Fayoum University
Faculty of Computers and Information
Computer Science Department and Information System Department
2014 / 2015

Credit Hours Online Registration System for El-Fayoum University

Phase 1:

For Faculty of Computers and Information

Supervised By:

Dr. Hala Hameed.

Faculty of Computers and Information , Fayoum University

Team Members:

1. Hanem Ahmed. (CS)
2. Nagwa Gamal. (IS)
3. Yousra Hamdy. (CS)
4. Enas Ragab. (IS)
5. Mona Mohamed. (IS)
6. Alaa Ahmed. (IS)

Table of contents

Acknowledgment……………………………………………………………………6

Abstract …………………………………………………………………………..7

1. Introduction………………………………………………………………….. 8
1.1 Motivation ………………………………………………………………….. 9
1.2 Problem Definition ……………………………………………………. ……... 10
1.3 System Objectives ………………………………………………………….. 10
1.4 System Features ………………………………………………………….. 10
1.5 Stakeholders ……………………………………………………................ 11
1.6 Feasibility study ……………………………………………………………… .11
1.7 System Overview ……………………………………………................. 12
1.8 Document Organization ………………………………………………… 13
1.9 Summary ……………………………………………………………………… 13

2. Background (credit hours) ……………………………………………… 14

2.1Credit hour's system Definition……………………………………………….. 15

2.2 Advantages of Credit hours system…………………………………………..... 15

2.3 Acceptance requirements and system study………………………………... 15

2.4 Grades and estimates ……………………………………………….…… 16

2.4.1 Study plan……………………………………………………….….. 17

2.5 Course Coding……………………………………………………………..... 18

2.6 Courses distribution……………………………………………………….... 19

2.6.1 College Requirements…………………………………………….. 19

2.6.2 Department requirements………………………………………...   23

2.7 Numeric and symbolic connotations of degrees and estimates …………... 26

2.7.1Cumulative average (rate) calculation………………………………… 27

3. Methodology (phase I)……………………………………………….. 28

3.1 Requirements Analysis…………………………………………………….29

3.1.1 Current system…………………………………………………….....30

3.1.2SWOT analysis on current system ………………………………… 32

3.1.3 New System……………………………………………………….35

3.2 Requirements Specifications…………………………………………….... 37

3.2.1 User requirement specification …………………………………… 38

3.2.2 System Requirement Specification …………………………….... 39

4. Analysis……………………………………………………………..41

4.1 OVERVIEW……………………………………………………………… 42

4.1.1 Systems Development Life Cycle (SDLC) Methodology…………… 42

4.1.2 System Development Phases……………………………………… 44

4.1.3 Overview of System Analysis and Design…………………………... 47

4.2 System Analysis ………………………………………………….…. 48

4.2.1 System Requirements……………………………………………... 48

4.2.1.1 Functional Requirements……………………………………..... 48

4.2.1.2 Non-Functional Requirements………………………………..... 50

4.2.2.3 Functions of system…………………………………………..... 51

4.2.2.4 Use case diagrams…………………………………...…... 54

4.2.2.5 Sequence analysis diagrams…………………………...… 73

4.2.2.6 Flow chart analysis diagrams…………………………...…….. 82

5. Architecture and Design……………………………………………..90

5.1 Overview………………………………………………………………..91.

5.2Design of system……………………………………………………92

5.3 Design of database…………………………………………………100

5.3.1 ERD Diagram …………………………………………………….100

5.3.2 Description of ERD Tables ……………………………………….101

6. Implementation ……………...…………...112

6.1Overview…………………………………………………………113

6.2 Implementation of data base…………………………………………..113

6.2.1 Creating data base tables ……………………………………….....114

6.3 Implementation of website……………………………………………….140

7. Security and Tools……………………………………………………………154

7.1Security…………………………………………………………………155

7.1.1Security in three ways…………………………………………………155

7.1.1.1 Using security browsers. ………………………………………………156

7.1.1.2 Using Three-Tier Architecture……………………………………160

7.1.1.3 Using SQL server authentication…………………………………161

7.2Tools and programming languages…………………………………162

7.2.1 Tools……………………………………………………………...162

7.2.1.1Analysisphase…………………………………………………162

7.2.1.2 Design phase………………………………………………………….162

7.2.1.3 Implementation phase…………………………………………………162

7.2.2 Programming languages used on Web development…………………163

8. Test ……………………………………………………………………164

8.1 Overview……………………………………………………………165

8.2 Test case 1: login process………………………………………………..166

8.3 Test case 2: Registration process……………………………………………168

8.4 Test case 3: View Exam table………………………………………………172

8.5 Test case 4: View result………………………………………………………..173

8.6 Test case 5: View Academic course schedule………………………………174

9. Summery and future work………………………………………………175

9.1 Summary …………………………………………………………………176

9.2 Future work……………………………………………………………….177

Appendix A: ………………………………………………………………178

1. List of Figures………………………………………………………………179
2. List of tables...183
3. List of equations…………………………………………………………183

Appendix B: A,R…. 184

1. Abbreviations 185
2. References 187

ACKNOWLEDGMENT

The success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The special thank goes to our helpful supervisor Dr. Hala Hameed**.** The

Supervision and support he gave truly help in the progression of our graduation project. The co-operation is much indeed appreciated.

Our greatly appreciate to the contribution of our faculty, Faculty of Computer and Information (FCI). Especially Prof. Nabila Hassan, listen to our problems and did her best to satisfy all groups. Also we want to thank Faculty Dean Prof. Hamdy Mahfouz and Prof. Mohamed Alrabeey the Vice Dean for their support.

Our grateful thanks also go to both Dr. Rasha Badry and TA. Mahmoud EL-Badry. A big contribution and hard worked from both of you during the last ten months. The project would be nothing without the enthusiasm, helping and support from both of you.

At Last but not least this project makes us realized the value of working together as a team and as a new experience in working environment, which challenges us every minute.

ABSTRACT

Obliging the student to eliminate the Tuition Duration in a specific period was making many problems. These problems are sometimes causing stress for some students, and causing space of time for other students according to their capabilities in terms of time and abilities of the study.

