Palestine Polytechnic University

College of Information Technology and Computer Engineering

Information Technology Department

Automated Final Exam Scheduling System

Prepared by:

Bayan A. Sarsor

Doaa' E. Hamamdah

Islam H. Albashiti

Supervised by:

Dr. Mahmoud AlSaheb

This project was submitted in partial fulfillment of the requirements for the degree B.Sc. in IT

Jan, 2013

Palestine Polytechnic University

Hebron – Palestine

AUTOMATED FINAL EXAM SCHEDULING SYSTEM

Prepared by:

Bayan A. Sarsor Doaa E. Hamamdah Islam H. Albashiti

Under the project supervisor guidance and with the approval of all members of the board of examiners, the project was submitted to the Information Technology Department at the College of Information Technology and Computer Engineering to meet the requirements of the B.Sc of Information Technology.

Supervisor Name : Mahmoud Saheb Supervisor Signed : .....

Dept.Chairperson Name: ..... Dept.Chairperson Signed: .....

Jan, 2013

Abstract

This project is built to find a solution near to the optimal for final exam scheduling problem in College of Administrative science and Informatics in Palestine Polytechnic University, that has suffer from some problems: it takes more time and gives exams schedule has a lot of conflicts. The implemented solution is an automated system based on Genetic Algorithm.

In this project we applied genetic algorithm using java programming language to get relatively excellent final exam schedule with less number of conflicts, it has some constrains that we used.

After we finished the system, we get final exam schedule better than final exam schedule that produced manually, that is used in the college of Administrative Science and Informatics.

ملخص

يأتي هذا البحث من أجل حل مشكلة تعاني منها الجامعات بشكل عام وجامعة بوليتكنك فلسطين بشكل خاص وهي عملية جدولة الامتحان النهائي للكليات والتي تحوي دائما على مجموعة من التعارضات لدى الطلبة واضطرارهم لتقديم أكثر من امتحان في نفس اليوم أو نفس الجلسة، عدا عن ذلك الوقت الذي تستهلكه الجهات المسؤولة عن وضع البرنامج والذي قد يصل إلى عدة أيام.

في هذا البحث تم الوصول إلى حل يخفف من وطأة تعارض المساقات لدى الطالب الممتحن وذلك من خلال تطبيق خوارزمية "Genetic algorithm" في إنتاج برنامج امتحانات نهائي لطلبة كلية الهندسة والتكنولوجيا.

وبعد الانتهاء من بناء النظام وتشغيله وعمل مجموعة من تجارب الاختبار، تم تحليل تلك النتائج وعمل مقارنة مع نتائج الطريقة اليدوية المتبعة في كلية العلوم الإدارية ونظم المعلومات، ظهرت فجوة ليست بالقليلة بين برنامجي الامتحانات النهائية الذي تم مقارنتهما على نفس البيانات فقد أخرج النظام برنامج يحوي القليل من التعارض والتي هي أقل بكثير من الطريقة اليدوية المتبعة وقد حقق النظام الأهداف التي بني من أجلها.

Acknowledgement

We would like to express our thanks to ALLAH, who gave us the strength to complete the work. Then we would like to thank our supervisor Dr. Mahmoud Alsaheb. Also we would like to thank our family, mothers and fathers. Finally we would like to thank our friends, and anyone help us.

Contents

List of figures

List of tables

Chapter 1

Introduction

1. Overview
2. Scheduling problem
3. Problem statement
4. Literature review
5. Project objectives
6. Project domain
7. The organization of this document

1.1 Overview

Universities face every year a problem in building a suitable final exam scheduling, which are generated manually. Even if the developers try to develop the optimal table many times, they will not be able to do that, and the final table will have problems for instructors, as well as for students.

For students, the final table will have many problems, such as two exams at the same time, large gap between exams; sometimes difficult exams come in the first days, and other problems. For instructors, examples of the problems that face them are: a limited time to correct the exams, many observations for some instructors than others, etc.

College of Administrative Science and Informatics in Palestine Polytechnic University (PPU) for example spends a lot of days in producing a final exam schedule. They alter it many times because the first version often has many problems, and the final version still has many problems for students and instructors.

In this project, we will solve this problem by developing automated system using an intelligent algorithm. We will take into account in our project: exams and rooms to create complete table.

1.2 Scheduling problem

Scheduling problem can be defined as *"a problem of finding the optimal sequence for executing a finite set of operations (tasks or jobs) under a certain set of constraints that must be satisfied"* [10]; This means that there is a little number of time slots, and a lot of events. These events must be arranged around the little number of time slots. There are a number of constraints that are divided into hard constraints, which must be satisfied (Critical and have infinity cost),

and soft constraints, which are not essential to satisfy (have less cost compared to hard constraints) [3, 5].

There are many types of scheduling problem, and these types differ in terms of the kind of constraints involved [5]. Examples of scheduling problems are: course scheduling, shop scheduling, multiprocessor scheduling, staff scheduling, etc. In this project, the university exam scheduling problem will be handled. In exam scheduling, exams are the events, and time slots can be considered as the separate timeslots for exams. Examples of hard constraints are: no student can take two different exams at the same time. Example of Soft constraints: small gab between each exam for each student [13].

Scheduling problem is considered to be a non-polynomial-time hard (NP-hard) problem. This means that there is a solution, but suboptimal; there is no proof for the solution to check if it is optimal or not.

1.3 Problem statement

Students and lecturers constantly face this scenario; the timetable of final exams has many problems and not suitable for them, also they should give their comments in order to alter the table.

The necessity of developing an optimal exam scheduling table and in limited time becomes very important in order to keep the resources of universities such as time, and to increase the lecturers and students satisfaction. Exam scheduling involves assigning exams and rooms to a specific number of timeslots, and satisfies a set of constraints.

1.4 Literature review

Exam scheduling problem is that the basic challenge facing universities to schedule examinations. It has a limited time period and number of conflicts and a number of side-

constraints that different between universities.

To solve this problem we can use heuristic methods that can produce good timetables, but the size and complexity of modern university timetabling problems related to use more general problem solving algorithms, or Meta heuristics, such as stochastic search algorithm, simulated annealing, evolutionary algorithms, Final Exam Scheduling Package-SA, Graph Coloring, Hybrid Algorithm, memetic algorithms.

In the following section we will describe the Background and the literature review of exam scheduling problem, and how to solve it such as:

1.4.1 Stochastic search algorithm

Stochastic search is the method of choice for solving many hard combinatorial problems. and it has a ways to search, for solving problem such as: Simulated annealing Algorithm (SA), and Genetic algorithm (GA), and Final Exam scheduling Package (FESP)-(SA)[8].

1.4.2 Simulated annealing Algorithm:

"*Is based on idea from physics and is analogous to the physical annealing of solid*"[8]. And another definition : it is a probabilistic methods proposed for finding the global minimum of a cost function that may possess several local minima[1]. It's based on search strategy which keeps track of one feasible timetable.

Each timetable followed as neighbor, this neighbor is accepted as the current timetable if it has a lower cost. If the neighbor has a higher cost which is related to a control parameter called temperature increased that's mean the iteration increase, to reach the nearest optimal solution.

The process is analogous to the cooling process in actual annealing. One drawback with simulated annealing is that the cooling process can take a long time in order to achieve good result [3].

1.4.3 Genetic algorithm:

"*it based on the idea of natural evolution. They simulated natural population reproduction and selection operation to achieve optimal result*" [6]. And it analogous to Darwinian evolution. A-

population consist of timetables, chromosome is maintained, the fittest timetables that have the lowest cost are selected to form the basis of the next generation, to improving the overall fittest whilst maintaining diversity. A crossed-over based on the cut and divided chromosome to create new timetables.

However, researchers found an intelligent mutation operator to be more successful than two-parent cross over.

The Automated Scheduling and Planning group at the University of Nottingham, has developed genetic algorithms for examinations scheduling which employ a large degree of heuristic knowledge, to generate the initial population, and to improve the genetic operators.

Nottingham crossover works on a timetable period level, taking meetings scheduled in both parents first, and then according to a heuristic ordering, until no more may be placed in the child timetable without conflict. The use of this type of crossover lead to very efficient room usage [2].

1.4.4 Final Exam scheduling Package_SA:

" proposed a heuristic algorithm for exam scheduling and called it Final Exam scheduling Package(FESP)"[11]. Consists of 4 steps and each step solve part of the problem until they reached to the final solution.

After that the experience has on the previous algorithms, they concluded: the results of (GA) and (SA)it's show the good final exam scheduling that are better than (FESP_SA), also (SA) and (GA) allows to reduce the number of exams days without any effect on Conflicts, however (SA) is more suitable as it's faster than (GA), also all three algorithms produced better result when the maximum period for holding exams increased and worse when decreased[11].

1.4.5 Graph Coloring

Graph Coloring it's a set of node (V)and, a set of non-directed edges between node(E). The edge connecting tow nodes are said to be adjacent. This problem is known as:

Node coloring : "assigns colors to the nodes of the graph such that no two adjacent nodes have the same color" [7].

Edge coloring : "assigns colors to the edges of the graph such that no two adjacent edges have the same color. Two edges are said to be adjacent if they both share a node in common" [7].

The main idea of the algorithm: Researchers represent the courses by nodes of a graph, where two nodes are adjacent if the two corresponding courses are registered by at least one student. Then, it is required to assign each course represented by a node a time slot, such that no two adjacent nodes have the same slot.

This algorithm provided a mechanism to solve the problem for exam scheduling through the achievement of the objectives of fairness, accuracy, and optimal exam time period [7].

1.4.6 Hybrid Algorithm

The researchers present hybrid algorithm for examination timetabling, that consisting of three phases: a constraint programming phase to develop an initial solution (feasible timetable) satisfying all constraints, a simulated annealing phase to improve the quality of solution, and a hill climbing phase for further improvement [9][11][12].

Examination timetabling is a good-studied combinatorial the best problem. We present a new hybrid algorithm for examination timetabling, contain of three level: a constraint programming phase to develop an initial solution, a simulated annealing level to improve the quality of solution, and a hill climbing level for addition improvement. The examination timetabling problem at the University of Melbourne is introduced, and the hybrid method is proved to be superior to the current method employed by the University. Finally the hybrid method is compared to established methods on the publicly available data sets, and found to perform good in comparison.

1.5 Project objectives

This project is expected to achieve the following objectives:

- To build a stand-alone program that generate a final exam scheduling and reduce the time that is consumed at traditional way.
- Reduce complexity.

- Avoid mistakes in final exams table.
- Maximize students study time.

1.6 project domain

The system will help the Palestine Polytechnic University to create final exam scheduling.

1.7 The organization of this document

This document is organized into the following chapters:

- **Chapter 1: Introduction**

  This chapter gives an overview about examination problem, scheduling problem, problem statement in general, and objectives of this project.

- **Chapter 2: Planning and system requirements**

  This chapter explains the project plan and schedule, also mention the methods that used for fathering information. It describes the current system in the college, and the alternatives. The primary section in this chapter is the functional and non-functional requirements. Finally, the feasibility study and potential risks are mentioned.

- **Chapter 3: System specifications**

  In this chapter, the functional requirements are specified in more details; also the models like context diagram and dataflow diagram are illustrated. Finally, it gives a general description of the database.

- **Chapter 4: System design**

  In this chapter, we will explain the problem modeling and represent the solution and the algorithm that we applied to solve the problem and steps and functions of the algorithm.

- Avoid mistakes in final exams table.
- Maximize students study time.

1.6 project domain

The system will help the Palestine Polytechnic University to create final exam scheduling.

1.7 The organization of this document

This document is organized into the following chapters:

- **Chapter 1: Introduction**

  This chapter gives an overview about examination problem, scheduling problem, problem statement in general, and objectives of this project.

- **Chapter 2: Planning and system requirements**

  This chapter explains the project plan and schedule, also mention the methods that used for fathering information. It describes the current system in the college, and the alternatives. The primary section in this chapter is the functional and non-functional requirements. Finally, the feasibility study and potential risks are mentioned.

- **Chapter 3: System specifications**

  In this chapter, the functional requirements are specified in more details; also the models like context diagram and dataflow diagram are illustrated. Finally, it gives a general description of the database.

- **Chapter 4: System design**

  In this chapter, we will explain the problem modeling and represent the solution and the algorithm that we applied to solve the problem and steps and functions of the algorithm.

- **Chapter 5: System implementation**

  This chapter comes after system design, we will begin programming the system and in this chapter we will explain the environment of implementation phase that includes needs of project from several tools.

- **Chapter 6: system discussion and testing**

  In this stage, we will apply system testing on the entire system and check if components of system work correctly, also we will compare the results to find best solution.

- **Chapter 7: system maintenance**

  This chapter represents the final phase of the project life cycle, we will describe maintenance system, upgrading and backup.

- **Chapter 8: Conclusion**

  In this chapter, we will list the conclusion and recommendations for our project.

Chapter 2

Planning and System Requirements

1. Introduction
2. Project schedule
3. Information gathering
4. Description of current system
5. Alternatives
6. Functional requirements
7. Nonfunctional requirements
8. Feasibility study
9. Risks analysis

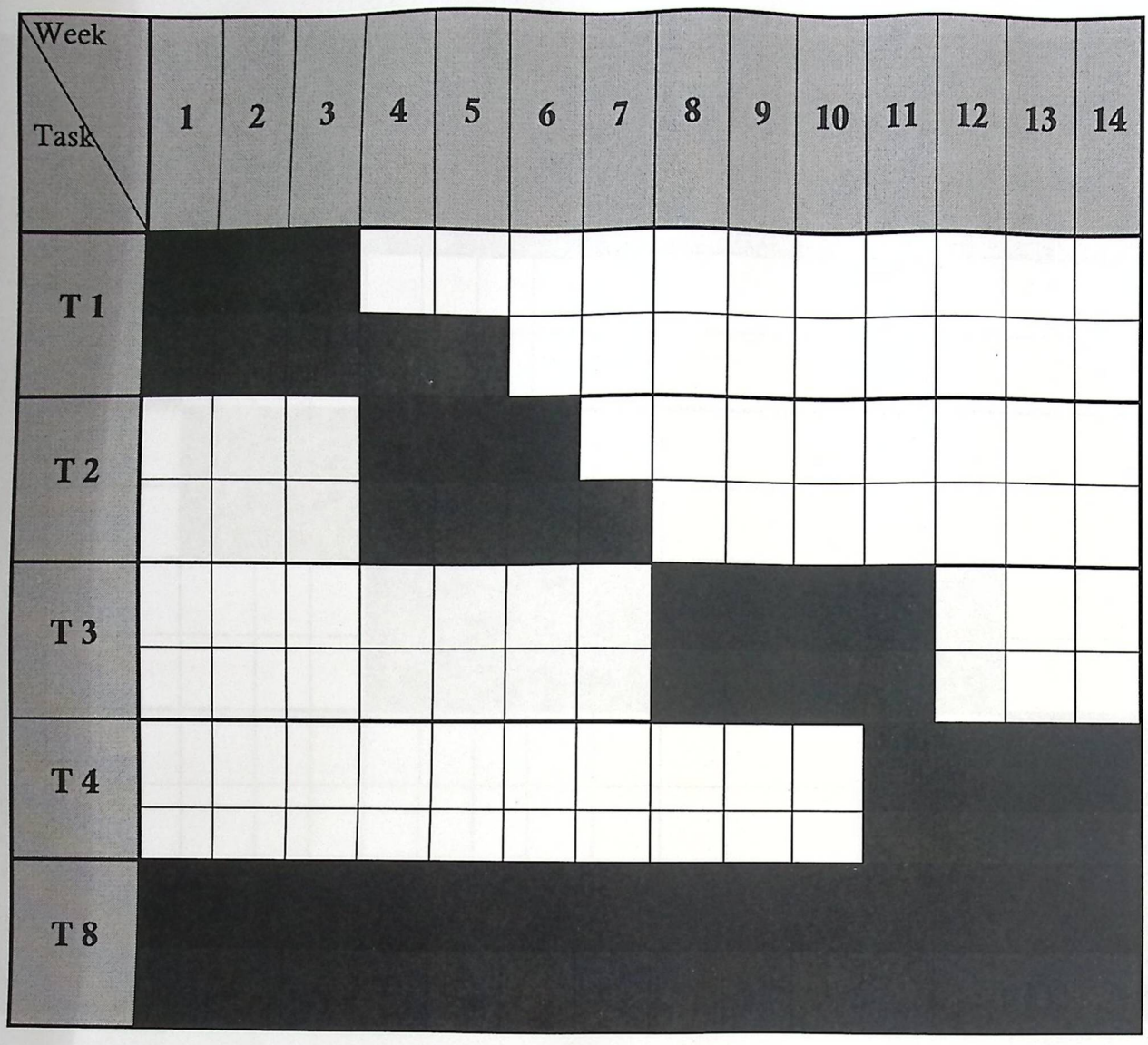

Figure 2.1:Gantt chart for first semester

2.1 Introduction

In this chapter, we will describe the project schedule and project plan. After that, we will explain the methods that we have used to gather information about the problem. Then, we will describe the system functional and non-functional requirements. Finally, we will explain the feasibility and analyze the risks that may face the project.

