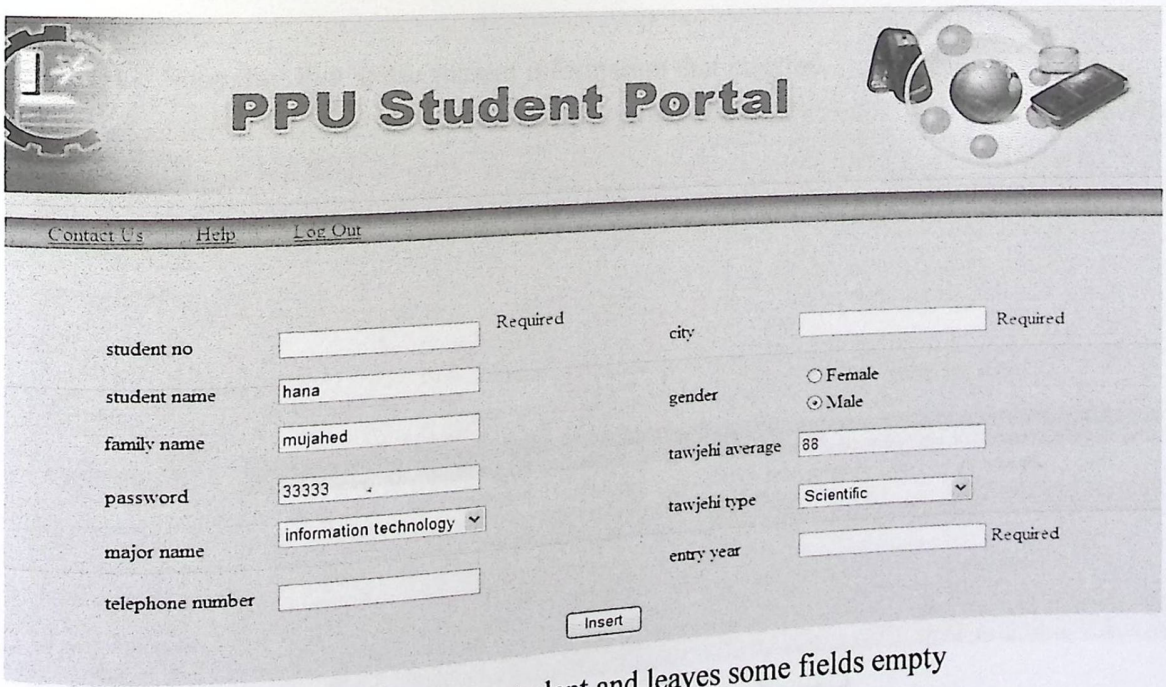


Figure (5.3) insert new student and leaves some fields empty

- Snapshot for administrator insert new student with insert all required fields.