The proposed system solves these ones by applying the credit hours system by giving the student a share of freedom as he has the right to choose the optimal number of courses that could be studied in the semester, according to his abilities and capabilities.

According to this system every student will have many of the features that have not been able to be owned before. Addition to that making this system online one gives more facilities for the students and for the organization in general, and this is a new and unique in this system in EL-Fayoum University in general and in the College of Computers and Information specially.

Chapter 1

"Introduction"

1. Introduction
1.1 Motivation
1.2 Problem Definition
1.3 System Objectives
1.4 System Features
1.5 Stakeholders
1.6 Feasibility study
1.6 System Overview
1.7 Document Organization
1.8 Summary

Introduction

The studying of credit hours system has recently appeared in the United Kingdom at the end of the nineteenth century and was introduced in the early twentieth century, where it was applied to the higher education system in the US and European universities.

Credit hours systems vary from university to university and college to another and from one curriculum to another decision, as vary the number of hours allowed in each semester, and the number of credit hours per course.

This chapter introduces the proposed system. Firstly, it mentions the motivation and the reason beyond proposing this system. Secondly, it defines the actual problem that is addressed in the proposed system. Then, the system objectives, features, beneficiaries and stakeholders are mentioned. After that, a general overview of the proposed system is explained. Finally, a brief look about the document structure and organization is mentioned.

1.1 Motivation

Motive behind the construction of this system is the development of the education process in the college and then in Fayoum University to promote its students.

Enable the student to choose the material according to its capabilities in terms of time and abilities of the study and take the time that is appropriate for the completion and graduation as he pleases, and this is by coming to make the student focus on the study and who wants to take advantage of the material.

1.2 Problem Definition

In the last time student was obliged to eliminate the Tuition Duration in a specific period, but now enables the student to choose the material according to its capabilities in terms of time and abilities of the study, and this is by coming to make the student focus on the study.

1.3System Objectives

The goal of the system is to deliver automatically all the necessary information to register students for the coming semester.

We build the system for phase one .our system depended on credit hours, online registration for faculty and selected courses as student like. In the future, we want to build a system to be applied for the whole university to serve all the other faculties.

1.4 System Features

This section provides the main features that the proposed system presents to the student, the system features are as follows:

- Is a good information channel for the new comers.
- Students can choice the courses they want according to their abilities and time.
- Through this website the student knows what it duties and its rights.

1.5Stakeholders

The system stakeholders are person, group, or organization that has direct or indirect stake in the proposed system because it can affect or be affected by the project's actions and objectives. The stakeholders are:

- **Students:** they are the persons that use the system In order to attend college by login function then make registration one, they also could search, read, and view result.

- **FCI (Faculty of Computers and Information) :**
  Which organizes the system and serves the students by applying it.

1.6 Feasibility study

A feasibility study is an analysis of the viability of an idea. The feasibility study focuses on helping answer the essential question of “should we proceed with the proposed project idea?

Our project idea consisted of two phases.

1. For Faculty of Computers&& Information.

2. For El-fayoum University.

1.7 System Overview

This section presents the general overview of the proposed system. It demonstrates the main components of the system and how they interact with each other. Figure 1.1 shows the proposed system overview, where the system applying that instead of obliging the student to eliminate the Tuition Duration in a specific period it enables him to choose the material according to its capabilities in terms of time and abilities of the study, and this is by coming to make the student focus on the study.

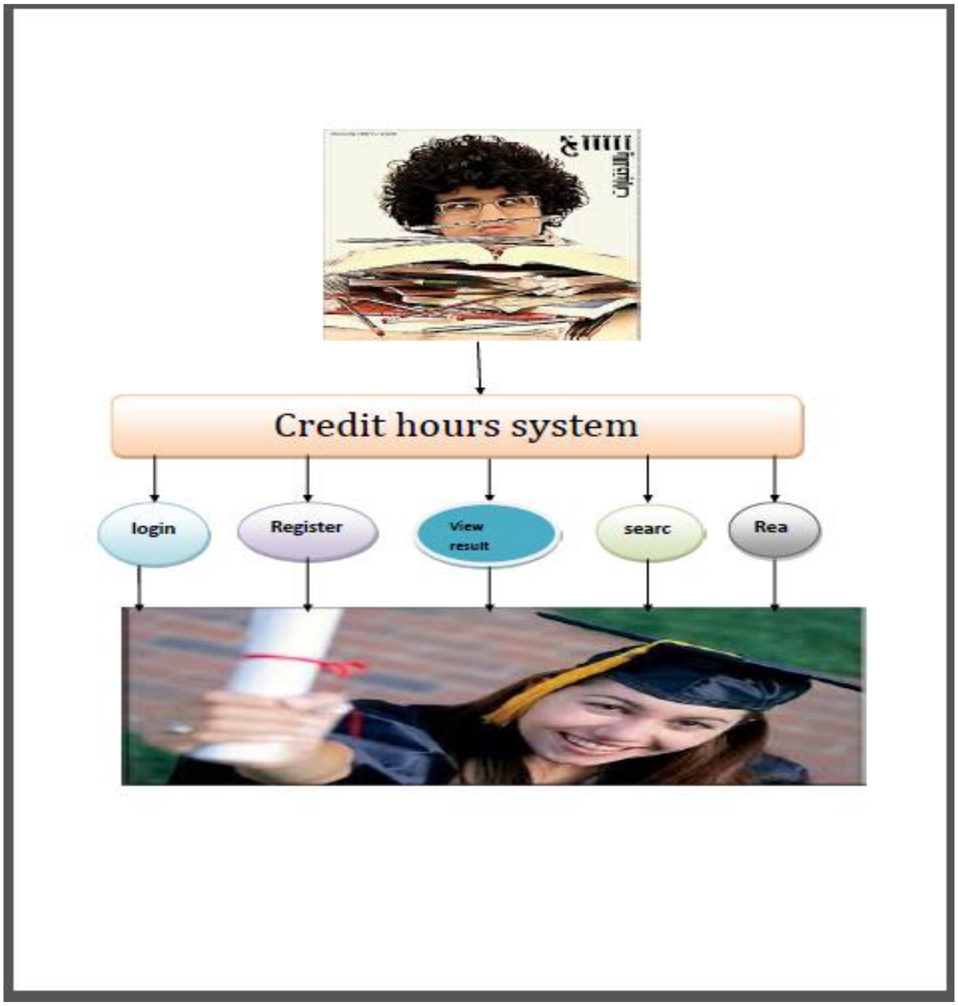


Figure (1.1)

1.8 Document Organization

At this section, this document is briefed. This document consists of (9) chapters that are organized as follows:

-Chapter 1 introduces the proposed system and presents its objectives and features.