2.2 Project schedule

We have a set of tasks, and by the end of the semester, we should finish them. The tasks and the number of weeks needed is shown in the table (2.1).

No.	Symbol	Tasks	Number of weeks	Number of semester
1	T1	Literature review	5	First and second semester
2	T2	System planning	3	First semester
3	T3	Requirements of collection	4	First semester
4	T4	Requirements specification and analysis	4	First semester
5	T5	System design	3	second semester
6	T6	System programming and implementation	11	second semester
7	T7	System testing	3	second semester
8	T8	System documentation	33	second semester

Table 2.1: Tasks distribution

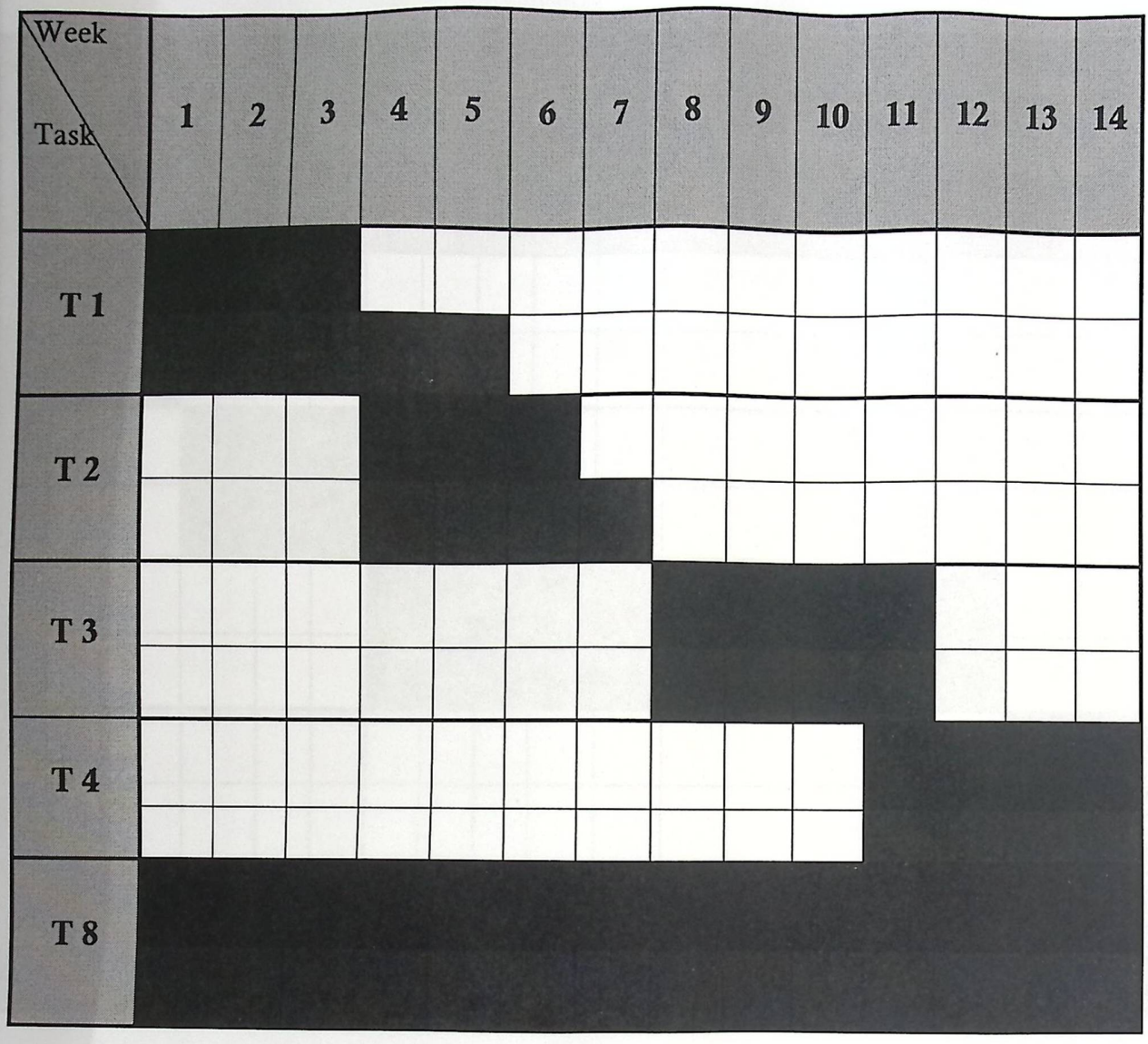

Figure 2.1:Gantt chart for first semester

Expected

Actual

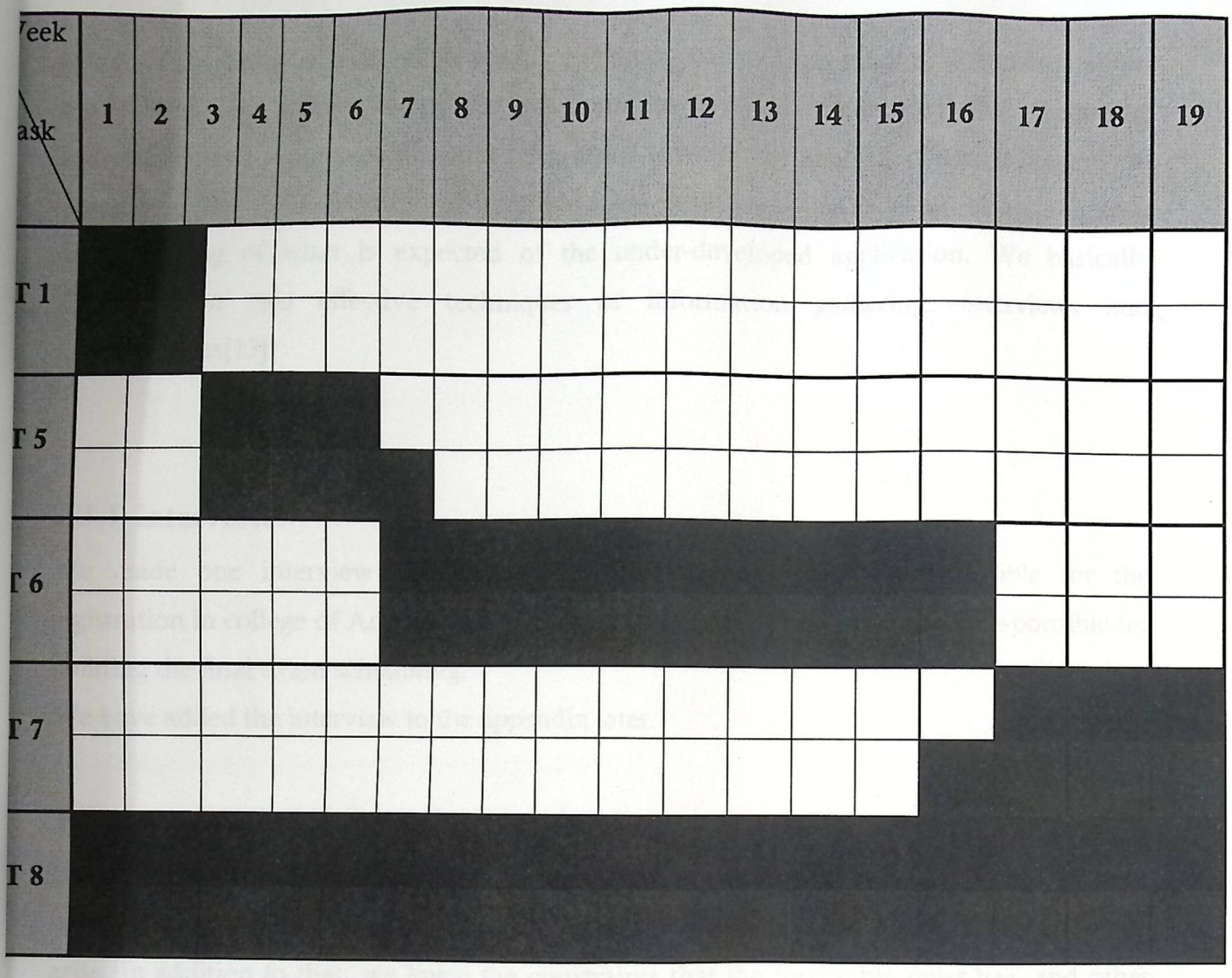

Figure 2.2: Gantt chart for second semester

Expected

Actual

2.3 Information gathering

Much of the knowledge in requirements elicitation comes from reading documents about the subject and talking to people who are involved in similar systems. Gathering Information is a significant step and basic stone towards the analysis phase. It helps us in organizing the large amount of required data and build up a clear non-ambiguous understanding of what is expected of the under-developed application. We basically depended on two effective techniques of information gathering, interviews and questionnaires[13].

2.3.1 Interview

We made one interview with Ms.Worod Al-Sharabati, who is responsible for the registration in college of Administrative Science and Informatics , and she is responsible for building the final exam scheduling.

We have added the interview to the appendix later.

What we concluded from the interview?

From this interview, we concluded the description of the current system in the college, which is done almost 90% manually. We knew the details about it, and the problems that arise. In addition to that, we knew the constraints that the final table must has, and other information such as the total number of students, rooms and exams. The interview helped us in understanding the problem deeper and clearer[13].

2.3.2 Questionnaire

We prepared a questionnaire and then distributed it to a sample of students in the college.

We have added the questionnaire to the appendix later[13].

What we concluded from the questionnaire?

From this questionnaire, we concluded the most important things that the students wish in the final exam scheduling, and what are the difficulties and problems that face them. Also, we concluded some of the requirements for our system, the results of questionnaire in summary, most of students reject to have two exams in same period and same day and they suffer from this point.

2.4 Current System Description

The current system for final exam scheduling in our college is done manually. The process takes at least a week. The system consists of students, exams and rooms. The academic supervisor place the examination table depending on the academic students plans. At each day, they have two periods in order to avoid the depletion of the university's resources (rooms and observers). Each period can serve at most 200 students (400 students per day).

The system finds the final exam scheduling with minimum number of conflicts. It is specialized for college of Administrative Science and Informatics in PPU in order to reduce the time and effort done by the human since it takes at least a week.

In this system, the mechanism that is used to distribute students into the rooms, exams on the students, and exams on the days are explained consecutively.

2.4.1 The distribution of students in the rooms

Each exam's students are distributed among the rooms depending on the capacity of each room, and on the number of exams that the room accommodates.

For example, if we have 120 students for "Organizational Behavior Course" and 60 students for "Computer Programming Course", we will distribute 60 students from the "Organizational Behavior Course" and 30 students from the "Computer Programming Course", and then we will put them in the rooms that have a capacity of at least 90 students.

2.4.2 The distribution of exams to the students

Exams are distributed for the students depending on the academic plan for those students. The constraints in this case are: try to make the student should not have two exams at the same time and no more than two exams in the same day. In addition, the exams that are in the same level must not be in the same day.

The exams for courses which are not possible to be in the same semester can be scheduled in the same day, such as Arabic Language, Islamic Culture and Palestine History. On the other hand, the course and its prerequisite course must not be in the same day to avoid conflicting.

2.4.3 The Distribution of exams on the days

The university policies limit the exams days to maximum 10 days, so each day can contains two exams as mentioned above. In addition to that the Saturday (which is the week end day) can have one exam.

2.5 Alternatives

The alternative of this current system is developing an automated system that creates the final table in few minutes, and depending on intelligent algorithms, so it has a minimal number of conflicts, and increases the usability of university resources, such as time and space.

Our system will take into account the distribution of exams, rooms to a specific number of time slots, and satisfying a set of hard and soft constraints.

The hard constraints are:

1- The maximum number of days is 10 (20 time slots).
2- Each day must have two time slots.

The set of soft constraints are:

1- The student should not have two exams at the same period.
2- The student should not have two exams at the same day.

2.6 Functional requirements

The minimal set of functional requirements that the system should perform are:

1- Import the data that related to students, courses, rooms and timeslots in the system.
2- Import the set of hard and soft constraints.
3- Store the imported data.
4- Update the system database manually.
5- Build the final timetable with minimal number of conflicts.
6- Display reports, which are: college exams timetable, rooms table, and student's exams table.

2.7 Non-functional requirements

1- Usability: the system must provide an easy to use Graphical User Interface (GUI), it will be easy to learn and no hidden keys.
2- Flexibility: the system must work with any new requirement. The system structure must be adaptive to new university rules.
3- Maintainability: if some errors occur in the system, it should be able to be maintained by the developers or by another team.
4- Speed: the system gives its results in a few times.

2.8 Feasibility study

This section will describe the software and the hardware needed to develop the project.

2.8.1 Software development resources

The system needs development software resources as explained in table (2.2):

Item	Quantity	Price $
Java (JDK)	3	0
Netbeans	3	0
MS-Office Professional 2010	3	89.45
Latex	1	0
Total		268.35

Table 2.2: Software development resources

2.8.2 Hardware development resources

To develop the system, we need to use a computer with specifications, according to the programs that will be used to build the system. The following table (2.3) shows the recommended hardware that needed to build our system:

Item	Specifications	Cost $
Personal Computer	CPU: 2 GH Ram: 2 GB H.D: 80 GB Monitor: 19 Inch	394

Table 2.3: Hardware development resources

2.8.3 Human resources

The following table (2.4) shows the human resources:

Name	No. of workers	Total cost
System analyst	3	1500
Programmers	3	4800
Total		3600

Table 2.4: Human resource

2.8.4 Total cost:

In table (2.5), we show overall project cost

Software cost	Hardware cost	Human resources	Total cost
268.35 $	394 $	3600 $	4262.35 $

Table 2.5: Total cost

2.9 Risks analysis

In this section, we will list the expected risks that we may face, and the proposed solutions.

The expected risks are:

1- The time constraint, time is limited by the end of the course.
2- System failure.
3- One or more of the workgroup member quit for any reason.

The proposed solutions are:

1- Commitment to time schedule.
2- To avoid system failure and loss of data, we will make periodic system backup.
3- The rest members will continue working by increasing their tasks.

Chapter 3

System Specification

1. Introduction
2. General system description
3. System development life cycle
4. Requirements specifications
5. Context diagram
6. Data flow diagram
7. Database general description
8. Test plan

3.1 Introduction

In this chapter we will describe the system functional requirements in more details and specifications. Also we will illustrate some models such as context diagram. Finally we will give general description of the database.

3.2General system description

The proposed system supposed to build the final exam scheduling table in the college of Administrative Science and Informatics in short time. The system consists of these main parts: exams information, students' information, and rooms' information. These data should be stored in the system.