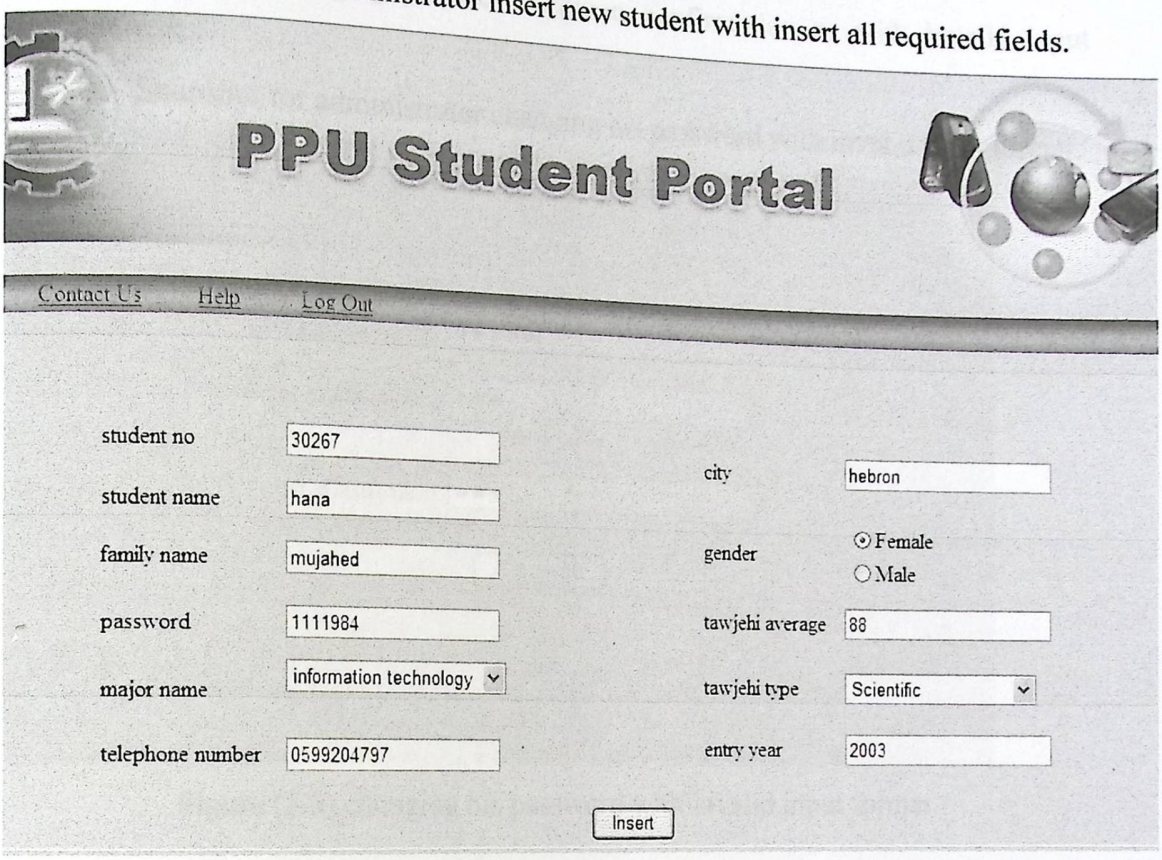


Figure (5.4) insert new student with insert all required fields

- Snapshot that show student information that inserted.

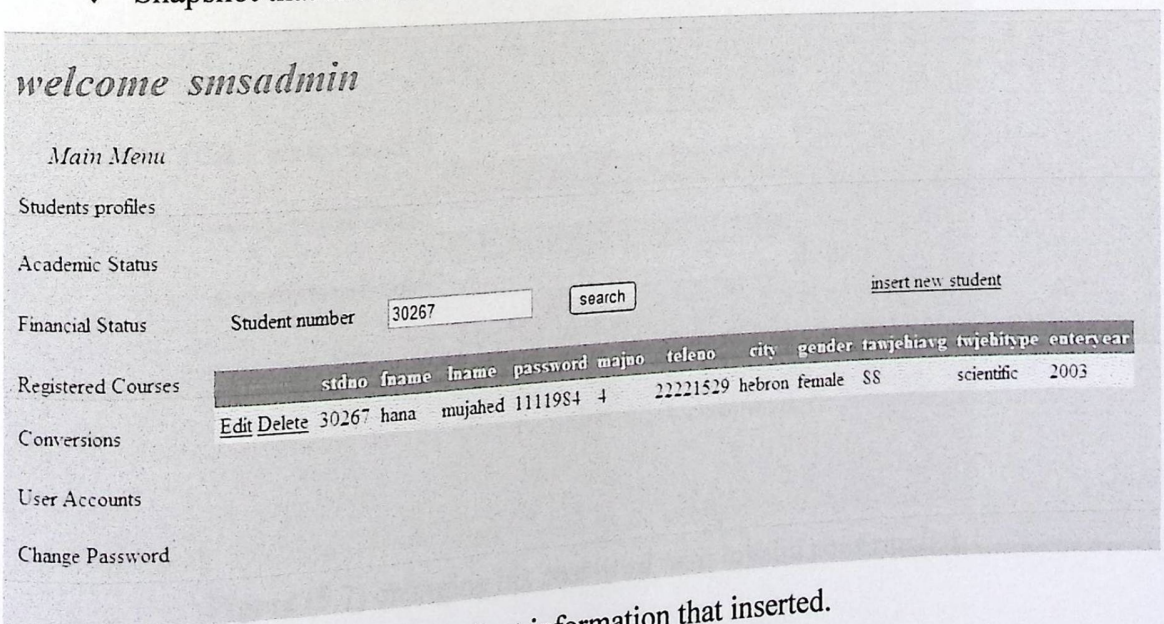


Figure (5.5) student information that inserted.

3. Not allowing changing password without confirmation or with invalid input format.

- Snapshot for administrator changing his password with invalid input format.