- Chapter 2 provides a background about the credit hours technology used in the proposed system.

- Chapter 3 presents the methodology and the requirement specifications of the system.

- Chapter 4 presents the analysis phase of the system.

- Chapter 5 presents the design phase of the system.

- Chapter 6 focuses on implementation phase of the system.

- Chapter 7 focuses on security issues and tools that we have used .

- Chapter 8 describes a set of test case scenarios for the proposed system.

- Chapter 9 concludes the system and lists some future work.

- Appendix A Contains List of Figures and tables that we have used.

- Appendix B Contains Abbreviations and References.

1.9 Summary

In this chapter, the main idea of the system is introduced; system features and objectives are also mentioned. The next chapters will include what the system should do in details and how it work.

Chapter 2

“Background (Credit hours)”

2.1Credit hour’s system Definition.

2.2 Advantages of Credit hours system.

2.3 Acceptance requirements and system study.

2.4 Grades and estimates

2.4.1 Study plan.

2.5 Course Coding.

2.6 Courses distribution.

2.6.1 College Requirements.

2.6.2 Department requirements.

2.7 Numeric and symbolic connotations of degrees and estimates

2.7.1Cumulative average (rate) calculation.

2.1Credit hour’s system Definition

The credit hours system means that “the student studies a set of specific courses with credit hours according to the list, which through it student obtain its university degree of specialization.

2.2 Advantages of Credit hours system

This system achieves the following advantages:

- Freedom of the student in the choice of materials they wish to study each semester with the help of her academic advisor.
- Student Freedom in reducing the number of materials that are being studied in the classroom if he has found conditions so warrant.
- The possibility of increasing the number of student credit hours registered for study in a given semester, if he got a high accumulative rate according to the list.
- The student can study one of the material selected from the last semester, so as to ease the burden of the school on the one hand and to improve the cumulative average of the other hand.
- The student can transfer to any foreign university easily to apply the same system. Calculates to him all the materials which studied and taught successfully. If these materials are taught in the university transferred to it.

2.3 Acceptance requirements and system study

- Acceptance requirements to the faculty of computers and information

El-Fayoum University to obtain a high school or equivalent (specializing

in scientific mathematics) through coordination office

- The school years dividedinto two semestereach semester 17weeks (15-week study) and thentwo weeks tothe endexams ofsemester andtheacademicburden onstudents in the semester between9-18credit hours.

- Fayoum University granted based on faculty Board request the following degrees:

Bachelor degrees in Computers and Information inthe following

Specialists:

- Department of Computer Science.
- Department of Information Systems.
- Department of Information technology.

The studying system followed in the faculty is a system of credit hours in the semester. The duration of the study to obtain a bachelor's degree in computer and information systems is four academic years, at least in accordance with the list Regulations for the universities regulation Law, and achieve this period four academic levels where single-level includes a two semester separate them a holiday of half a year in addition to the optional summer and a maximum duration of the study is fourteen semester.

Semester consists of seventeen weeks distributed as follows:

- **Registration period is one week.**
- **The study period is fourteen weeks.**
- **The exams two weeks.**

2.4 Grades and estimates

The basic measure of newsystem iscredit hours .on next section talking about the distribution of credit hours to satisfy the basic requirements (University requirements, University requirements , Department requirements)

Credit hour consider academic measure unit to determine the weight of

course among other courses and is equivalent to 50 minutes lecture per week and 100 for each Lab or section.

2.4.1 Study plan

- The bachelor of science (B.Sc) program in Computer Science & Information System require 144 credit hours (CH).the distribution of these credit hours according to the proposed categories is as follows:
- 12 CH as University requirements.
- 72CH as c
- 60 CH as Department requirements.

(A) Studying is based entirely on the credit hour system, and divides the academic year into two semesters.

(B) To get Bachelor student must pass successfully one hundred and forty-four credit hours over eight semesters, at least, is divided into four levels of studying.

(C) The studying in the first level and the second is common to all disciplines, and starts at the third level of specialization. And each section must put the qualifying conditions to join to him after the approval of the Faculty Council.

(D) Organizing summer training for first and second level equivalent to 3 credit hours (2 lectures/ 2 practical) during July and August.

2.5 course coding

C5	**C4**	**C3**	**C2**	**C1**

- (c1, c2) ------------→course number inside department.
- (c3,c4)--------------→department number inside college.
- (c5) ------------→course level.

The codes for studying levels

Code	Study level
1	First
2	second
3	third
4	Forth

Symbolic system for scientific departments

Department code	Shortcut scientific name	Department	N
01	G	Basic sciences	1
02	CS	Computer science	2
03	IS	Information system	3
04	IT	Information technology	4

Table (2.1)

2.6 Courses distribution

On following sections we will show the distribution of Courses and their number of credit hours and their prerequisites if exist.

Prerequisites courses show that there are courses must be taken before and that was helpful to us on registration process.