The system will find the sub-optimal solution by applying an intelligent algorithm, and the result will be outputted through the reports.

3.3System development life cycle

Our project will passes in several stages, that broken down to fundamental levels from define the problem and put functions and requirements of system for solve this problem to last stage that tests the system if it do correctly without mistakes.

It is important to understand steps that we should achieve it and it helps to walk sequentially through building our system.

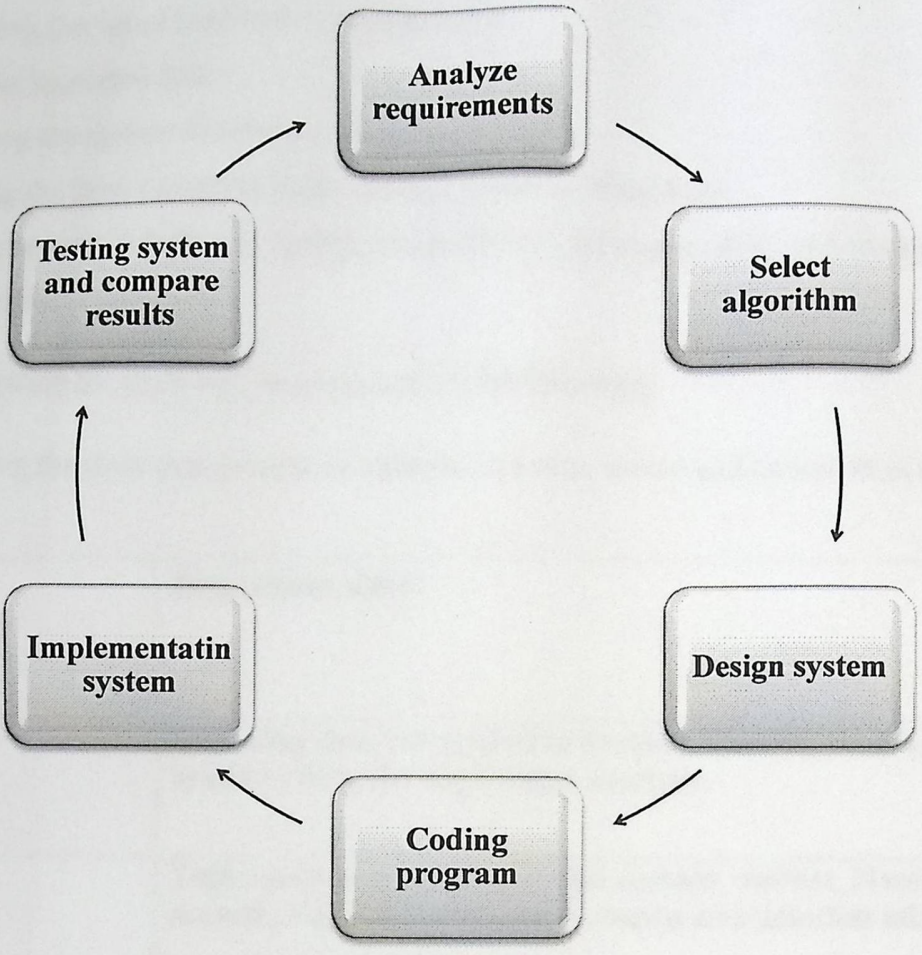

Figure (3.1): System life cycle

3.4 Requirements specifications

In this section, we will determine the specification for each functional requirement in the system.

The set of functional requirement are:

1. Importing the data that related to students, courses, rooms and timeslots in the system.

2. Importing the set of hard and soft constraints.
3. Store the imported data.
4. Updating the system database manually.
5. Building the final timetable with minimal number of conflicts.
6. Display reports, which are: college exams timetable, rooms table, and student's exams table.

The specification for each will be explained in the following:

1- Importing the data that related to students, courses, rooms and timeslots in the system.

Function	**Importing data**
Description	Importing data that related to students, courses, rooms, and timeslots from the registration database.
Input	Tables such as student table that contain student_Name, student_No, …etc, and course, rooms and timeslots tables.
Source	Registration database
Output	Imported data in the system.
Target (destination)	System database
Requirements	Logging on as administrator
Precondition	Access to the registration database.
Post condition	Old data will be removed and only the new data will be available.

Table 3.1: Importing data

2. Importing the set of hard and soft constraints.
3. Store the imported data.
4. Updating the system database manually.
5. Building the final timetable with minimal number of conflicts.
6. Display reports, which are: college exams timetable, rooms table, and student's exams table.

The specification for each will be explained in the following:

1- Importing the data that related to students, courses, rooms and timeslots in the system.

Function	**Importing data**
Description	Importing data that related to students, courses, rooms, and timeslots from the registration database.
Input	Tables such as student table that contain student_Name, student_No, …etc, and course, rooms and timeslots tables.
Source	Registration database
Output	Imported data in the system.
Target (destination)	System database
Requirements	Logging on as administrator
Precondition	Access to the registration database.
Post condition	Old data will be removed and only the new data will be available.

Table 3.1: Importing data

2- Importing the set of hard and soft constraints.

Function	Importing constraints
Description	Upload excel file that contains hard and soft constraints.
Input	Excel file that contain set of hard and soft constraints.
Source	Administrator
Output	Imported data
Target	System database
Requirements	Logging on as administrator.
Post condition	Enable administrator to save and update the data.

Table 3.2: Importing constraints

3- Store the imported data.

Function	**Storing data**
Description	Enable the administrator to save the imported data to the system database.
Input	Imported data such as student's information, constraints, ...etc.
Source	Administrator
Output	New data in the system database.
Target	System database
Requirements	Logging on as administrator.
Precondition	Import the data.
Post condition	Enable the administrator to update the data.

Table 3.3: Storing data

4- Updating the system database manually.

Function	**Updating the system database manually**
Description	Enable the administrator to update the database such as insert, delete, modify.
Input	Stored data
Source	System database
Output	New updated data
Target	System database
Requirements	Logging on as administrator.
Precondition	Import the data, save the data
Post condition	Enable administrator to save and update the new data.

Table 3.4: Updating database

5- Building the final timetable with minimal number of conflicts.

Function	**Building final timetable**
Description	The data will be entered to the algorithm that will make some processes, and then build the final timetable.
Input	Stored data
Source	System database
Output	Final solution
Target	System database
Requirements	Logging on as administrator.
Precondition	Import the data, save the data

Table 3.5: Building final table

6- Display reports, which are: college exams timetable, rooms table, and student's exams table.

Function	**Display reports**
Description	Display reports, which are: college exams timetable, rooms table, and student's exams table.
Input	Stored data, final solution
Source	Database, administrator
Output	Reports
Target	
Requirements	Logging on as administrator
Precondition	Build the final solution
Post condition	Modify the reports

Table 3.6: display reports

3.5 Context diagram

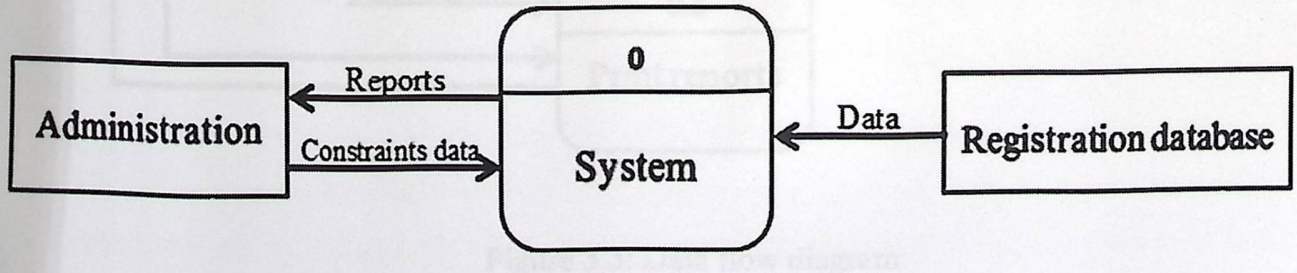

Figure 3.2: Context diagram

3.6 Data flow diagram

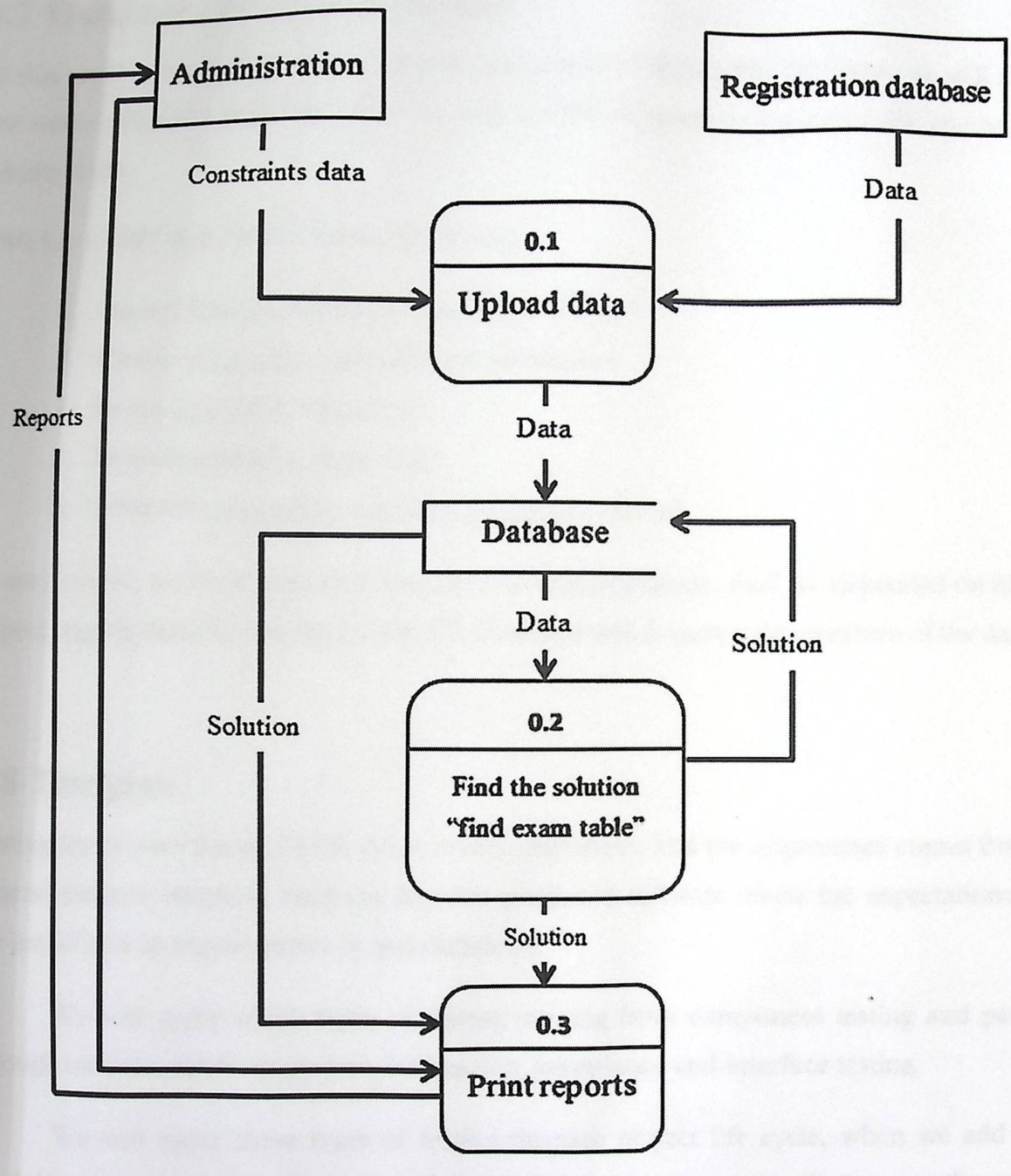

Figure 3.3: Data flow diagram

3.7 Database general description

In this section, we will present a general description of the system database, we will resent the tables and their attributes, also we will use ER-diagram to represent the entities and relationships.

We have 5 tables in our database, which are:

1. Student (stdNo, stdName,stdMajor, stdYear)
2. Course (courseNo, courseName, instructor)
3. Room (roomNo, Capacity)
4. Period (periodNo, type, day)
5. Schedule (courseNo, roomNo, periodNo, stdNo)

In our system , we have used files instead of building database. And we depended on files to import/export data. See figure 3.3 the ER-Diagram which shows the structure of the data.

3.8 Test plan

Testing the project during its life cycle is very important, and the importance comes from its general purpose which is verifying that the produced software meets the expectations that are predefined as requirements or specifications.

We will apply seven types of testing starting from component testing and passing through module, database, system, integration, acceptance and interface testing.

We will apply these types of testing through project life cycle, when we add new module or function, and when we will finish the project, then we will compare the results with requirements of system even treat problems and do any task not exist in system, at last we will put recommendations about results.

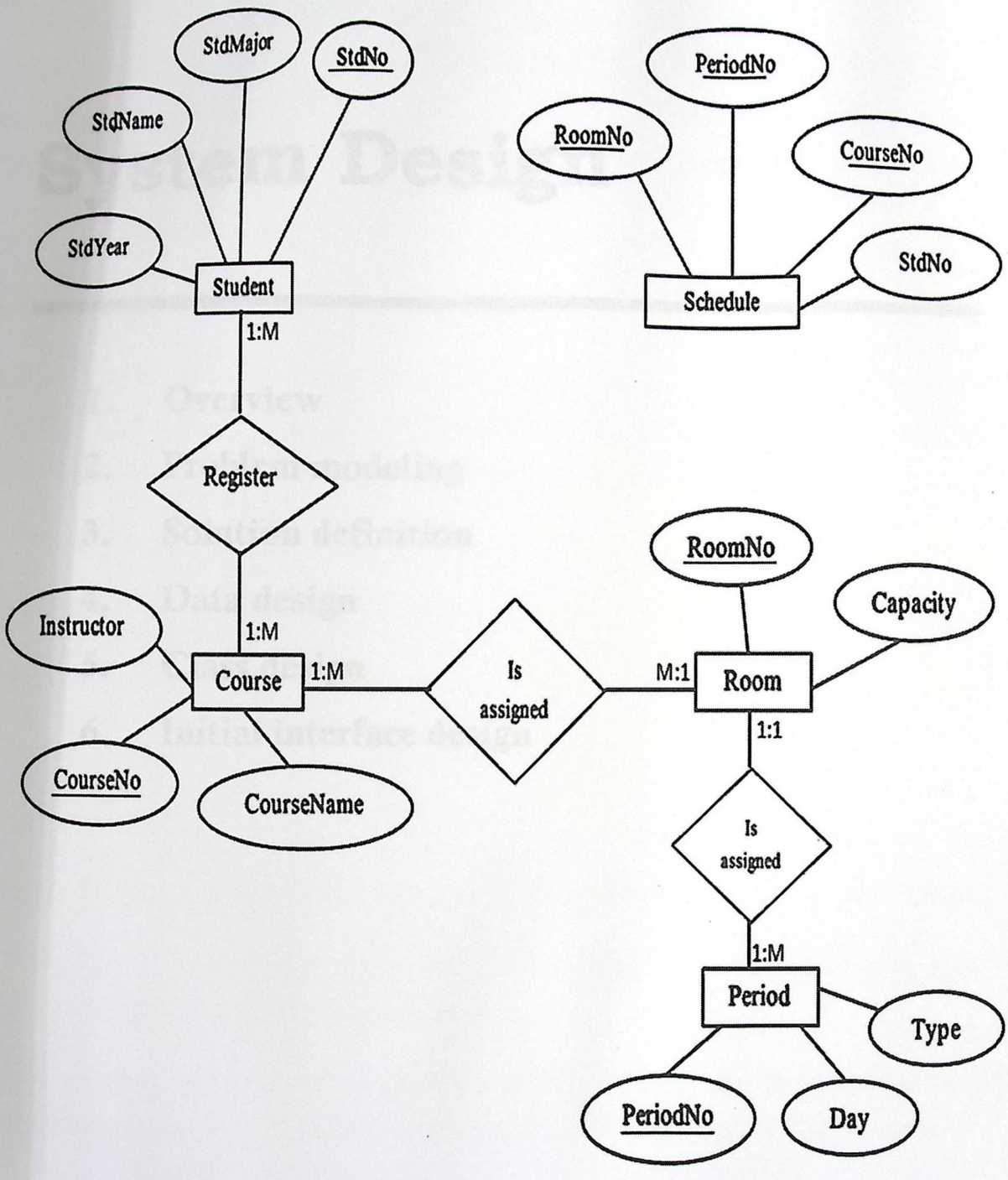

Figure 3.4: ER-diagram of system database

Chapter 4

System Design

1. Overview
2. Problem modeling
3. Solution definition
4. Data design
5. Class design
6. Initial interface design

4.1 Overview

In this chapter, we will present the problem modeling, then we will present the solution and the algorithm that we applied to solve the problem. After that, we will describe the algorithm steps and functions, database schema and the Graphical User Interface (GUI).