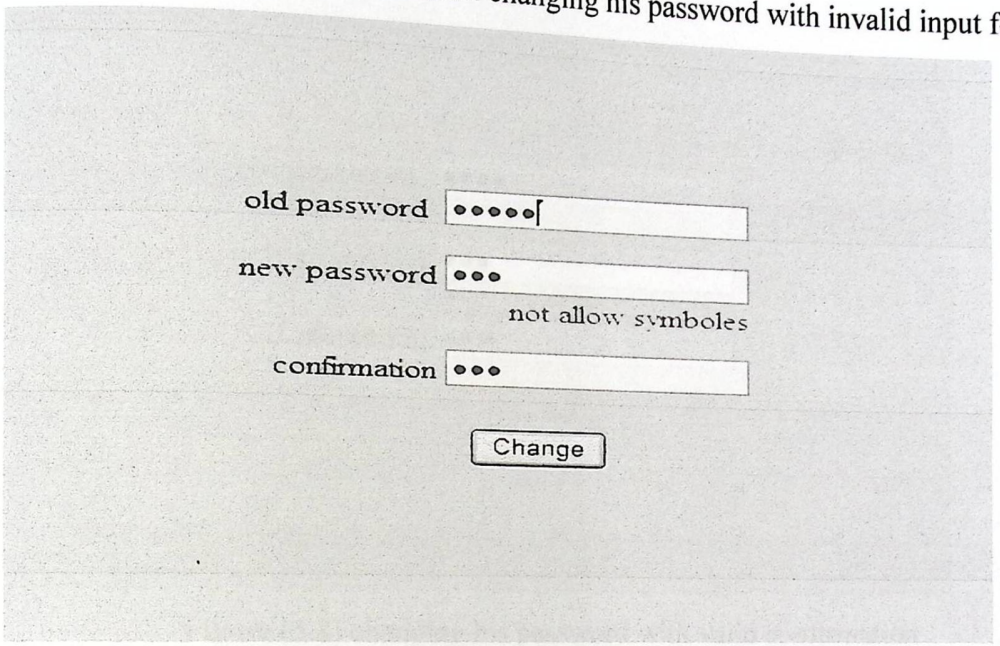


Figure (5.6) changing his password with invalid input format.

- Snapshot for administrator changing his password with invalid confirmation of new password.

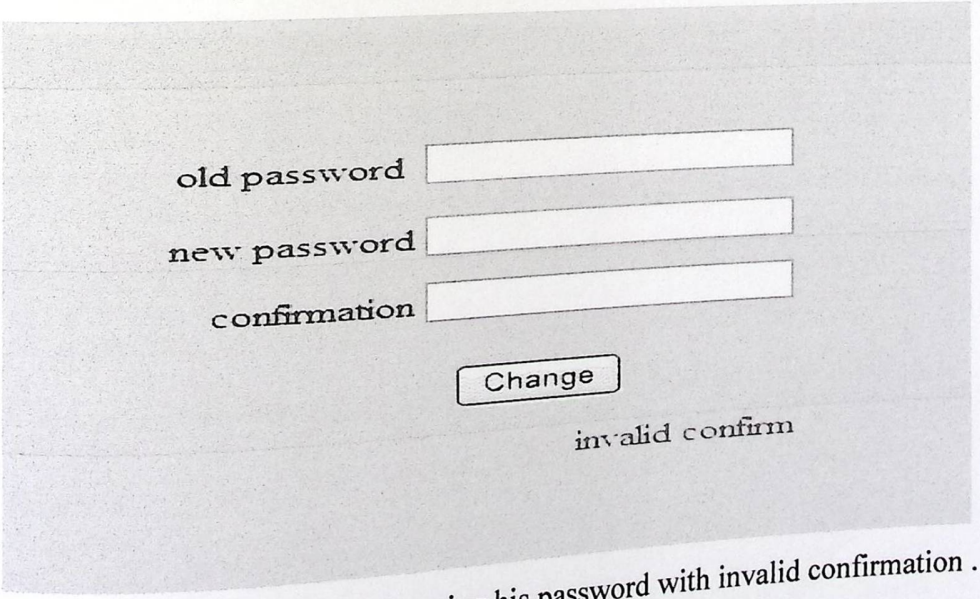


Figure (5.7) changing his password with invalid confirmation .

- Snapshot for administrator changing his password with valid confirmation of new password.

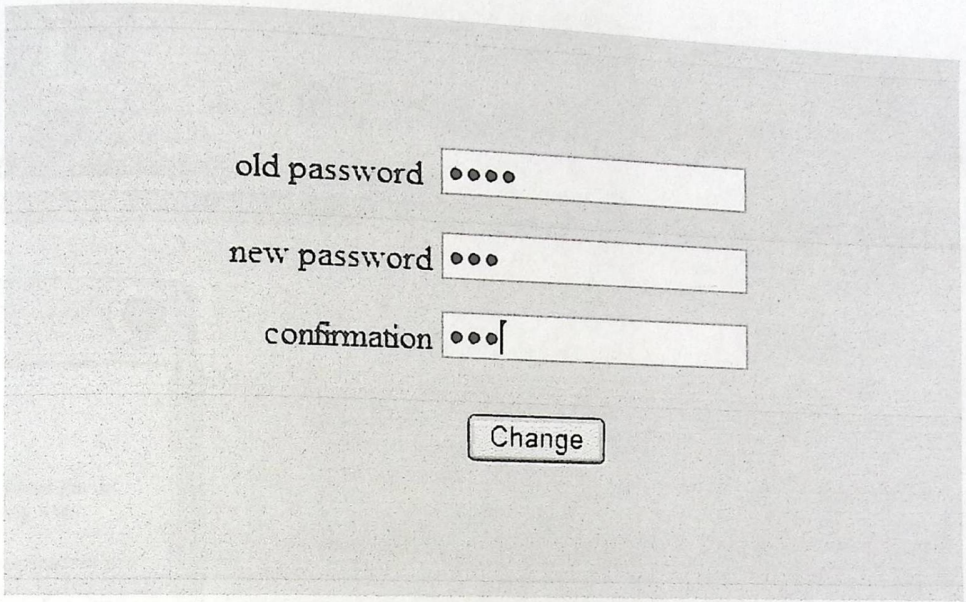


Figure (5.8) changing his password with valid confirmation .