2.6.1 College Requirements:

That includes 72 credit hours (63 hour mandatory +9 hours Optional)

(A) The mandatory requirements of 63 credit hours

Course code	Courses	**Training hours**			Degree			total	Prerequisite requirement
		Credit hours	**lecture**	**lab /section**	Theoretical	Year works	Oral/practical		
20202	هياكل البيانات **Data Structures**	3	3	2 / -	60	15	25	100	**برمجة الحاسبات-1** حسب – 112
30202	**خوارزميات** Algorithms	3	3	- / 2	60	15	25	100	**برمجة الحاسبات-1** **حسب 112**
20402	**تراسل البيانات** **Data Communication**	3	3	- / 2	60	15	25	100	**رياضيات –2** **ريض 113**
201 08	**بحوث العمليات** Operations Research	3	3	- / 2	60	15	25	100	**-**
10109	مبادئ الإدارة Fundamentals of Management	2	2	-/ -	70	20	10	100	-
203 01	مقدمة فى نظم المعلومات	3	3	2 / -	60	15	15	100	-

Course code	Courses	Training hours			Degree			total	Prerequisite requirement
		Credit hours	**lecture**	**lab /section**	Theoretical	Year works	Oral/practical		
	Introduction of Information Systems								
203 02	نظم قواعد البيانات 1 – Database System – 1	3	3	2 / -	60	15	15	100	-
204 03	شبكات الحاسبات – 1 Computer Networks – 1	3	3	2 / -	60	15	15	100	تراسل البيانات تقن 221
204 04	تكنولوجيا الإنترنت Internet Technology	3	3	2 / -	60	15	15	100	-
202 04	نظم التشغيل – 1 Operating Systems – 1	3	3	2 / -	60	15	15	100	برمجة الحاسبات-1 حسب 112
202 05	هندسة البرمجيات– 1	3	3	- / 2	60	15	15	100	مقدمة حاسبات حسب 111

Course code	Courses	**Training hours**			Degree			total	Prerequisite requirement
		Credit hours	**Lecture**	**lab /section**	Theoretical	Year works	Oral/practical		
20202	هياكل البيانات **Data Structures**	3	3	2 / -	60	15	25	100	**برمجة الحاسبات-1** حسب – 112
30202	**خوارزميات** Algorithms	3	3	- / 2	60	15	25	100	**برمجة الحاسبات-1 حسب 112**
20402	**تراسل البيانات**	3	3	- / 2			25		**رياضيات –2**

Course code	Courses	Training hours			Degree			total	Prerequisite requirement
		Credit hours	**Lecture**	**lab /section**	Theoretical	Year works	Oral/practical		
	Data Communication				60	15		100	**ريض 113**
201 08	**بحوث العمليات** Operations Research	3	3	- / 2	60	15	25	100	**-**
1010	مبادئ الإدارة Fundamentals of Management	2	2	-/ -	70	20	10	100	-
203 01	مقدمة فى نظم المعلومات Introduction of Information Systems	3	3	2 / -	60	15	15	100	-
203 02	نظم قواعد البيانات – 1 Database System – 1	3	3	2 / -	60	15	15	100	-
204 03	شبكات الحاسبات – 1 Computer Networks – 1	3	3	2 / -	60	15	15	100	تراسل البيانات تقن 221
204 04	تكنولوجيا الإنترنت Internet Technology	3	3	2 / -	60	15	15	100	-
202 04	نظم التشغيل – 1 Operating Systems – 1	3	3	2 / -	60	15	15	100	برمجة الحاسبات-1 حسب 112
202 05	هندسة البرمجيات– 1	3	3	- / 2	60	15	15	100	مقدمة حاسبات حسب 111

Table (2.2)

(B) The elective requirements 9 credit hours

Course number	Course name	Training hours			Degrees distribution				
		Credit hour	**lecture**	**Lab/ section**	Theoretical	Year work	Oral	Total	Prerequisite
201 03	إحصاءواحتمالات –2 Probability and Statistics – 2	3	3	- / 2	60	15	25	100	**إحصاء واحتمالات–1** احص 121
304 05	عمارة الحاسبات Computer architecture	3	3	2/-	60	15	25	100	**التصميم المنطقى**
	إدارة المشروعات **Projects Management**	3	3	- / 2	60	15	25	100	**-**
204 05	**إشارات ونظم** **Signals and Systems**	3	3	- / 2	60	15	25	100	**رياضيات –2** **113**
	لغات المحاكاة Simulation Languages	3	3	2 / -	60	15	25	100	**نمذجة و المحاكاة** نال 241
20402	**تراسل البيانات**	3	3	- / 2	60	15	25	100	**رياضيات –2** **ريض 113**
204 01 **اساسى**	**صيانة الحاسب** **Computer Maintenance**	3	3	2/-	60	15	25	100	-
	النمذجة و المحاكاة Modeling and Simulation	3	3	- / 2	60	15	25	100	**-**

Table(2.3)

2.6.2. Department programs requirements

A) Mandatorycourses (42 credit hours)

Course number	Course name	**Credit hour**	**Lecture**	**Lab/ section**	**Prerequisite requirement**
302 01	**برمجة الحاسبات –3** **Programming – 3**	3	2	2/-	برمجة الحاسبات -2
	تحليل و تصميم نظم المعلومات – 1 System Analysis and Design – 1	3	2	2/-	-
302 03	مفاهيم لغات الحاسب Concepts of Programming Languages	3	3	- / 2	برمجة – 2 حسب 213 cs
302 04	تنظيم الحاسبات Computer Organization	3	3	- / 2	**تصميم منطقى -** **حسب 221**
303 01	نظم قواعد البيانات–2 Database Systems – 2	3	3	2/-	نظم قواعد البيانات-1
304 07	نظم الرسم بالحاسب–1 Computer Graphics – 1	3	3	2 / -	**برمجة الحاسبات –1** حسب 112
302 06	نظم التشغيل–2 Operating Systems – 2	3	3	2 / -	**نظم التشغيل – 1** **حسب 241**
302 07	هندسة البرمجيات–2 Software Engineering – 2	3	3	2 / -	**هندسة البرمجيات – 1** **حسب - 251**
404 11	الوسائط المتعددة Multimedia	3	3	2 / -	**برمجة الحاسبات – 1** **حسب 112**
302 08	الذكاء الاصطناعي Artificial Intelligence	3	3	2 / -	**برمجة الحاسبات – 1** **حسب 112**
402 10	المترجمات Compilers	3	3	- / 2	**هياكل بيانات** حسب 214
404 12	معالجة الصور–1 Image processing	3	3	2/-	**خوارزميات**
402 11	مشروع Project	6	3	6 / -	**هندسة البرمجيات – 1** **حسب - 251**

Table (2.4)

B) The elective courses (18 credit hours)