4.2 Problem modeling

In this section we will start working on the design for a tool that will solve the final exam scheduling problem for college of Administrative Science and Informatics, and to handle the problem correctly, we have defined the problem to be in six different sets:

D: The number of days for the exam period.

T: The number of time slots on each day.

C: A set of courses.

R: A set of rooms.

S: A set of students.

We want to find a schedule $K : C \rightarrow (\{1..D\},\{1..T\},R)$ that is acceptable by the students.

$D = \{d_1, d_2, d_3, \ldots, d_n\}$, each element in D is a number (d_i) where d_i is the day number. Number of days (n) will be specified at run-time.

$T = \{t_1, t2, t_3, \ldots, t_m\}$, each element in T is a number (t_i) where t_i is the number of time slot in day (d_i), the number of timeslots (m) in any a day will be also specified at run-time.

$C = \{c_1, c_2, c_3, \ldots, c_l\}$, each element in C is a vector of courses (cNo,nOs) where

cNo is the course number and, nOs is the number of student in each course.

$R = \{r_1, r_2, r_3, \ldots, r_o\}$, each element in R is a vector (rNo, cap) where rNo is the room number, and cap is the maximum capacity of this room.

4.1 Overview

In this chapter, we will present the problem modeling, then we will present the solution and the algorithm that we applied to solve the problem. After that, we will describe the algorithm steps and functions, database schema and the Graphical User Interface (GUI).

4.2 Problem modeling

In this section we will start working on the design for a tool that will solve the final exam scheduling problem for college of Administrative Science and Informatics, and to handle the problem correctly, we have defined the problem to be in six different sets:

D: The number of days for the exam period.

T: The number of time slots on each day.

C: A set of courses.

R: A set of rooms.

S: A set of students.

We want to find a schedule $K : C \rightarrow (\{1..D\},\{1..T\},R)$ that is acceptable by the students.

$D = \{d_1, d_2, d_3, \ldots, d_n\}$, each element in D is a number (d_i) where d_i is the day number. Number of days (n) will be specified at run-time.

$T = \{t_1, t2, t_3, \ldots, t_m\}$, each element in T is a number (t_i) where t_i is the number of time slot in day (d_i), the number of timeslots (m) in any a day will be also specified at run-time.

$C = \{c_1, c_2, c_3, \ldots, c_l\}$, each element in C is a vector of courses (cNo,nOs) where

cNo is the course number and, nOs is the number of student in each course.

$R = \{r_1, r_2, r_3, \ldots, r_o\}$, each element in R is a vector (rNo, cap) where rNo is the room number, and cap is the maximum capacity of this room.

$\mathbf{S} = \{s_1, s_2, s_3, \ldots\ldots, s_p\}$, each element in S is a vector (sNo, ci) where sNo is the student number, and ci is an array indicating to the courses that this student have registered.

4.3 Solution definition

Scheduling problem is considered to be a non-polynomial-time hard (NP-hard) problem, and the solution domain is very large. Many algorithms when are applied to solve this problem like hill-climbing find the solution that is in local minima. This leads to unsatisfactory solution. So, we need an algorithm that finds global optima well. Genetic algorithm is the technique that we have chosen, and we show other approaches in the literature review [3].

4.3.1 Genetic algorithm

Genetic algorithm is classified as a type of evolutionary search methods, which are a part of artificial intelligence. It does not guarantee to find the optimal solution. The idea comes from the nature; there are a large number of species, and the survival of the strongest and the fittest.

In genetic algorithm, there is a population, which contains a number of solutions. The solutions are called individual or chromosome. The chromosome consists from many genes, and each gene represents one case for a specific solution. Genetic algorithm generates the first generation of the population randomly. Then, it measures the quality for each chromosome according to a fitness function. After that, it generates a second generation through cross over operation and make mutation for each chromosome. Finally, it chooses the chromosomes according to the fitness function. These steps are repeated until find the suitable chromosome. Here is a pseudo code for the algorithm:[20],[21].

initializePopulation P

for each *soli* from P **do**

calculateFitness(sol)

repeat

select two parents *sol1* and *sol2* from P

child = crossover(*sol1, sol2*)

mutate(*child*)

calculateFitness(*child*)

replaceSome(*P, child*)

until stop condition meet

end for

The steps in more details are discussed below:

1- Initialize the population

G = {g1, g2, g3, ….., gn} is the set of genes which constitute the individual (chromosome), each gene is a vector (exam, dayNo, timeNo, roomNo) where exam is the course number, day is the day number, time is the time slot (the period of a day), and the room is the room number.

G1	G2	.....	Gn
Course1	Course2	....	Course n
dayNo	dayNo		dayNo
timeNo	timeNo		timeNo
roomNo	roomNo		roomNo

Table4.1: The individual

The next flowchart explains the steps of initialize the first generation of the population:

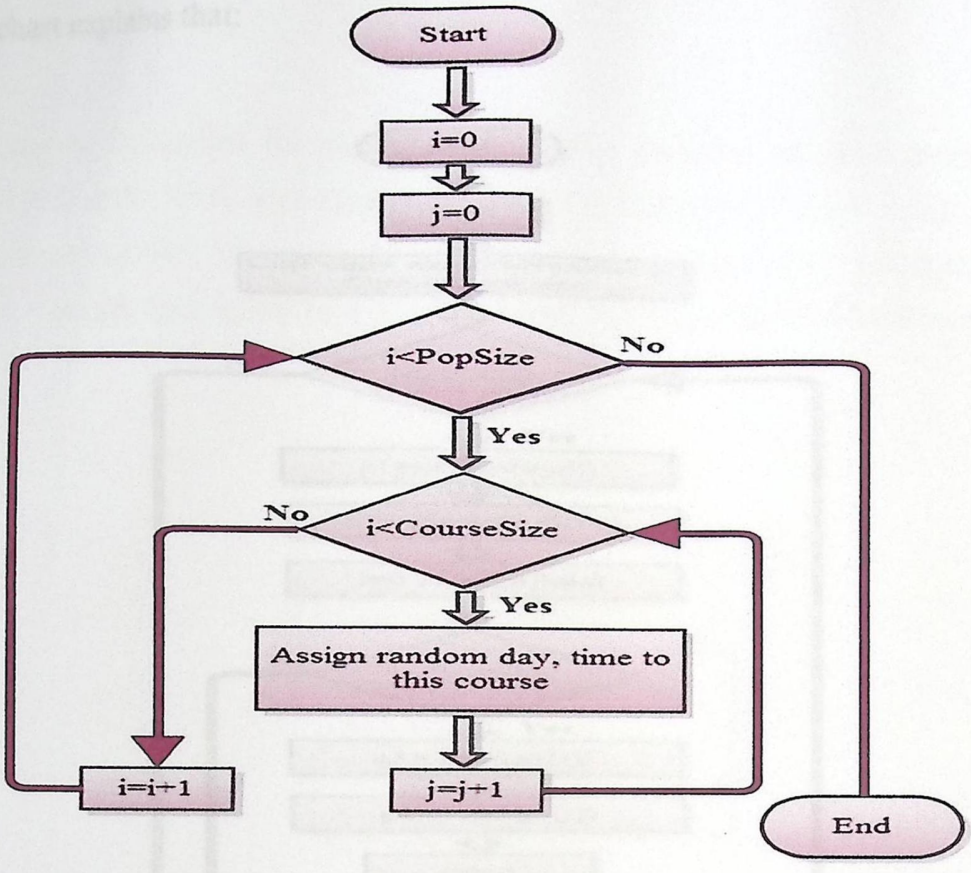

Figure 4.1: Initialize the first generation

2- Evaluate individuals

Every individual has a cost; sum of hard and soft costs. Each constraint has a cost, and hard constraints have a big cost rather than soft constrains. The fitness function is used to determine if the individual live or die depending on his cost.

3- Crossover

The crossover process happens for two individuals, if the number of population is 20 (10 pairs and 20 children), five crossovers will occur(in the children only). The crossover

operator chooses a point in first individual, and cuts it, then pastes it to the second individual, and cuts from the second individual and pastes it to the first individual, the following flowchart explains that:

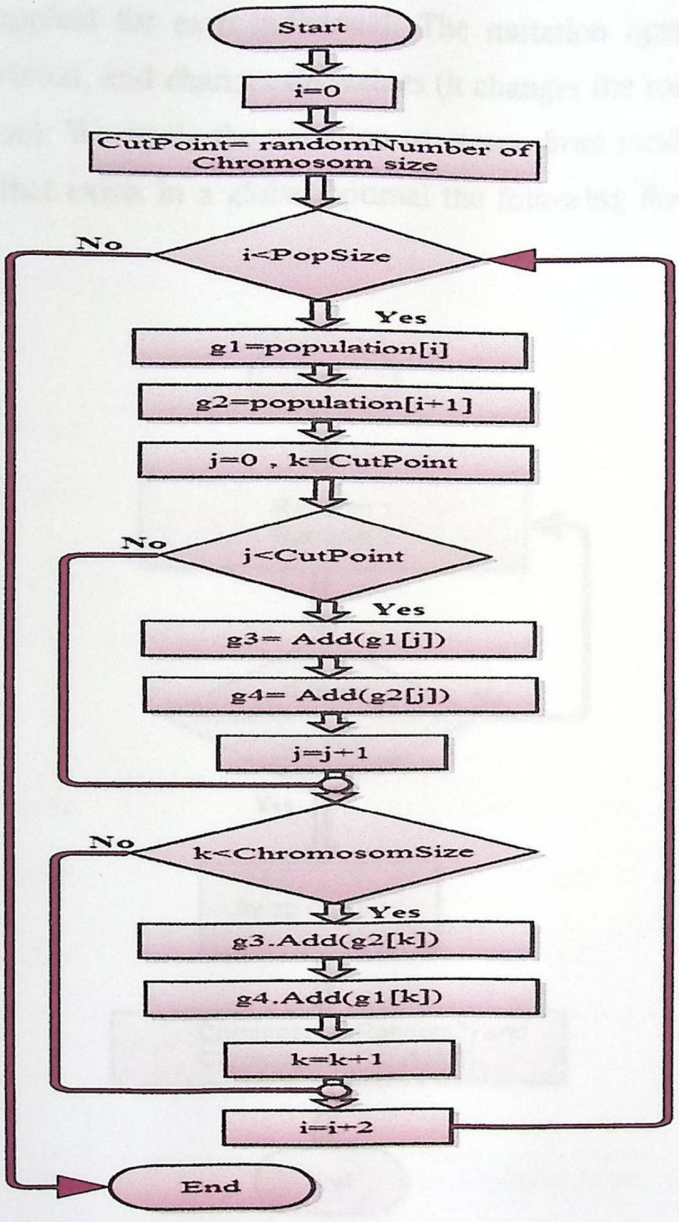

Figure 4.2: Cross over

Through our system, we calculate the cost according to the following equation:

Cost = number of hard constraints * hard constraint weight + number of soft constraints * soft constraint weight.

4- Mutation

The mutation process is applied for each individual. The mutation operator chooses a random point in each individual, and changes the values (it changes the room number and day number for some course). We apply the mutation to escape from local maximum, we want to find the solution that exists in a global optimal the following flowchart explains that:

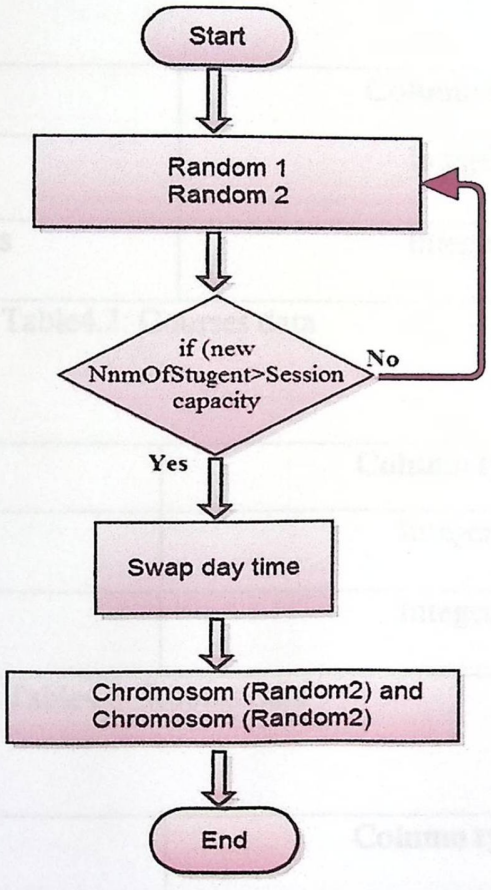

Figure 4.3: Mutation process

5- Select individuals

In each repetition we choose the half number of population. The selection process depends on the calculated fitness value. We repeat the above steps until we find the near optimal individual.