5.6.2 *Some of Students web sites:*

1. Snapshot for student invalid log in.

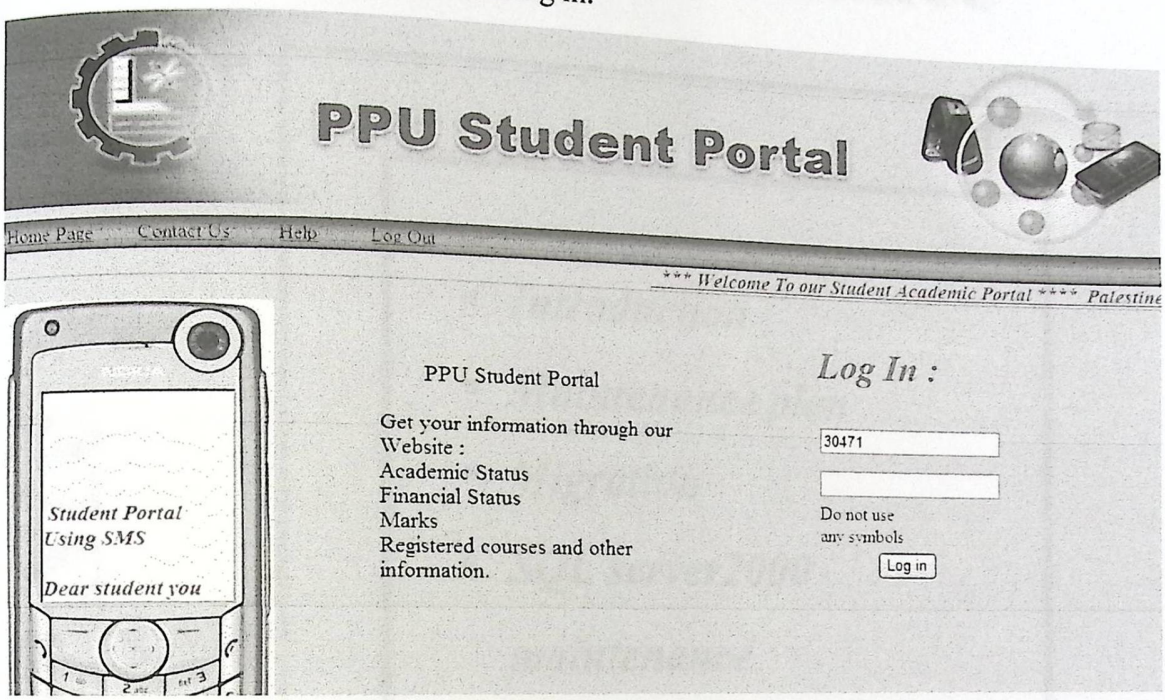


Figure (5.9) student log in with invalid inputs.