Course number	Course name	Training hours							
		Credit hours	lecture	Exercises	Theoretical	Year works	Oral/practical	total	Prerequisite requirement
حسب 423 CS 423 302 10	تنظيم حاسبات متقدم Advanced Computer Organization	3	3	- / 2	60	15	25	100	التصميم المنطقي حسب 221
304 03	مواجهات الحاسبات Computer Interfaces	3	3	- / 2	60	15	25	100	المعالجات الدقيقة تقن 312
304 06	شبكات الحاسبات–2 Computers Networks – 2	3	3	2 / -	60	15	25	100	شبكات الحاسبات – 1 تقن 222
304 10	التعرف على الأنماط Pattern recognition	3	3	- / 2	60	15	25	100	
303 06	تطبيقات الإنترنت Internet Applications	3	3	2 / -	60	15	25	100	تكنولوجيا الإنترنت 223
402 03	معالجة اللغات الطبيعية Natural Languages Processing	3	3	- / 2	60	15	25	100	الذكاء الأصطناعى حسب 361
402 08	المعالجة على التوازي	3	3	2 / -	60	15	25	100	تنظيم الحاسبات

	Parallel Processing								**حسب 322**
302 04	لغة التجميع Assembly Language	3	3	2 / -	60	15	25	100	**المعالجات الدقيقة تقن 312**
404 16	الرؤية بالحاسب Computer Vision	3	3	2 / -	60	15	25	100	خوارزميات حسب 316

Table (2.5)

2.7 Numeric and symbolic connotations of degrees and estimates

The college follow credit hour system which depend on that the basic unit is the course rather than the year and assessment depend on the appreciation in each course with points which determined according to the following table

Percentage of degree	Estimation	Points	Estimation descriptive
90% or more	A+	4.0	Excellent
90%- less than 95	A	3.7	Excellent
85 less than 90	A-	3.4	Excellent
80 less than 85	B+	3.0	Very good
75 less than 80	B	2.8	Very good
70 less than 75	C +	2.3	Good
65 less than 70	C	2.0	Good
60 less than 65	C-	1.7	Acceptable
55 less than 60	D	1.4	Weak
50 less than 55	D-	0	Very weak

Table (2.6)

2.7.1Cumulative average (rate) calculation

You need to calculate your GPA (Grade Point Average, School use GPA's as a means of ranking students based on their academic performance.

The cumulative grade point average of the student account (GPA) as follows:

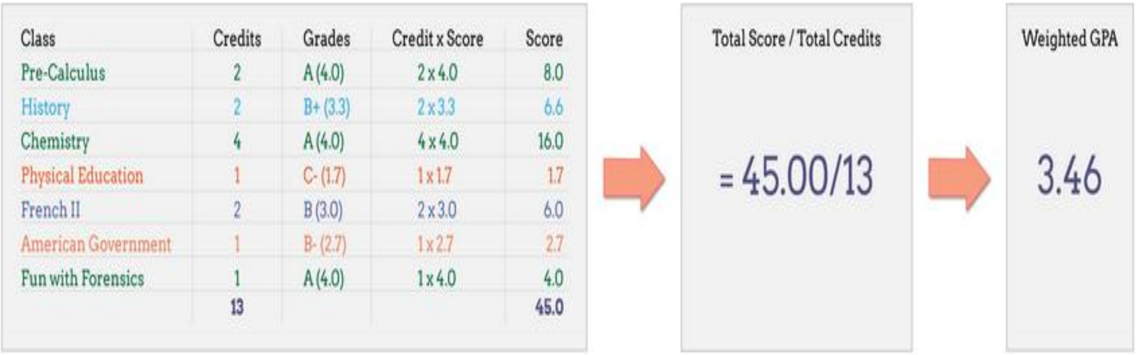
1. The estimate is multiplied by the value of each course (the points outlined in the table) in the number of credit hours for this decision to get the number of points for each Course.

2. Adding points are all courses in which the student record.

3.Dividethe total points on the total hours recorded for the student to get a cumulative grade point average as follows:

GPA = (grades * credits) / credits.   (Equation 4.1)

For Example:



Class	Credits	Grades	Credit x Score	Score
Pre-Calculus	2	A (4.0)	2 x 4.0	8.0
History	2	B+ (3.3)	2 x 3.3	6.6
Chemistry	4	A (4.0)	4 x 4.0	16.0
Physical Education	1	C- (1.7)	1 x 1.7	1.7
French II	2	B (3.0)	2 x 3.0	6.0
American Government	1	B- (2.7)	1 x 2.7	2.7
Fun with Forensics	1	A (4.0)	1 x 4.0	4.0
	13			45.0

Total Score / Total Credits
= 45.00/13
Weighted GPA
3.46

Figure(2.1)

Chapter 3

"Methodology (Phase 1)"

3. Methodology (Phase 1)
3.1 Requirements Analysis
3.1.1 Current system
3.1.2SWOT analysis on current system
3.1.3 New System
3.2 Requirements Specifications
3.2.1 User requirement specification
3.2.2 System Requirement Specification

Methodology (Phase 1)

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

As we see in later chapters that our goal to achieve two phases:

Phase 1

Credit hours online registration system for our faculty**.**

Phase 2

Credit hours online registration system for the university.

On this chapter we will talk on detail about our methodology on phase 1 and to achieve that we must first study the current system for our faculty an analysis its functions to make later process modifications and process improvements.

As known that the first step to create new software is to make Requirements analysis on current system and requirement specifications on new system to achieve specified project goals and services.

3.1 Requirements Analysis

- Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, **analyzing,** documenting, validating and managing software or system requirements.

- Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design

3.1.1 Current system

When we say current system we mean the current system for **FCI** (Faculty of Computer and Information) for Fayoum University.

We study carefully the current system and extract a list of functions on it and then comparing them with existing functions on other systems because these functions help us to know what we need to enter new functions or modify ones to create later the new system.

Current system functions:

1. Dean's Note.

2. A bout Faculty

2.1 Faculty Establishment.

2.2 Faculty Logo.

2.3 Vision and Mission.

2.4 Internal Regulations.

2.5 Study Regulations.

2.6 Time Table.

2.7 Exam Schedule.

2.8 Photo Gallery.

2.9 Code System.

2.10 Students' Guide.

3. Administration

3.1 Dean.

3.2 Vice Dean.

3.3 Faculty Council.

3.4 General Manager.

3.5 Special Units.

3.6 Labs

4. Departments

4.1 Department Of Computer Science.

4.2 Department Of Information Systems.

4.3 Department Of Basic Sciences.

5. News.

6. Projects

6.1 QAU(Quality Assurance Units).

6.1.1 Management

6.1.1.1 Management Team.

6.1.1.2 Rules for Naming Leaders.

6.1.2 A bout QAU.

6.1.3 Projects

6.1.3.1 Alumni Office.

6.1.3.2 Crisis Management Unit.

6.1.3.3 E-Learning.

6.1.4 Strategic Plan.

6.2DSASU (Development of Students' Assessment Systems Unit) .