4.4 Data design

In this project, we have used files to store the data we needed. The data we collected it from the registration department in the university. This real data is related to the College of Administrative Science and Informatics . The following tables represent the components for each file:

Column name	Column type
courseNumber	Integer
numberOfStudents	Integer

Table4.2: Courses data

Column name	Column type
roomNumber	Integer
roomCapacity	Integer

Table4.3: Rooms data

Column name	Column type
studentId	Integer
studentCourses	Integer

Table4.4: Students data

In our project, we have solved the problem within the object oriented principles using java language, so we have designed all the sets that we mentioned in the previous section into classes. Each class is described in the following tables:

Name	Data type	Description
courseNo	Int	Course number
noOfStudents	Int	Number of students who registered in this course

Table 4.5: The class of courses (Course)

Name	Data type	Description
roomNo	int	Room number
Cap	int	Capacity of this room

Table4.6: The class of rooms (Room)

Name	Data type	Description
stdNo	Int	Student number
stdCourses	ArrayList<int>	Array list of courses that this student have registered

Table 4.7: The class of students information(Student)

Name	Data type	Description
Course	Course	Course number and number of student from course class
dayNo	Int	Capacity of this room
timeNo	Int	
Room	Room	Room number and capacity from room class

Table 4.8: The class of gene information(Gene)

Name	Data type	Description
List	ArrayList<Gene>	Array list of genes from Gene class
Cost	Int	Cost of this chromosome(hard costs and soft costs)

Table4.9: The class of chromosome information(Chromosome)

4.5 Class design

In this section, we will present the data members and the methods as a class diagram. We took the advantage of the encapsulation in object oriented programming. The following figures show the diagrams:

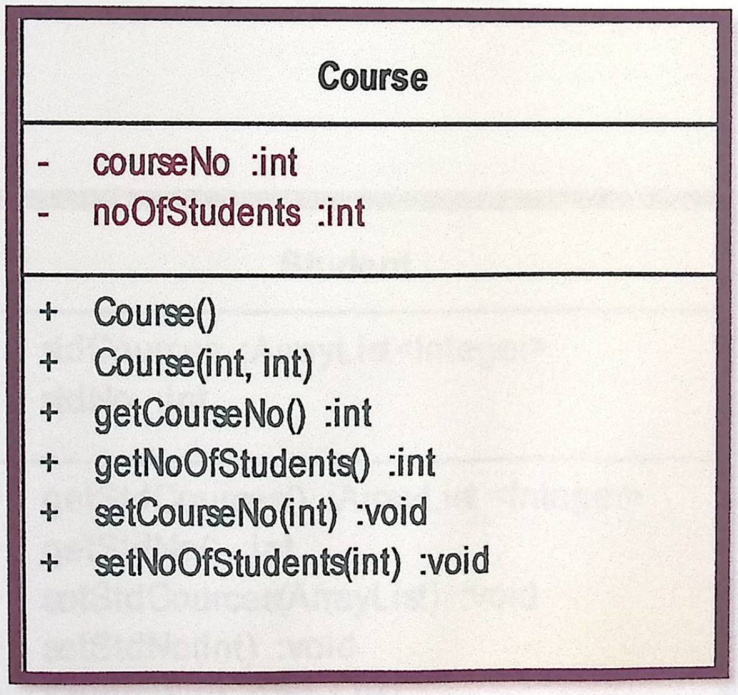

Figure4.4: Course class

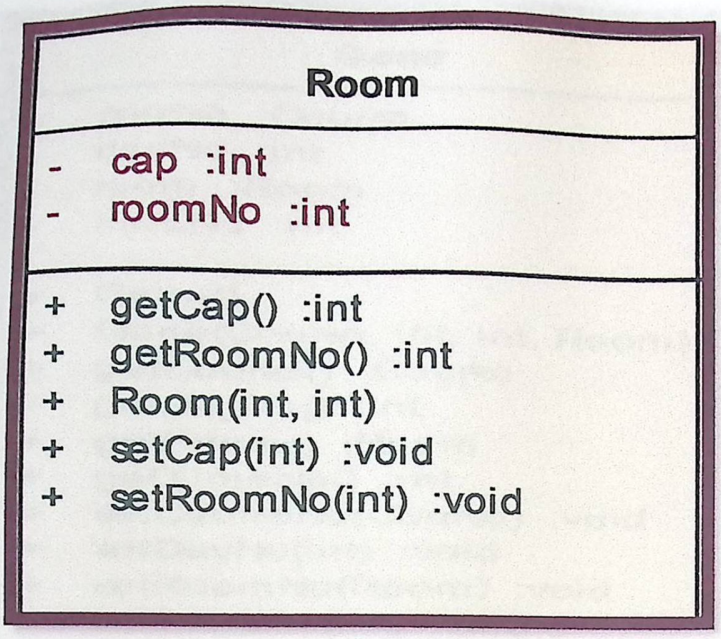

Figure 4.5: Room class

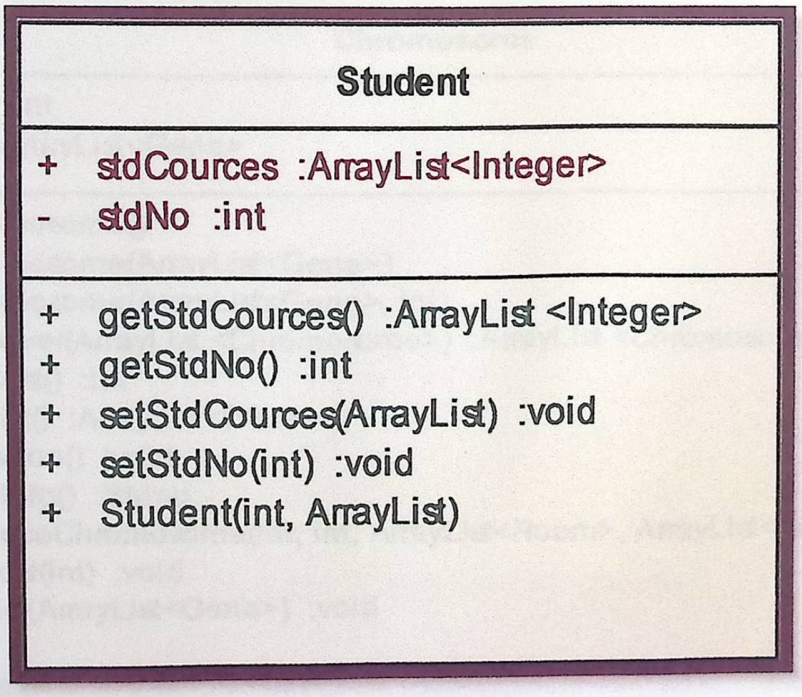

Figure4.6: Student class

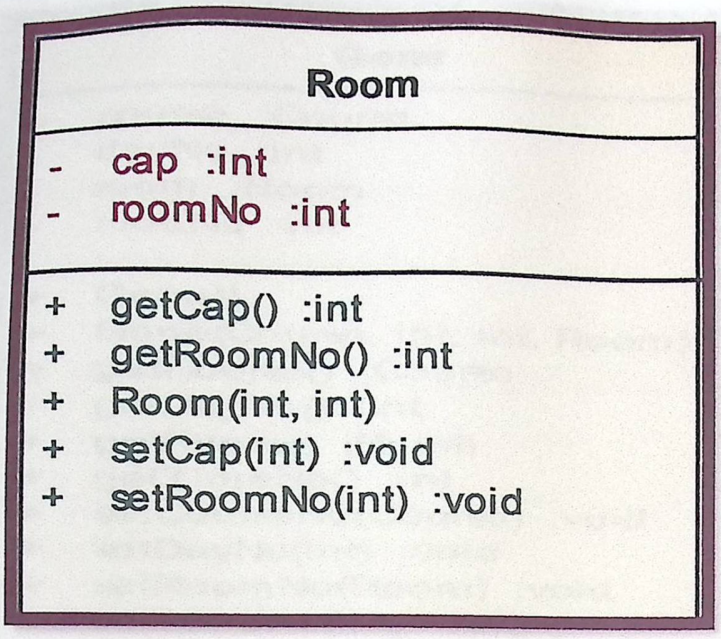

Figure 4.5: Room class

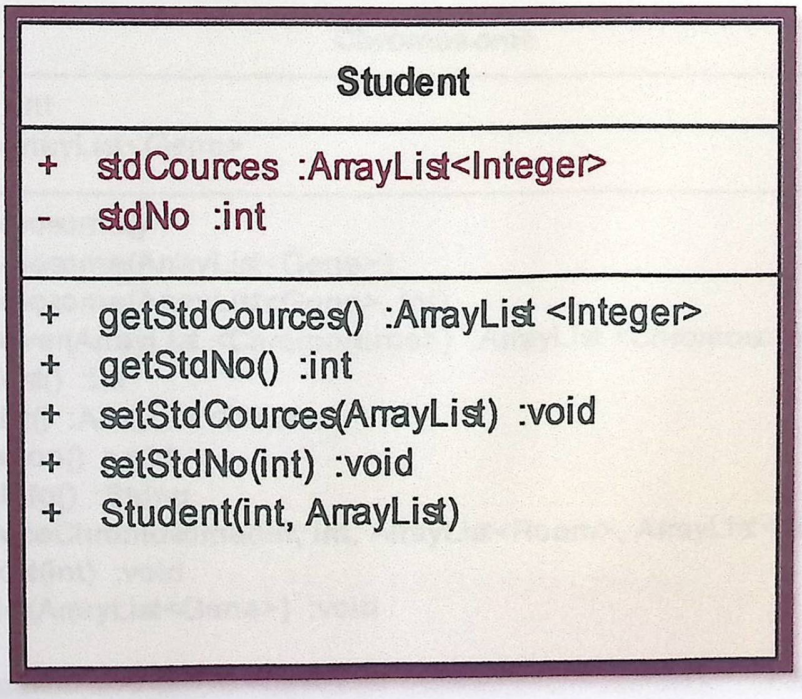

Figure4.6: Student class

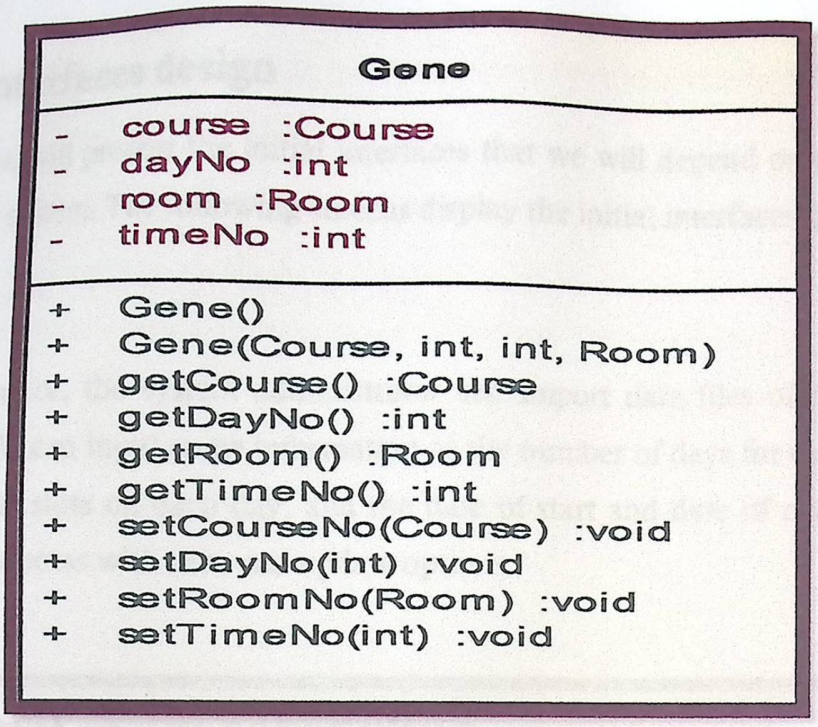

Figure 4.7: Gene class

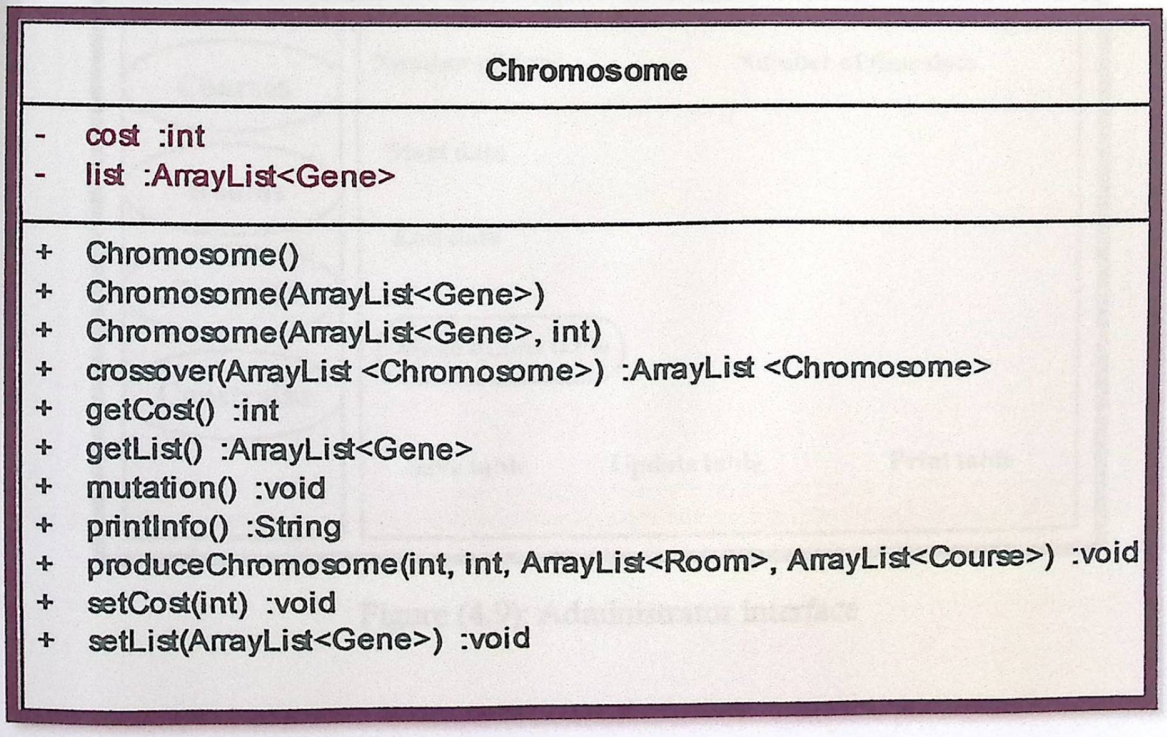

Figure 4.8: Chromosome class

4.6 Initial interfaces design

In this section, we will present the initial interfaces that we will depend on them to design real interfaces for system. The following screens display the initial interfaces of the project:

Administrator

Through this interface, the system administrator can import data files of data and make update on them. He can input some information as the number of days for the exam period, the number of time slots on each day, and the date of start and date of end. Also he can make exam table process with save and update options.

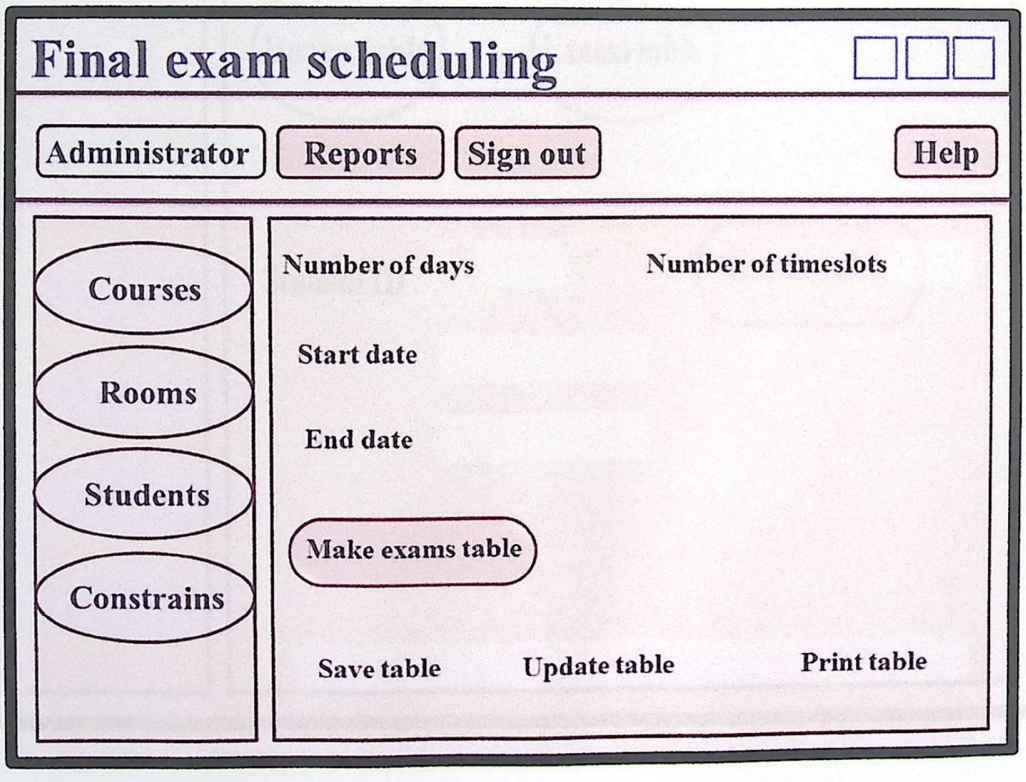

Figure (4.9): Administrator interface

Reports

Through this interface administrator can get final exams table and room table. Also he can search for any student and get his final exams table.