2. Snapshot for student valid log in.

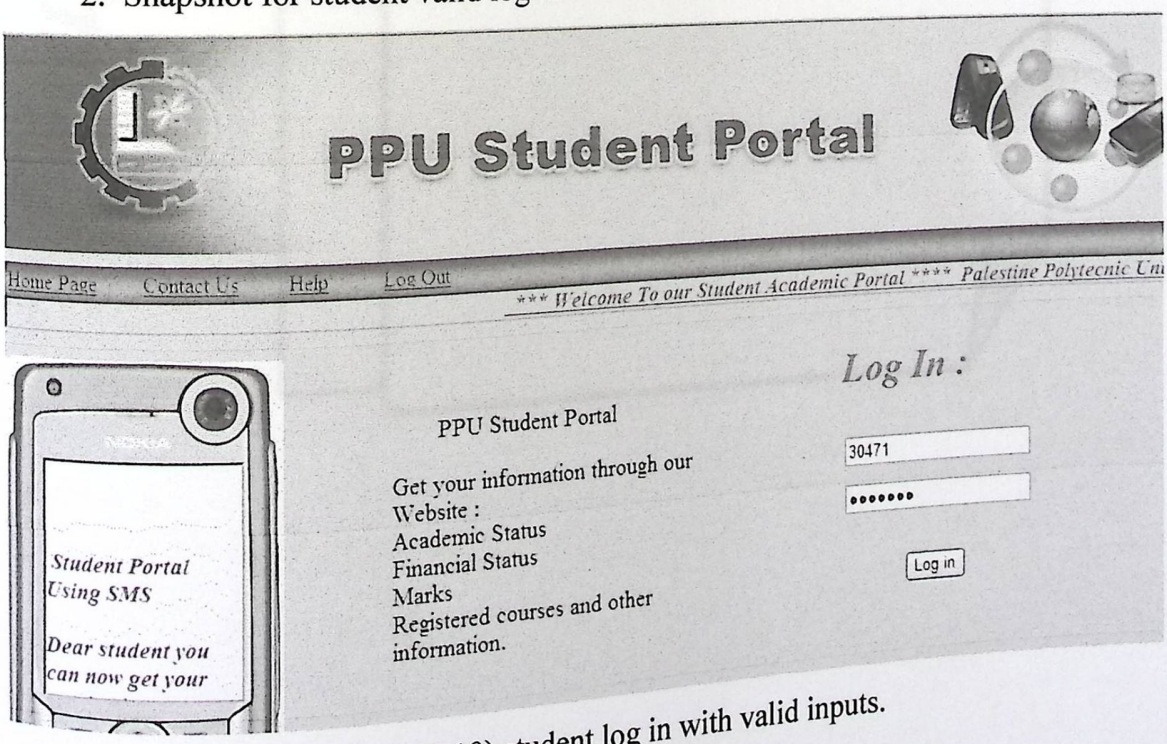


Figure (5.10) student log in with valid inputs.

System Maintenance

Chapter Six

- *Introduction*
- *Maintenance plan*
- *Migration*
- *SQL server2000 maintenance*
- *.NET framework maintenance*

6.1 *Introduction :*

At this chapter we will describe the process that the system administrator used to maintain the system after running it.

In this chapter we will describe:

- Maintenance plan.
- Migration.

Also we will talk about the system maintenance which includes:

- SQL server 2000 maintenance.
- The .NET framework maintenance.
- Mobile maintenance.
- Nokia pc suite maintenance.

6.1 *Introduction :*

At this chapter we will describe the process that the system administrator used to maintain the system after running it.

In this chapter we will describe:

- Maintenance plan.
- Migration.

Also we will talk about the system maintenance which includes:

- SQL server 2000 maintenance.
- The .NET framework maintenance.
- Mobile maintenance.
- Nokia pc suite maintenance.

6.2 *Maintenance plan:*

When we operating the system, there are always possibilities of system failure, occur problems and errors. So in this section we will show plan to maintains the system, which includes:

- Backup:

Database is the most important part in the system that includes tables, stored procedures, views and data of the system. So we should keep these data by take a backup for it periodically.

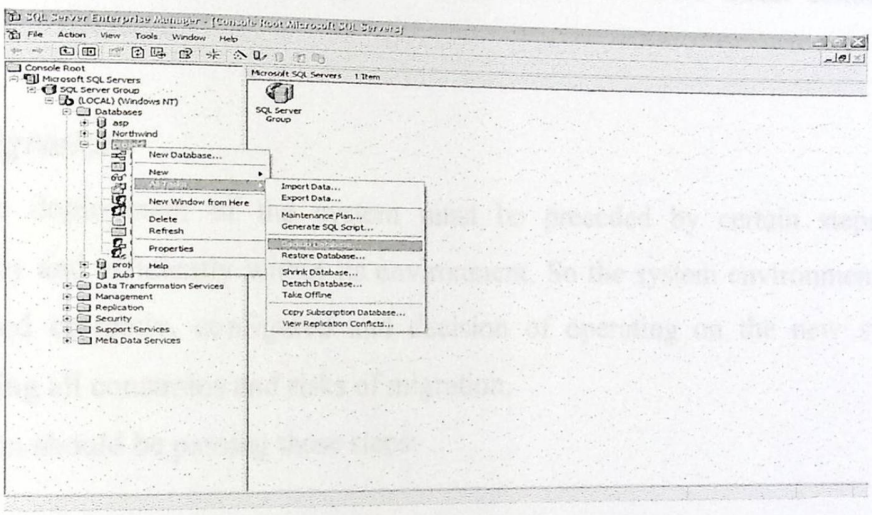


Figure (6.1) Database backup.