7. Activities

7.1 Student Activities.

7.2 Workshops.

We will almost simulate some these functions and adding to them new ones, then provide new technology called Credit hours Online Registration System to meet the new needs and services for students we will talk about them on detail later.

We first make SWOT analysis about the current system to know the strength and weakness points on it as fast-moving business environment, which leads to real problems, so radically that the software is effectively useless on future, as we notice that most system now apply the credit hours technology and meet later the new internal regulations for our faculty.

3.1.2 SWOT analysis on current system

A SWOT analysis (alternatively SWOT matrix) is a structured planning method used to evaluate the strengths, weaknesses, opportunities and threats involved in a project or in a business venture.

It involves specifying the objective of the business venture or project and identifying the internal and external factors that are favorable and unfavorable to achieve that objective.

The degree to which the internal environment of the firm matches with the external environment is expressed by the concept of fit. Asfigure(3.1).

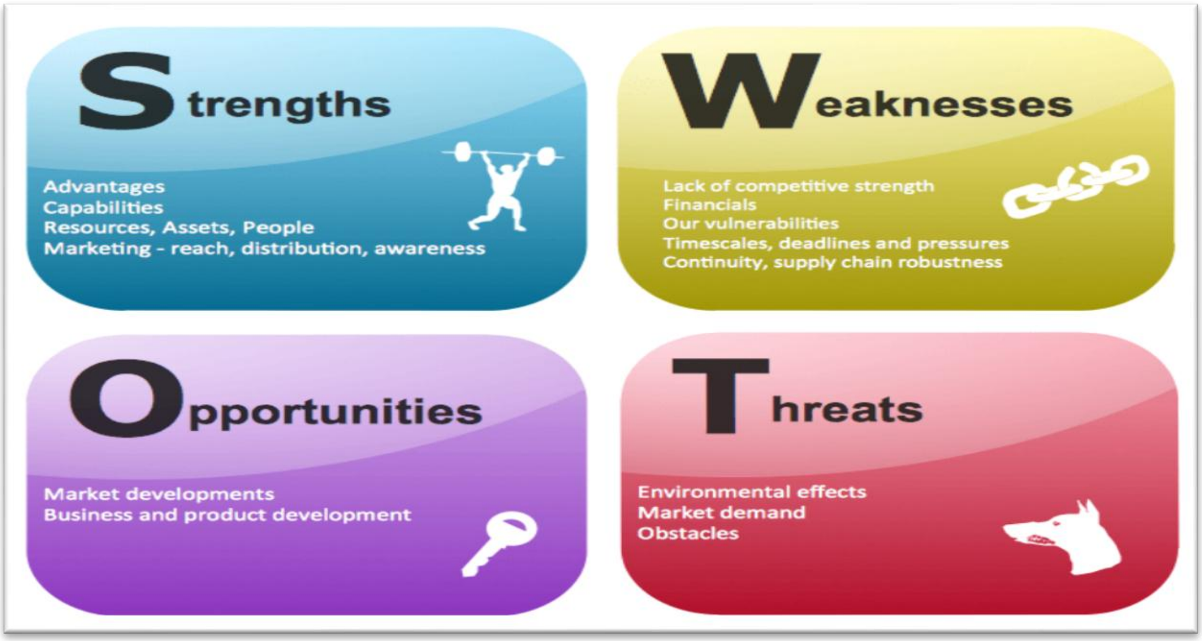

Strengths
Advantages
Capabilities
Resources, Assets, People
Marketing - reach, distribution, awareness
Weaknesses
Lack of competitive strength
Financials
Our vulnerabilities
Timescales, deadlines and pressures
Continuity, supply chain robustness
Opportunities
Market developments
Business and product development
Threats
Environmental effects
Market demand
Obstacles

Figure (3.1)

- **Strengths**: characteristics of the business or project that give it an advantage over others.
- **Weaknesses**: characteristics that place the business or project at a disadvantage relative to others.
- **Opportunities**: elements that the project could exploit to its advantage.
- **Threats**: elements in the environment that could cause trouble for the business or project.

Identification of SWOTis important because they can inform later steps in planning to achieve the objective.

Users of SWOT analysis need to ask and answer questions that generate meaningful information for each category (strengths, weaknesses, opportunities, and threats) to make the analysis useful and find their competitive advantage.

1. Strength points:

- Provide many functions that are necessary for students and staff.
- provide news up to date such as activities, studying and exams tables, champions, prizes, events, results,…………
- The current system is available on two languages (Arabic and English)
- Pictures are suitable to events.
- Fonts and Colors are suitable.

2. Weakness points:

- Technology that applied by the time will be not suitable.
- Not deliver automatically all the necessary information to register students for the coming semester.
- Don't apply the credit hours technology that becomes later popular on many faculties.
- Students cannot pay the fees online.

3. Opportunity points:

- Provide technology that is up to date.
- Apply credit hours system on the current system to make students comfortable and have a better chance to obtain the desired classes and navigation through the available courses.
- Provide more functions and services that we will apply later on new system.

3. Threat points:

- By the time the current system "without credit hours" will be not suitable for the new challenges.
- Students will be annoyed when using current system because it does not provide services that they want.
- The current system will be different than other faculties if they apply the credit hours system and that will make them is not advanced.

3.1.3New System

We will make new system for our faculty to improve the current system and make it acceptable to student's needs and fast moving environment business, also we will enhance it with new system technology called credit hours system.

This new system is called (Credit hours Online Registration System for

El-Fayoum University) .