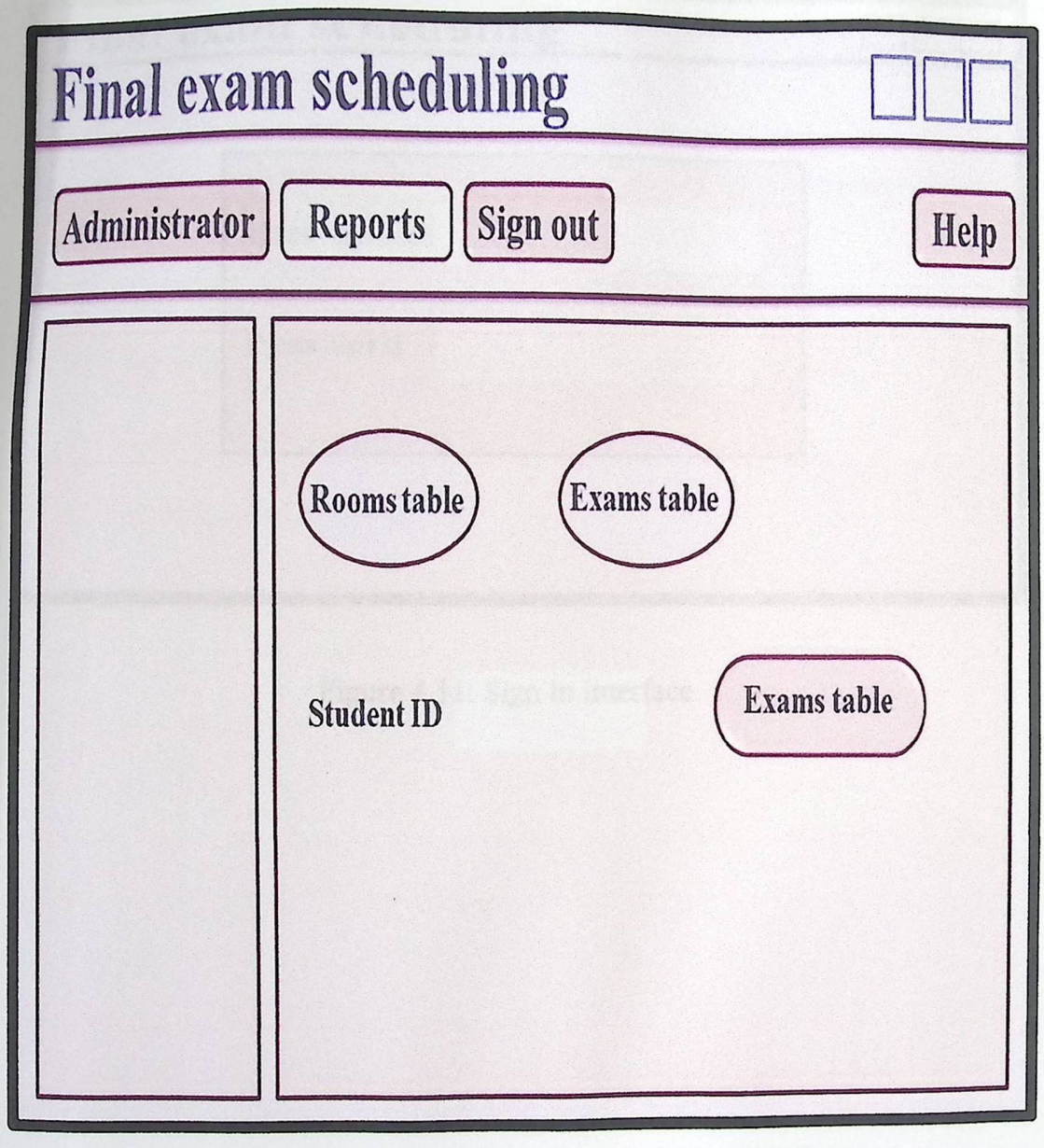

Figure 4.10: Reports interface

Sign in

Through this interface the administrator can enter to system with username and password.

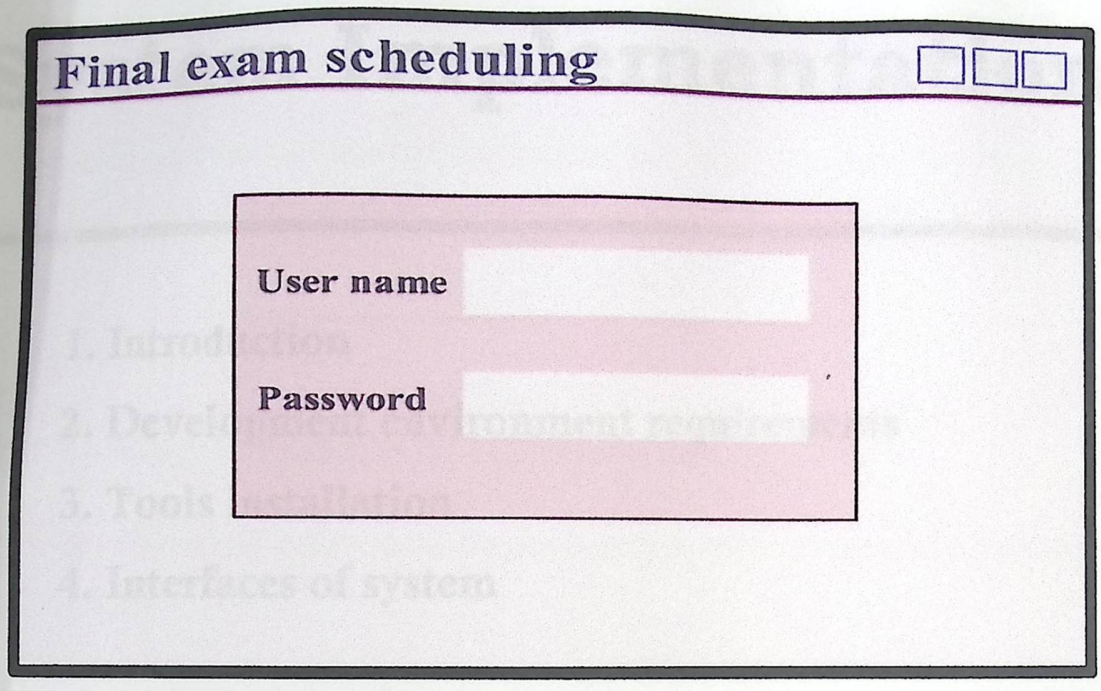

Figure 4.11: Sign in interface

Chapter 5

System Implementation

1. Introduction
2. Development environment requirements
3. Tools installation
4. Interfaces of system

5.1 Introduction

This chapter will explain the environment of implementation phase. This phase will contain the project needs such as the software, hardware, data, and other tools.

Implementation phase converts any project into practical phase, in this section we will provide a detailed explanation about the development environment and required environment.

5.2 Developing environment requirements

This phase describes some details about the environment requirements and programming to develop the project.

1- **Operating system windows 7:** Windows 7 is one of the modern operating systems that produced by Microsoft. It is characterized by reliability, compatibility and security, those characteristics were needed to develop this project as reliability, compatibility and security.

2- **NetBeans:** Sun Microsystems NetBeans produced as open source. It is an integrated development environment (IDE). NetBeans application platform that helps the developers to rapidly create their web, enterprise, desktop, and mobile applications by using the Java platforms, as well as PHP, JavaScript and Ajax, Groovy and Grails, and C/C++, that it improves developers productivity through a smart with faster editor under one of IDE[15]. "*NetBeans IDE 7.1 introduced support for JavaFX 2.0 and NetBeans IDE 7.1.2 introduces support for JavaFX 2.1 by enabling the full compile/debug/profile development cycle for JavaFX 2.1 applications. The release also provides significant Swing GUI Builder enhancements, CSS3 support, and tools for visual debugging of Swing and JavaFX user interfaces. Additional highlights include Git support*

integrated into the IDE, new PHP debugging features, various JavaEE and Maven improvements, and more"[15]. It is available for Windows, Mac, Linux, and Solaris. And it has new important features as:

- **JavaFX**
  It is support the JavaFX 2.1 and it is suitable tool for the all three deployment models: Desktop, Applet, JNLP and it improves application loading experience.
- **Java**
  It has New Visual debugger, it is a gap support in Grid Bag Layout customizer and it comes with improved Java doc highlighting and formatting.
- **Java EE**
  It gives support for Glass Fish 3.1.2 and considered Cluster and Instance deployment support for Glass Fish, and it makes improvements in Java Persistence, Web Services, EJB, Web Logic and more.

3- **Java programming language:** Java is originally language developed by Sun Microsystems, it is versatility, efficiency, platform portability, and the security make it the ideal technology for network computing.
It is an object oriented that enables developers to write software on one platform and run it on virtually any other platform and then it will operate everywhere. And we used NetBeans IDE that provide the java developers with all tools they need to produce a professional java program[4].

4- **JDK 1.7:** JDK is free software from Sun that means Java Development Kit. It used to support java programs, implement the basic set of tools needed to write, the need to compile, test and debug Java applications and applets [14].

5- **Microsoft Excel 2010:** Microsoft Excel is one from Microsoft office group, and it provides the ability to analyze, manage, and sort data in tables. It will used to import data and gives more efficiency and flexibility to accomplish the goals of the project.

5.3 Tools installation:

We used NetBeans program to develop our system and it has some steps to install it before we start use it, so this case will explain installation guidelines. it needs to download the required software that will support it as: JDK6 or JDK5 and JavaFX 2.1.[14]

When you want to download the NetBeans IDE, you can obtain one of several installers, each one of them contains the base IDE and additional tools that you can use to develop any system. And the following installers are available such as: Java SE that Support all standard, Java SE development features ,Java EE that provides tools for developing Java SE and Java EE applications ,C/C++ that supports development in the C, C++, etc, php provides tools for PHP 5.x development, Zend and Symfony Framework support, all of these are a full download options, which contains all the runtimes and technologies available. The following are the steps of installation:

5.4 Interfaces of system:

After the team developed the system that tried to give optimal final exams scheduling, it became ready to use, and users can use any operating system to run it, as Unix, Windows7 and Vista, and it does not need other supportive tools for the implementation. We design suggestion dynamic interfaces where it has more than one active page but we do not use it in this project because we focus on results of system and make comparison, in the following are the most important interfaces of system:

1. **Log-in page:** In figure 5.1 page admin can enter period of time of exams and import courses, rooms, students and constraints files. And he can make exams scheduling process and saves or prints the results.

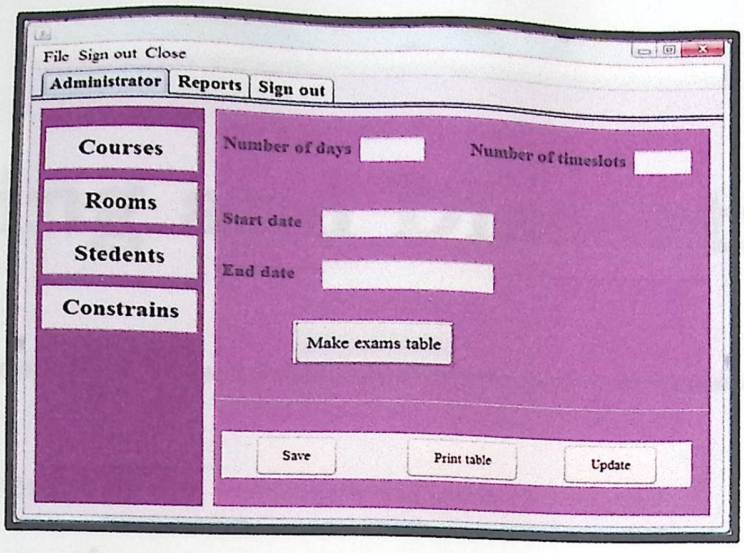

Figure5.1: Admin page

2. **Reports page:** In figure 5.2 page used to get reports about exams table and rooms table, and the user can enter student ID to get his/her exams table.

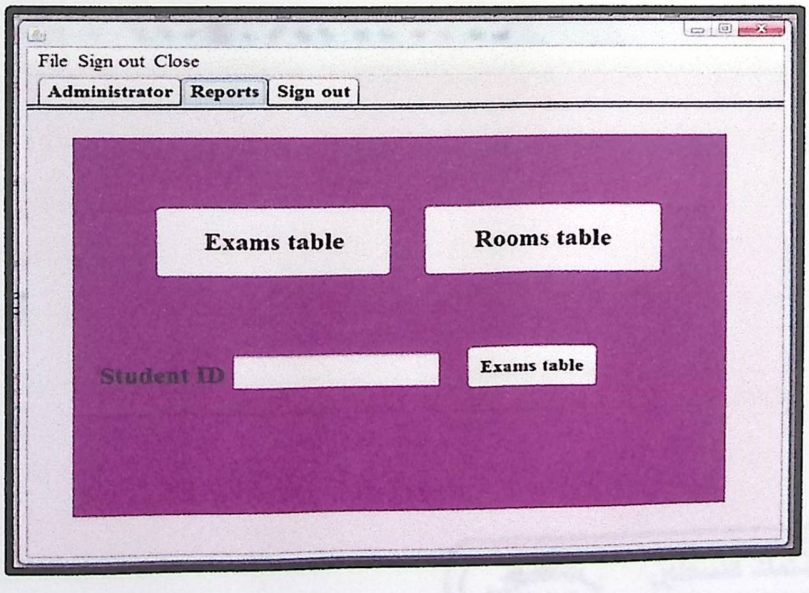

Figure (5.2): Reports page

Chapter 6

Testing and Dissection

1. Introduction
2. Result testing
3. Methodology testing
4. System acceptance testing
5. System testing

6.1 Introduction:

The testing is very important and one of the most important steps during the system life cycle. It comes to make sure that the developers implement their system correctly and work as the expected system requirements, specifications and functions. And the system needs the testing to check the effectiveness and efficiency. In this chapter we will test our system according to applying these levels of testing on it:

1. Results testing.
2. Methodology testing.
3. System Acceptance testing.
4. System testing.

6.2 Results testing

This level is the most important in testing, it depends on analyzing the results of some experiments to get better results and make sure that the system works correctly.

In our system, we tried seven times with repetition, each one to five times because the result depends on random variables that will change in each different experiment, so after each five experiments, we compared the five results with each other, then we chose the best result of them. Each sample has a specific population size and a number of generations, we start from small sample then move to the large in order, that was as the following:

No.	Population size	Number of generation
1	40	100
2	400	100
3	800	100
4	40	1000
5	400	1000
6	800	1000
7	1600	10000

Table 6.1: The class of chromosome information (Chromosome)

6.2.1 Best cost

In the following seven figures, we repeated the experiments five times then we calculate the average and the best cost for each 5 experiments. Then we conclude that, when the number of individuals is small, with a few generations, the result was not enough and the solutions were not good. The following figure shows the results:

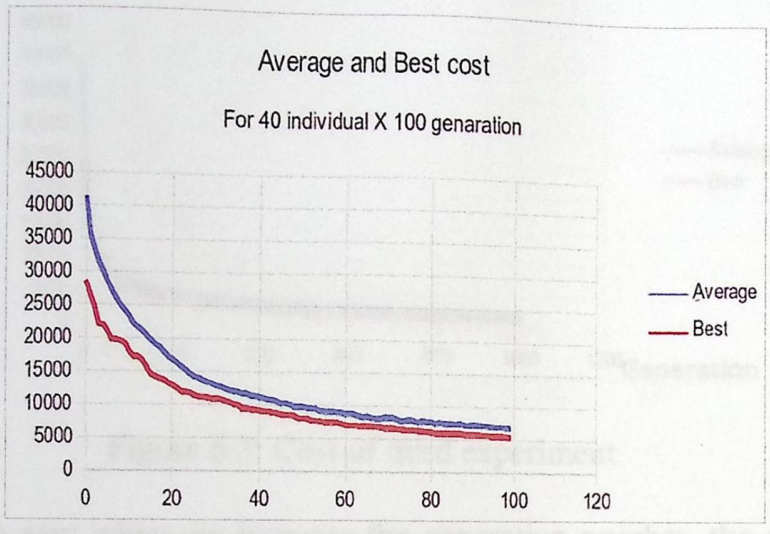

Figure 6.1: Cost of first experiment

Where the cost is the number of hard constrains times, the hard constrains weight, plus the number of soft constrains times, and the soft constrains weight. In the next figure, we increased the population size, and kept the number of generation 100 individuals. The carve slop became less than previous.