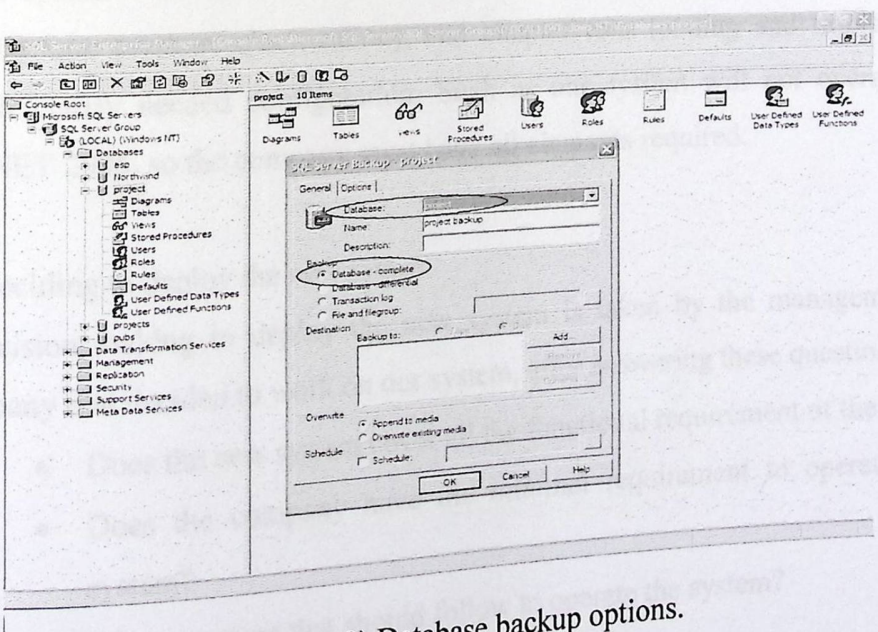


Figure (6.2) Database backup options.

- Upgrade system:

  * By getting newer copies of software, such as SQL server 2005 we improve performance of the system.

  * Make changes on the system to increase its effectiveness.

- Used some procedures in building system:

  * using validation to forbid any invalid inputs.

  * using data set and stored procedure to forbid direct connection with database

6.3 Migration:

The deployment of the system must be preceded by certain steps, to work effectively and efficiently within its environment. So the system environment should be established correctly, configured and decision of operating on the new system with considering all constrains and risks of migration.

Migration should be passing these steps:

1. Establishment of the production environment:

The system must have the minimal requirement to running it. Which described in chapter one (system specification), and chapter four (coding and implementation) which clarify needed configuration. Such as our system will not operate without VB.NET 2005, so the company must have all elements required.

2. Deciding to deploy the new system:

Decision making to deploy the new system is taken by the management of the company that decided to work on our system, after answering these questions:

- Does the new system cover all the functional requirement of the company?
- Does the company have the minimal requirement to operate the new system?
- What are steps that should follow to operate the system?

3. Running the system:

After the completeness of the system it can be running.

6.4 SQL server 2000 maintenance:

Database is the most important part in the system. (Project) in our system. That contains system data, tables, and stored procedures. And the other part is security that determines the privileges of each user of the system.

Main component in SQL server 2000 according to our system:

- The database (project).

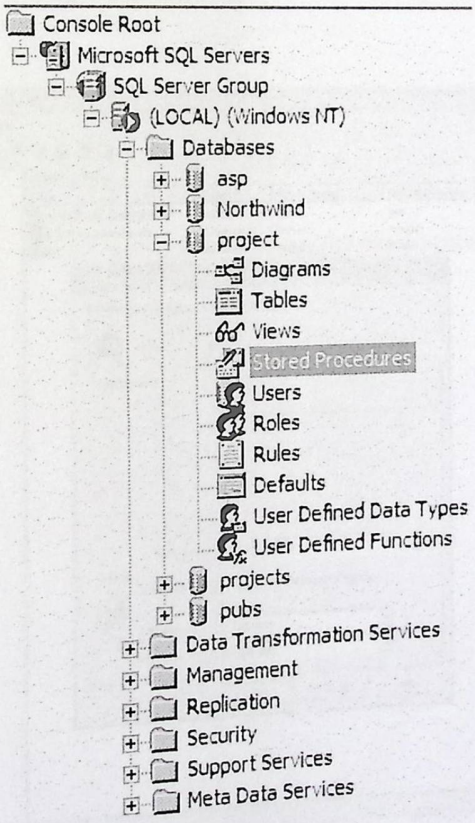


Figure (6.3) SQL server database.

- Security.

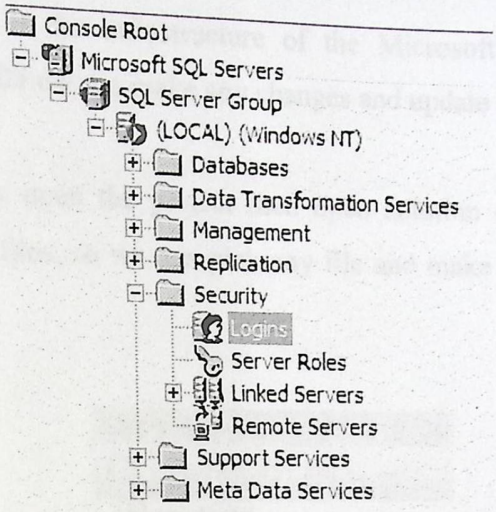


Figure (6.4) SQL server security.