System Overview:

The goal of the system we are designing to:

- Deliver automatically all the necessary information to register students for the coming semester.
- The proposed system overview isapplying that instead of obliging the student to eliminate the Tuition Duration in a specific period it enables him to choose the material according to its capabilities in terms of time and abilities of the study, and this is by coming to make the student focus on the study.

Aim of this system:

- We want to build a system to be applied for the whole of the university to serve all the other faculties.
- To be a good information channel for the new comer's students.
- Students can choice the courses they want according to their abilities and time.
- Through this website the student knows what it duties and its rights.

Main services of this system:

The system provides great services to the students:

- It makes it possible to register for the coming semesters without physically being in Paris.
- It takes less time to conclude the registration process.
- Students will pay the fees online and complete their registration(future work chapter 8).
- It will provide a better chance to obtain the desired courses and navigation throw the available courses.

As we see later that when we analysis the functions of that system we will take on our mind the functions of the current system so we list under the main functions of new system.

Main functions of new system:

1. A bout FCI.
2. Academics.
3. Prospective students.
4. Current students.
5. Resources of faculty.
6. Graduation Projects.
7. Researches.

These functions include other sub functions that we will take about them in detail on next chapters.

3.2 Requirements Specifications

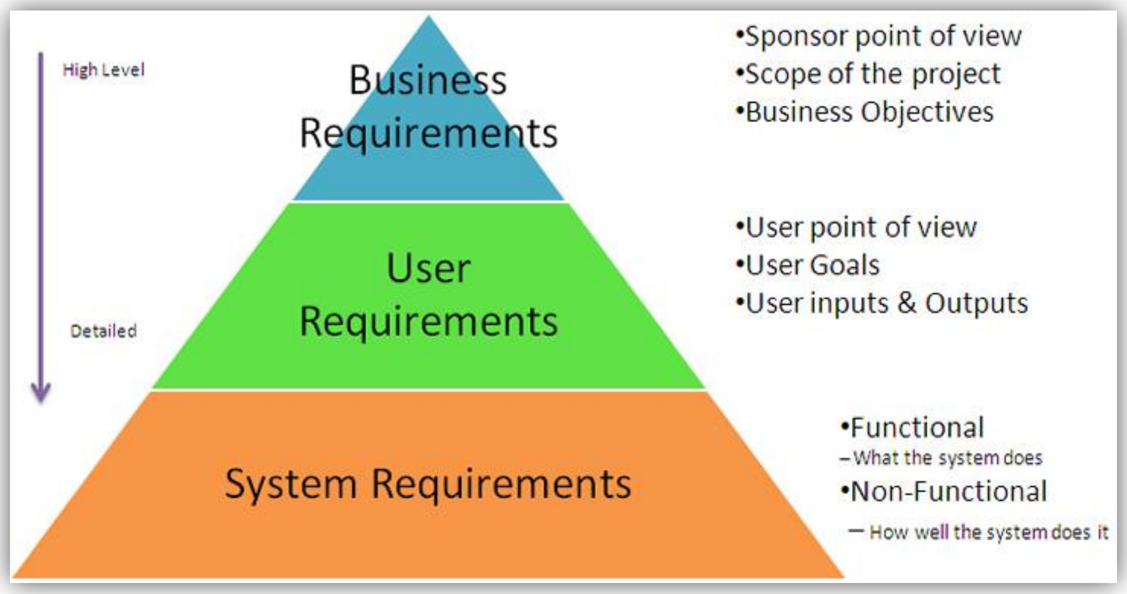
Requirements specification in systems engineering and software engineering is the direct result of a requirements analysis and can refer to:

- **Software requirements specification**
    - A software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements (will explained on detail on detail on chapter 4), and may include a set of use cases that describe interactions the users will have with the software.
    - Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.
    - The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

By that we mean the meaning of software requirements specification and its importance. Software and requirements specification is divided into User and System requirements.

First we must explain the difference between the system and user requirements.

To explain that, see first the next figure (3.2).



Figure(3.2)

3.2.1 User requirement specification:

The user requirement(s) document (URD)or user requirement(s) specification is a document usually used in software engineering that specifies what the user expects the software to be able to do.

Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand features not in the URD, while the developer cannot claim the product is ready if it does not meet an item of the URD.

The URD can be used as a guide to planning cost, timetables, milestones, testing, etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described.

We specify what the user expects the software to be able to do such as:

- He expects that it possible to register for the coming semesters without physically being in Paris.
- He expects that he would take less time to conclude the registration process.
- He expects that he can pay the fees online and complete their registration.
- He expects that he will have a provided better chance to obtain the desired classes and Navigation through the available courses.
- He hopes that there will be exist communication channel with staff that will be easily and know what its duties and rights.

3.2.2 System Requirements Specification

A System Requirements Specification (abbreviated SRS when need to be distinct from a Software Requirements Specification SRS) is a structured collection of information that embodies the requirements of a system.

A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

A system requirements specification is normally produced in response to a user requirements specification.

A system requirements specification may also be used in contract negotiations during and after the purchase of the system, which must meet the specifications that the contractor has agreed to accept. This is especially important where the purchaser sets out the requirements in terms of a range of functions and performance levels that the contractor commits to supplying and meeting, rather than as an inventory of components (hardware and software) that the contractor undertakes to supply.

We specify what the system requirements specification we will able to do such As:

- Providing system with functions that are necessary to respond to the user requirements.
- Taking care about the new technology and provide it on the system.
- Applying the credit hours online registration technology on the system.

Chapter 4

"System Analysis"

4.1: OVERVIEW

4.1.1 Systems Development Life Cycle (SDLC) Methodology

4.1.2 System Development Phases

4.1.3: overview of System Analysis and Design

4.2: SYSTEM ANALYSIS

4.2.1 System Requirements

4.2.1.1 Functional Requirements

4.2.1.2 Non-Functional Requirements

4.2.2.3 Functions of system

4.2.2.4 Use case diagrams.

4.2.2.5 Sequence analysis diagrams.

4.2.2.6 Flow chart analysis diagrams.

4.1: OVERVIEW

4.1.1 Systems Development Life Cycle (SDLC) Methodology

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

Various SDLC methodologies have been developed to guide the processes Involved, including the waterfall model (which was the original SDLC method), System is developed using the waterfall model.