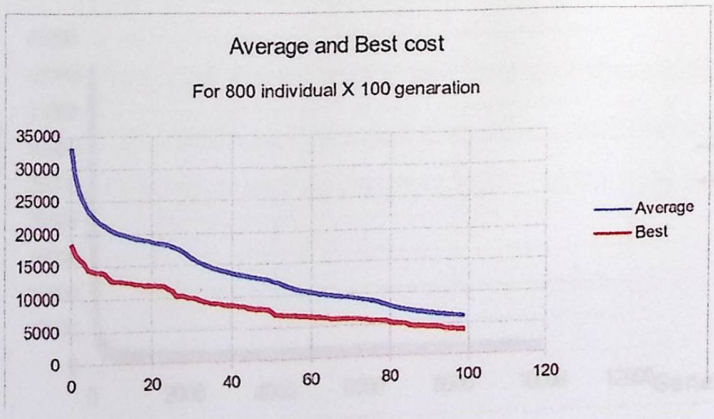

Figure 6.2: Cost of second experiment

When we chose to make a test with small population size but with large generation size, after the generation number 700, there was no need to make more generation since the cost didn't reduced.

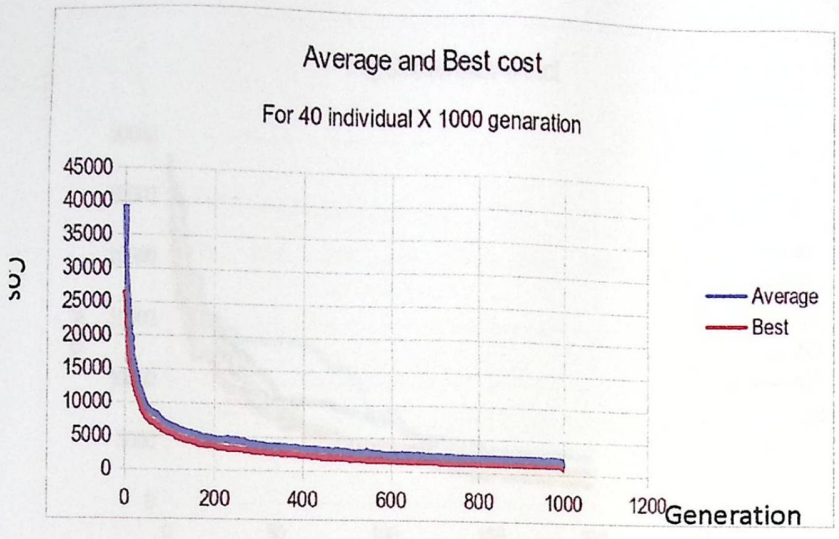

Figure 6.3: Cost of third experiment

In figure 6.4, we can say: when we increase the generation number, the cost became less. And we note that, the cost will not reduce very much if we used a large number of generations more than 700.

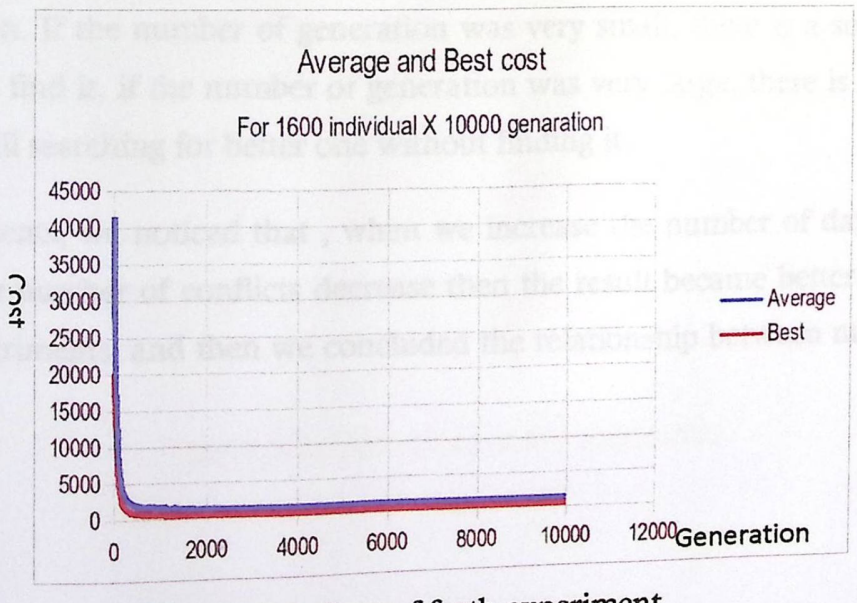

Figure 6.4: Cost of forth experiment

In the figure 6.5, we show the relationship between number of population and cost, each line represent a set of test, the generation number is 200 and the population size was 40, 80, 100, 200, 800 and 1400. Mainly if the population size increases the cost will reduce.

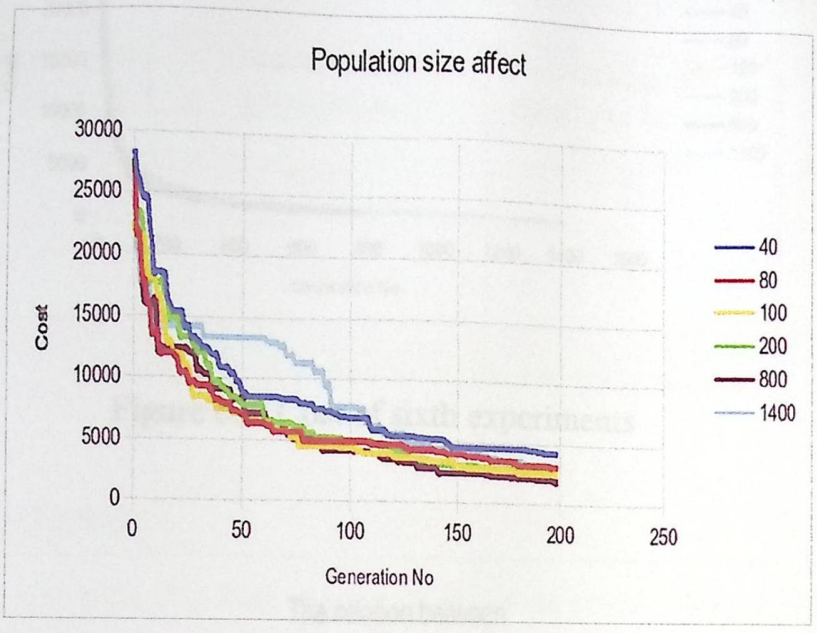

Figure 6.5: Cost of fifth experiments

In figure 6.6, we show the relationship between the number of generation and the cost, when we increase the number of generation, the cost decreases until the algorithm reaches very good solution. If the number of generation was very small, there is a solution but the application didn't find it, if the number of generation was very large, there is a solution but the application still searching for better one without finding it.

Through experiments, we noticed that , when we increase the number of days to hold the examinations, the number of conflicts decrease then the result became better. In figure 6.7 we did eight experiments, and then we concluded the relationship between number of days and the cost.

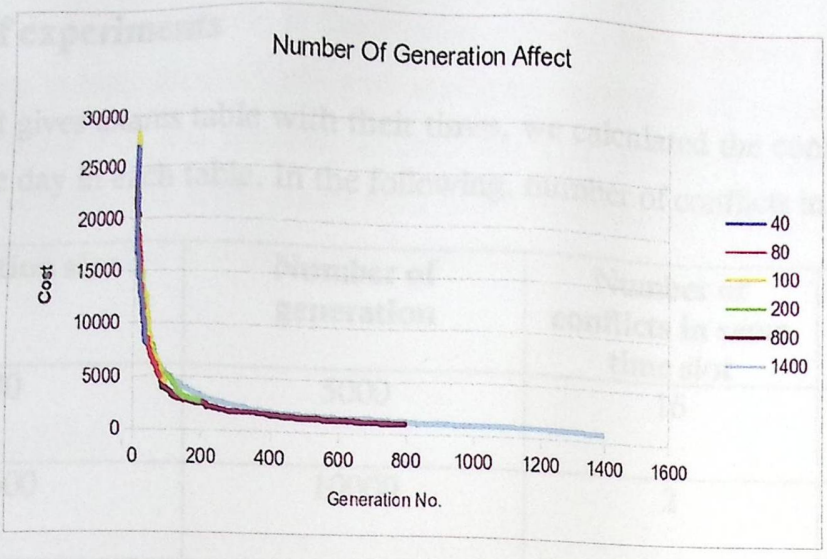

Figure 6.6: Cost of sixth experiments

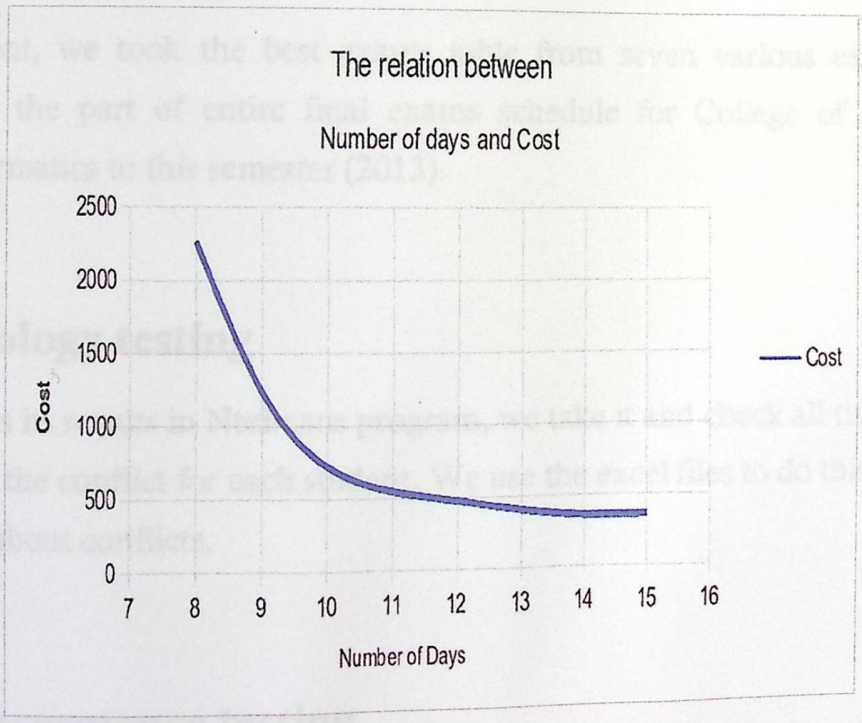

Figure 6.7: Cost of seventh experiments

6.2.2Result of experiments

Each experiment gives exams table with their times, we calculated the conflict in same time slot and the same day in each table. In the following, number of conflicts in two experiment.

No.	Population size	Number of generation	Number of conflicts in same time slot	Number of conflicts in same time day
1	80	5000	16	305
2	1600	10000	2	149

Table 6.2: Number of conflict in some results

6.2.3 Best exam schedule

In our experiment, we took the best exams table from seven various experiments, the following shows the part of entire final exams schedule for College of Administrative Science and Informatics to this semester (2013).

6.3 Methodology testing

After system gives its results in NteBeans program, we take it and check all times of courses and search about the conflict for each student. We use the excel files to do this process and record the notes about conflicts.

6.4 System acceptance testing

The requirements of users is the most important part from building system to get there assent and satisfaction on it, so we must take it as basic point after deploying the system in real world.

In this case, we will check if the system achieve its functional requirements and goals. The system is built for students and academic supervisor of college to get the best final exam scheduling with less number of conflicts. Our system currently reduces number of conflicts with courses and gives entire final exam schedule for college of Administrative Science and Informatics, and it suffers some of problems that appear in traditional way.

6.5 System testing

We have tested the system under several of data, and we extracted the results and found the problems, an there were some of problems in results of system, upon these results and its problems all identified problems have been treated.

Chapter 7

System Maintenance

1. Overview
2. Maintenance plan
3. NetBeans maintenance
4. Handle errors
5. Request maintenance

7.1 Overview

Maintenance phase keeps the system up to date and solves problems that happen after the implementation of the system, and it is important to support the connection between developers and users, it is consisting of four parts, the corrective maintenance: that is used to correct design, logic and code errors and repairs fault, adaptive maintenance: deals with changing the environment of the system as hardware components, and make the system adaptive with any change in the environment, perfective maintenance: that used to enhance the requirement and the functions of system to increase the system's performance by using comments of the users on the system and preventive maintenance: that increase the ability of the system to maintain as adding comments and updating documentation [14].

This chapter will explain suggest maintenance plan for the system and the system deployment.

7.2 Maintenance plan

When a system works in real life, it will face some of errors and problems, there is a possibility for that to happen, and that should be avoided. The users may do not have ability to solve these probability problems, so it will need some procedures from the developer to handle and avoid problems that maybe occur and end users cannot deal with.

This part put suggestions to build maintenance plan for the project, and it contains:

7.2.1 System deployment

The final system is an executive program, and does not have complex steps for installation or other supportive tools to run it, it is easy for the users to deal with it, and deals with many users that can implement it on their devices by take only executive program.

7.2.2 Backup

7.1 Overview

Maintenance phase keeps the system up to date and solves problems that happen after the implementation of the system, and it is important to support the connection between developers and users, it is consisting of four parts, the corrective maintenance: that is used to correct design, logic and code errors and repairs fault, adaptive maintenance: deals with changing the environment of the system as hardware components, and make the system adaptive with any change in the environment, perfective maintenance: that used to enhance the requirement and the functions of system to increase the system's performance by using comments of the users on the system and preventive maintenance: that increase the ability of the system to maintain as adding comments and updating documentation [14].

This chapter will explain suggest maintenance plan for the system and the system deployment.

7.2 Maintenance plan

When a system works in real life, it will face some of errors and problems, there is a probability for that to happen, and that should be avoided. The users may do not have ability to solve these probability problems, so it will need some procedures from the developer to handle and avoid problems that may be occur and end-users cannot deal with.

This part put suggestions to build maintenance plan for the project, and it contains:

7.2.1 System deployments

The final system is an executive program, and does not have complex steps for installation or other supportive tools to run it, it is easy for the uses to deal with it, and deals with many user that can implement it on their devices by take only executive program.

7.2.2 Backup

All the systems suffers from the possibility of destruction in its' hardware or software, so any system should have a full and correct backup plan of the entire system and its units, and any modification to the system should be stored on offline storage.

In this process we preserve copy of the program itself and its file of data and additional components on an offline media, in safe and far place from the work environment. And these backup copies will be used when any failure that will happen.

In our system, during the building process, we created a weekly backup copy weekly to prevent the possibility of losing data, and after the deployment of the system, a copy will be taken of executive program and its modulation files every week at users and us, so we will commend for any user to necessity of weekly backup that will be token by copying system files and save each copy in several places.

7.2.3 Upgrade

The upgrading process makes a new versions of the system, that grant the enhancement on productivity and tasks. They will have new characteristics for the system as add suggestion graphical user interface to system, that we set it in design chapter. This needs to replace the existing system with a new version to facilitate and help users to achieve the best service from system.

7.3 NetBeans maintenance

Using the NetBeans to make improvements on design and the functions of the system as add some enhancement on user interfaces and modify reports that come from processes of the system in the program to increase the performance and reliability of the system.