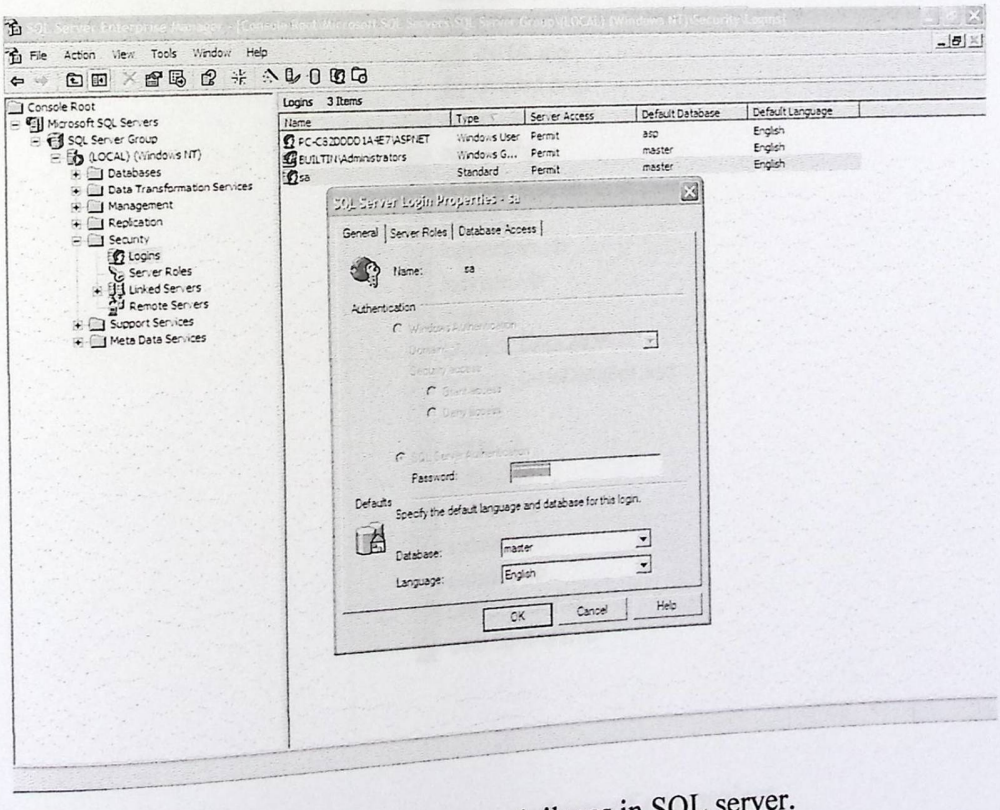


Figure (6.5) changing privileges in SQL server.

6.5 *The .NET framework maintenance*

.NET framework is the infrastructure of the Microsoft .NET technology, and through VS.NET 2005 we can make any changes and update any forms in the system.

We can justify by open the project then open solution explorer window, which contains all project files, so we can pick any file and make any justification on it or updated it.

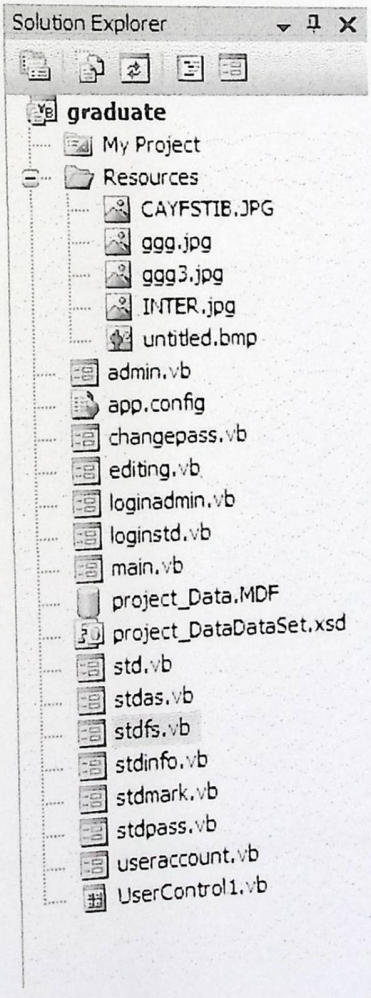


Figure (6.6) Solution explorer of our project.

6.6 *Mobile maintenance:*

Each period mobile should be maintained, by checking battery and credit.

6.7 *Nokia PC suite*

To maintain nokia Pc suite it need to be update each period.

Conclusion And Recommendation

Chapter Seven

- *Conclusion*
- *Recommendation and Future work*
- *Appendix*
- *References*

7.1 Conclusion and Recommendation

7.1. 1 Conclusion:

- The system provides easy access for student to get their university information.
- Our system provide strongly interface for users to attract attention for them, and minimize user errors by provide suitable validation message.
- We consider our system as a dynamic system, so it cans response as a real time system.