7.3 Handle errors

It means the expectation and detection of errors then tries to solve them. If any idleness occur then recover executive program from the error that occurred without terminating the application, or in the worst case terminate an affected application after save last modulation

periodically. A run-time error allocate place during the execution of the program on the memory when entering invalid data especially if the memory processes more than one program at the same time, then will lack of sufficient memory to run an application or a memory conflict with another program. Run-time errors can be resolved by minimize number of programs that is using memory.

Some errors happen when viruses attack the system and destroy parts or the entire system, to prevent this problem we must have an anti-virus program on the device, if this problem occurs with existing anti-virus try to do scanning or replace anti-virus to a better one.

7.5 Request maintenance

If system suffer problems that the user cannot solve, he/she can send request a maintenance to the developers of the system to explain the errors and problems of the project to them through a fill model of request that is attached with system.

Chapter 8

Conclusion

1. Introduction
2. Conclusion
3. Recommendations

8.1 Introduction

This chapter will list a some of conclusions and recommendations, that we have reached, and that we can use in future works.

8.2 Conclusion

Final exams scheduling comes to solve traditional process in set final exams in Palestine Polytechnic University, that has some problems as number of students, that may have more than one exam at the same time and this process take more time may be 10 days to set good exams table, but this project apply genetic algorithm to try find the optimal solution, the system will help academic supervisors to make the best final exams scheduling with no more conflict in a less small time.

In our system we reached to get a final exam schedule with minimum of conflicts, we did more than 37 experiments to obtain the best solution with less cost. The last experiment gave less cost and has 2 conflicts in the same time slot and 149 conflicts in the same day and different time slots, this experiment has worked on data of College of Administrative Science and Informatics.

To test the efficiency of our system, we took the final exams table of the college of Administrative Sciences and Informatics of this semester (2012), that have made in traditional way and we calculated number of conflicts. We compared it with results of our system, the following show the comparison of differences between two results:

	Number of conflicts in same time slots	Number of conflicts in same day
Final exam schedule by traditional way	58	144
Final exam schedule by our system (Automated Final Exam Scheduling System using an Genetic Algorithm)	2	149

Table 8.1: comparison between traditional way and automated system

Depending on the previous comparison we can say that the traditional way has cost higher than Automated Final Exam Scheduling System using the Genetic Algorithm, and it gives its result in less time may be 8 hours to 889 student with 1600 size of population and 10000 number of generation for one process.

8.3 Recommendations

This system does not stand at final point, but it has the ability to make enhancement and development in the future, so the teamwork recommended to who work in this filed the following:

Special recommendations for future work:

1. Adding some features as distributing students in rooms.
2. Use professional graphical user interface.
3. Adding additional soft constraints as defining min and max period between times of exams for each student.

General recommendations for future work:

1. Using a modern database management system rather than excel files to input and output data.
2. Compare our results with other techniques.
3. Trying other methods to solve the problems such as heuristic algorithm or another algorithm to generate the first generation.
4. Using other hard and soft constrains to enhance results of this system.
5. Using the system in the first, second and midterm basic exams instead of just final exams.
6. Trying expand the scope of the system to include other colleges, that will make greater and complex space system.
7. Using this system for further research in final exams scheduling problems.

Bibliography

[1] D Bertsims, and J Tsitsiklis, "Simulated Annealing", 1993.

[2] E. Burke and K. Jackson, "Automated university timetabling: The state of the art," *Australia*, 1996.

[3] E. K. Burke, J. P. Newall, "A Multi-Stage Evolutionary Algorithm for the Timetable Problem", in *Department of Computer Science in University of Nottingham*, 1996.

[4] O. Co, "http://www.java.com/en/download/faq/whatis_java.xml", *What is Java technology and why do I need it?*, Last accessed 06 Jun 2012.

[5] D. corne, H. Fag, and C. Mellish, "Solving the modular exam scheduling using a genetic algorithm", in *DAI research*.

[6] M. Keet, "http://www.meteck.org/gaover.html", (2002), *Genetic Algorithms An overview*, Last accessed 06 Jun 2012.

[7] M. Malkawi1, M. A.-Hassan, and O. A,-H. Hassan, "A New Exam Scheduling Algorithm Using Graph Coloring", in *The International Arab Journal of Information Technology*, 2008.

[8] N. Mansor and, M. Timahy, "Stochastic Search Algorithm For Exam Scheduling", 2007.

[9] L. Merlot and N. Boland, "A Hybrid Algorithm for the Examination Timetabling Problem", in *The university of Melbourn, Australia* 2001.

[10] M. Oprea, " A Multi-Agent System for University Course Timetable Scheduling", in *International Journal of Computers, Communications and Control*, 2007.

[11] N. Pillay and W. Banzhaf, "A study of heuristic combinations for hyper heuristic systems for the incapacitated examination timetabling problem", in *European Journal of Operational Research,* 2008.

[12] A. Salwani and A. Hamdan, "a Hybrid Approach for University Course Timetabling", 2008.

[13] k. Sirin, H. Mayada and M. Musa, "Exam Scheduling System", in *PETRA University*, 2007.

[14] Stafford, "Software Maintenance As Part of the Software Life Cycle", in *Department of Computer Science Tufts University*, 2003.

[15] Tinu, "http://netbeans.org/community/releases/71/", (2012), *NetBeans IDE 7.1.2 Release Information*, Last accessed 06 Jan 2012.

Appendices

Appendix

A.1 Interview

بداية أستاذة ورود نود أن نوضح لك هدف المقابلة التي نعقدها معك ،وهو جمع بعض المعلومات عن كيفية وضعكم لبرنامج الامتحانات النهائية في الكلية وعن الأمور التي تأخذونها في عين الاعتبار عند وضعكم للبرنامج .

س١: هل تعتمدون كليا على الطريقة اليدوية في وضع برنامج الامتحانات؟

ج: نعم،يتم جدولة المساقات المطروحة في أي فصل دراسي باستخدام الطريقة اليدوية والتي تعتمد على بناء مجموعات من المساقات لا يوجد تعارض في تسجيل الطالب لها، ويتم عمل خمس عشرة مجموعة بناء على عدد الأيام المتاحة لانعقاد الامتحانات النهائية.

س٢: كيف تقومون بوضع البرنامج ، ما الآلية التي تتبعونها في توزيع الامتحانات على الأيام المحددة لذلك؟

ج: يتم جدولة المساقات لمطروحة حسب:

١. المساقات العامة المشتركة على مستوى الجامعة وهي اللغة الانجليزية، تاريخ فلسطين، الحاسوب وأساسيات البرمجة، اللغة العربية ، الثقافة الإسلامية، تفاضل وتكامل، هذه الفئة من المساقات يتم تعيين موعد معين لانعقادها.
٢. يتم بناء مجموعات من الامتحانات من المتوقع عدم وجود تعارض بينها، ويتم فحص هذه المجموعات للتأكد من وجود أقل عدد من التعارض (بالتجربة والخطأ أحيانا).
٣. يتم وضع تواريخ "أيام" للامتحانات بحيث يكون كالآتي:

- معظم مساقات الطلبة الخريجين والمسجلين لمساق مشروع التخرج تأتي في بداية اسبوع الامتحانات الأول وذلك ليتسنى لهم التحضير للمناقشة.
- يتم أخذ بعين الاعتبار المساقات ذات عدد الشعب الكبيرة لتعقد بالبداية أيضا من ناحية المدرس الذي لديه أكثر من شعبة لمساق معين يكون بالبداية.
- قدر الإمكان يتم توزيع المساقات حسب الخطة التي تقدمها الكلية للطلبة بحيث تكون مجموعة المساقات المسجلة على مدار خمسة أيام.

س٣: كيف تتعاملون مع مشكلة التعارض التي تظهر في برامج الطلبة ؟

ج: يتم بحث إمكانية تغيير موعد الامتحان الذي يسبب التعارض ويتم فحصه في جميع الأيام (المجموعات) لحين وجود يوم يقل فيه التعارض.

س٤: هل تتعرضكم مشاكل أخرى غير التعاراضات عند الطلبة ،وما هي؟

ج: المشاكل التي تعترض عملية جدولة المساقات:

١. ضغط الامتحانات لمجموعة من الطلبة.
٢. التباعد الكبير بين الامتحانات لمجموعة.

س٥: كم يستغرق وقت وضع البرنامج النهائي؟

ج: الوقت المستغرق بصورة متفرقة لبناء المجموعات وتحديد تعارض الطلبة أسبوع تقريبا، بالإضافة إلى أسبوع آخر في بناء برنامج القاعات ولمراقبة.

س٦: هل تجدون أن هذه العملية مرهقة لكم؟

ج: العملية مرهقة جدا، حيث انه يتم الفحص باستخدام أرقام المساقات أحيانا يحدث خطأ في إدخال رقم معين لمساق فتظهر النتائج لمساق آخر ، مما يؤدي إلى إعادة عملية الفحص بعد اكتشاف الخلل.

س٧: ما الأمور والشروط التي تأخذوها في عين الاعتبار عند وضع البرنامج؟

ج: يتم الأخذ بعين الاعتبار مساقات الطلبة المنتظمين بالتسجيل حسب الخطة الأكاديمية للتخصص بالإضافة إلى المتطلبات السابقة لبعض المساقات وبعض الاستثناءات التي ترد من قبل رؤساء الدوائر والمتعلقة إما بسفر المدرسين أو خروجهم في إجازات وما شابه.

س٨: مواعيد الامتحانات في البرنامج تعني من غير الطلبة إذا كان هناك آخرين يعنيهم المواعيد؟

ج: مواعيد الامتحانات تعني الطلبة،المدرسين والإدارة.

س٩: هل تتعرضون إلى شكاوى من الطلبة على البرنامج بعد تنزيله للطلبة؟

ج:نعم ، ويتم تسجيل هذه الاعتراضات ويتم عرضها على العميد لأخذ الإجراء المناسب إذا كانت الاعتراضات متعلقة بضغط امتحان لدى مجموعة من الطلبة (العدد كبير) يتم تغيير موعد الامتحان. وإذا كان هناك طالب لديه (٣) امتحانات بنفس ليوم يتم تأجيل احدها حسب قانون الجامعة.

س١٠: ما هي الشكاوى التي تأتي إليكم؟

ج: - ضغط امتحانات

- فراغ بين امتحانات
- تعارض مساقين في نفس اليوم
- ممكن ا نياتي طلب لديه أربعة امتحانات أو ثلاثة خلال يومين أو لديه ثلاثة امتحانات في نفس اليوم ،هنا يتم بحث حالاتهم.

س١١: هل تخرجون تقارير ؟ولمن تكون هذه التقارير؟وما فحواها؟

ج: في حين طلبها فقط، أسماء الطلبة الذين لديهم امتحانين أو أكثر.

س١٢: هل تعتقدون أن وجود برنامج محوسب يقوم بوضع البرنامج سيكون حلا أفضل من الطريقة اليدوية؟

ج: بكل تأكيد،أي برنامج سيحقق العملية.

س١٣: إذا توفر برنامج محوسب يقوم بهذه العملية هل تؤيدين استخدامه؟

ج: نعم سأقوم باستخدامه، ويتم بحث إذا كان بحاجة إلى تطوير.

A.2 Questionnaire

استبيان لجمع معلومات من الطلبة عن مواصفات برنامج الامتحانات النهائي الذي يريدون

يهدف هذا الاستبيان إلى الحصول على ما يطلبه الطالب أن يتوفر في برنامج الامتحانات النهائية ، وكيف يرغب أن يكون هذا البرنامج الموضوع؟ وكذلك رصد انطباع الطلبة عن برامجهم للامتحانات النهائية في كل فصل دراسي، لذا يرجى الإجابة على الأسئلة التالية حسب وجهة نظرك وحسب ما ترغب من مواصفات للبرنامج ، حتى يمكن الاستفادة من أرائك في وضع برنامج يأخذ بعين الاعتبار ما يحتاجه ويرغب به الطالب.

التخصص: .....

السنة :.....

١. هل أنت راض عن برنامج الامتحانات النهائية في نهاية كل فصل دراسي؟
- ☐ ممتاز
- ☐ جيد جدا
- ☐ مقبول نوعا ما
- ☐ غير مقبول

٢. ما رأيك في النظام اليدوي الذي تتبعه جامعة بوليتكنك فلسطين في وضع برنامج الامتحانات النهائية ؟
- ☐ ممتاز
- ☐ جيد جدا
- ☐ مقبول نوعا ما
- ☐ غير مقبول نهائيا
- ☐ محايد

٣. هل سبق وان واجهتك مشاكل بسبب البرنامج ؟

☐ أحيانا
☐ دائما
☐ لم يحدث

٤. ما هي المشاكل التي واجهتها؟

.....
.....
.....
.....
.....

٥. هل حاولت حلها؟
☐ نعم
☐ لا

٦. إذا نعم ،هل تم حل المشاكل التي واجهتك في برنامجك للامتحانات؟
☐ نعم
☐ لا
☐ لم أحاول حلها
☐ تعقدت أكثر

٧. كيف قمت بحل المشكلة ؟
☐ التوجه للمشرف الأكاديمي وتقديم اعتراض
☐ تقديم اعتراض إلى رئيس الدائرة وعميد الكلية
☐ التنصل من تقديم الامتحان في موعده المقرر وتقديمه كامتحان اكمال

٨. هل طبيعة برنامجك للامتحانات النهائية يؤثر عليك؟
☐ نعم
☐ لا
☐ محايد

٩. إذا نعم ،كيف يؤثر عليك البرنامج الموضوع لامتحاناتك؟
☐ يؤثر على تحصيلي الدراسي بشكل كبير
☐ يسبب لي الضغط والاكتئاب
☐ يؤثر على تنظيمي لوقت الدراسة
☐ يؤثر قليلا على تحصيلي في الامتحانات
☐ لا أجد وقتا كافيا للدراسة

☐ يجعلني أهمل الدراسة على الامتحانات
☐ جميع ما ذكر
☐ تأثيرات أخرى كـ .....

١٠. برأيك ما هو سبب المشاكل التي واجهتك أو الصعوبات في برنامجك النهائي؟
☐ طريقة وضع البرنامج كونها يدوية
☐ عدم اهتمام واضعي البرنامج بما يحتاجه الطالب
☐ أمر طبيعي أن يحدث مشاكل
☐ أسباب أخرى هي:
.....
.....

١١. هل ترغب أن تتغير طريقة وضع البرنامج وتصبح حاسوبية؟
☐ نعم
☐ لا
☐ محايد

١٢. هل تعتقد أن المشاكل ستقل أو تختفي إذا تم تغيير آلية وضع البرنامج وأصبحت حاسوبية؟
☐ نعم
☐ لا
☐ محايد

١٣. هل لديك طلبات تحب أن تتوفر في برنامجك لنهائي؟
☐ نعم
☐ لا
☐ محايد

١٤. ما هي طلباتك التي تود أن تتوفر في برنامجك النهائي وتتأمل أن تُطبق؟
.....
.....
.....

١٥. ملاحظـات
.....
.....
.....

شكرا لمساهمتكم معنا

B.1 Final exam schedule for college of Engineering and Technology from one experiment on our system

Course number	Day Number	Time slot Number
4002	0	0
4786	0	0
4787	0	0
4749	0	0
4567	0	0
4267	0	1
5297	0	1
4570	0	1
4001	0	1
4246	1	0
4614	1	0
5376	1	0
4735	1	0
5054	1	0
4642	1	0
4323	1	1
4559	1	1
4731	1	1
4551	1	1
4752	1	1
4541	1	1
5064	2	0
5015	2	0
4741	2	0
4556	2	1
4503	2	1
4256	2	1
5014	3	0
4543	3	0
4284	3	0
5052	3	0
5018	3	1

4896	3	**1**
4502	3	1
5292	3	1
4617	4	0
5038	4	0
4643	4	0
4755	4	0
5295	4	0
4257	4	0
5348	4	0
5069	4	0
4544	4	0
4780	4	0
4320	4	1
4576	4	1