7.1.2 Skills:

- We have learned how to work as a team.
- We have several chances to operate with a different developing program as VS.Net language .and HyperTerminal, Microsoft office including Visio.
- We have knowledge to how work in a software system in different phase.

7.2 Recommendations And future work

7.2.1 Recommendations

- We recommend publishing our system in PPU University .
- Developing our system to include registration operation.
- Accomplishing administrator checked information using SMS.
- Using GSM Modem instead of intermediate mobile.
- We recommend PPU to agree with Jawwal about certain SMS price .

7.2.2 Future work

- Make administrator checking information using SMS.
- Develop this application to allow student to change his information using SMS.
- Develop our system to allow student to pay fees using SMS.

References:

Many of web site:

www.developershome.com
www.dream_in_code.com
www.logixmobile.com
www.arabteam2000.com

Appendix:

We have programmed this VB.NET 2005 code that is provide authentication by checking username and password, if is valid information allow student to enter card number , then checking card number ,if it is not used previously it send "ok " SMS to his mobile number which is stored in database.

//////////form1

```
Imports System.Data.SqlClient
Public Class Form1
    Dim flag As Boolean
    Dim conn As SqlConnection
    Dim ds As SqlDataAdapter
    Dim com As SqlCommand
    Dim dr As SqlDataReader
    Dim user As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        flag = False
        Me.conn.Open()
        dr = Me.com.ExecuteReader
        While (dr.Read)
            If TextBox1.Text = dr(0) And TextBox2.Text = dr(1) Then
                flag = True
                ' session("user") = TextBox1.Text
                '

                Label2.Text = dr(2)
                Form2.Show()
                Me.Visible = False

            End If
        End While
        If flag = False Then
            Label1.Text = "invalid username or password"
        End If
        Me.conn.Close()

        'Me.Visible = False



    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.conn = New SqlConnection
        Me.com = New SqlCommand
        conn.ConnectionString ="Data Source=HADENA230A;Initial
Catalog=mrhani;Integrated Security=True"
        com.Connection = Me.conn
        Me.com.CommandText = "select id,password,telnumber from users"
```

```
    End Sub
End Class
```

////////form 2

```
Imports System.Windows.Forms
Imports System.Data.SqlClient
Public Class Form2

    Dim conn As SqlConnection
    Dim ds As SqlDataAdapter
    Dim com As SqlCommand
    Dim dr As SqlDataReader
    Dim dss As DataSet

    Dim comSerial As New System.IO.Ports.SerialPort
    Dim atcCommand As String
    Dim sphnPhoneNo As String
    Dim cescQuote As Char
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Label1.Text = Form1.TextBox1.Text
        Label4.Text = Form1.Label2.Text

        Me.conn = New SqlConnection
        Me.com = New SqlCommand
        conn.ConnectionString = "Data Source=HADENA230A;Initial
Catalog=mrhani;Integrated Security=True"
        com.Connection = Me.conn
        Me.com.CommandText = "select num,flag from numbers"

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.conn.Open()
        Dim flag As Integer
        Dim dr As SqlDataReader
        ' Label6.Text = flag
        Me.com.CommandText = "update numbers set flag=1  where num='" &
Me.TextBox1.Text & "' "
        Me.com.ExecuteNonQuery()
        Me.conn.Close()
        Me.conn.Open()
        Me.com.CommandText = "select  num,flag from numbers where num=" &
Me.TextBox1.Text & " "
        dr = com.ExecuteReader
        dr.Read()
        flag = dr(1)

        Me.conn.Close()

        If flag = 1 Then
```

```
            Label3.Visible = False

            'sending(sms)
            With comSerial
                .PortName = "COM3"
                .BaudRate = "57600"
                .StopBits = IO.Ports.StopBits.One
                .DataBits = 8
                .Parity = IO.Ports.Parity.None
                .ReadBufferSize = 10000
                .ReadTimeout = 1000
                .WriteBufferSize = 10000
                .WriteTimeout = 10000
                .RtsEnable = True
                .Open()

                .Write("AT+CMGF=1" + vbCrLf)


                cescQuote = Char.ConvertFromUtf32(34)
                sphnPhoneNo = Label4.Text
                ' .Write("at+cmgs=" + cescQuote + sphnPhoneNo + cescQuote
+ vbCrLf + "suna" + Char.ConvertFromUtf32(26))

                atcCommand = "AT+CMGS=" + cescQuote + sphnPhoneNo +
cescQuote + vbCrLf + "ok" + Char.ConvertFromUtf32(26)

                .Write(atcCommand + vbCrLf)

                .WriteLine(atcCommand + "556456" + vbCrLf)

                TextBox2.Text = .ReadExisting()
                .DiscardOutBuffer()
                .Close()

            End With


        Else
            Label3.Visible = True
            Label3.Text = "number not found or the number used
previousely"
        End If

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

    End Sub
End Class
```