- ✓ **What is the waterfall methodology?**

Waterfall methodology is a sequential design process. This means that as each of the stages (Requirement gathering analysis, Design, Implementation, Testing, Deployment,Maintenance) are completed, the developers move on to the next step.

As this process is sequential, once a step has been completed, developers can’t go back to a previous step – not without scratching the whole project and starting from the beginning. There’s no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then followed carefully.

- ✓ **Advantages of the Waterfall Methodology**

- The waterfall methodology stresses meticulous record keeping. Having such records allows for the ability to improve upon the existing program in the future.

- With the waterfall methodology, the client knows what to expect. They’ll have an idea of the size, cost, and timeline for the project. They’ll have a definite idea of what their program will do in the end.

- In the case of employee turnover, waterfall’s strong documentation allows for minimal project impact.

When should you use waterfall methodology?

- When there is a clear picture of what the final product should be.
- When clients won't have the ability to change the scope of the project once it has begun.
- When definition, not speed, is key to success.

Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process.

In general, a SDLC methodology follows these steps:

1. If there is an existing system, its deficiencies are identified. This is Accomplished by interviewing users and consulting with support Personnel.

2. The new system requirements are defined including addressing any Deficiencies in the existing system with specific proposals for Improvement.

3. The proposed system is designed. Plans are created detailing the Hardware, operating systems, programming, and security issues.

4. The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects of performance must be tested. If necessary, adjustments must be made at this stage.

5. The system is put into use. This can be done in various ways. The new system can phased in, according to application or location, and the old system gradually replaced. In some cases, it may be more cost-effective to shut down the old system and implement the new system all at once.

6. Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

- *The life cycle of any software project is very important for successful project.*

4.1.2 System Development Phases

There are following six phases in every Software development life cycle model:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

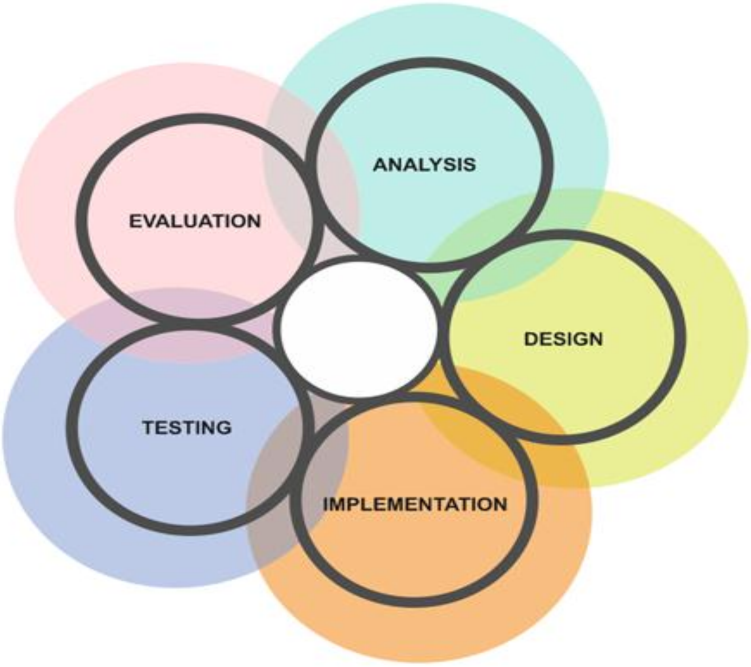


Figure (4.1): SDLC

1) Requirement gathering and analysis:

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

2)Design:

In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

3)Implementation/ Coding:

On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.

4)Testing:

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

5) Deployment:

After successful testing the product is delivered / deployed to the customer for their use.

6) Maintenance:

Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

In this chapter we will discuss the first step of SDLC of the system functions and the interaction between the user and system and steps between them.

4.1.3: Overview of System Analysis and Design

Systems development can generally be thought of as having two major components: System analysis and System design.

System design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective.

System analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system. This is the job of the systems analyst.

The purpose of analysis is to understand what will be built, why it should be built, how much it will likely cost to build (estimation), and in what order it should be built (prioritization).
This is similar to requirements elicitation, the purpose of which is to determine what your users want to have built.

Another way to look at analysis is that it represents the middle ground between requirements and design .

On this chapter we will talk about system analysis of our project and on the next chapter we will talk about the system design of our project.

4.2: SYSTEM ANALYSIS

This chapter presents the analysis phase of the proposed system. In this phase, the system requirements are discussed in order to fulfill the features and objectives previously mentioned in Chapter 1. These requirements are translated to users' interactions in use cases. These interactions described in terms of relation between the user and the proposed system through system sequence diagrams.

4.2.1: System Requirements

The requirements that the system must satisfy are of two types, which are the functional and non-functional requirements.

Functional requirements are the requirements that define a function of software that run on the system.

Non-Functional requirements are the requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors.

In the following subsections, both functional and non-functional requirements of the proposed system are listed.

4.2.1.1: Functional Requirements

The functional requirements of the proposed system can be categorized into two categories according to different users' perspectives. The two perspectives are as follows:

1. Student :

- Freedom of the student in the choice of materials they wish to study each semester with the help of her academic advisor.
- Freedom of the student in reducing the number of materials that are being studied in the classroom if he has found conditions so warrant.

- The possibility of increasing the number of student credit hours registered for study in a given semester, if he got a high accumulative rate according to the list.
- The student can during the summer semester studying some of the material selected from the last semester, so as to ease the burden of the school on the one hand and to improve the cumulative average of the other hand.
- Students can choice the courses they want according to their abilities and time.
- The student can transfer to any foreign university easily to apply the same system. Calculates to him all the materials which studied and taught successfully. If these materials are taught in the university transferred to it.

2. *FCI :*

- Motive behind the construction of this system is the development of the education process in college even promote its student andenable the student to choose the material according to its capabilities in terms of time and abilities of the study and take the time that is appropriate for the completion and graduation as he pleases, and this is by coming to make the student focus on the study and who wants to take advantage of the material and this enable the faculty to graduate well-understood students .

- The goal of the system we are designing is to deliver automatically all the necessary information to register students for the coming semester.
- Through this website the student knows what it duties and its rights.

4.2.1.2: *Non-Functional Requirements*

The system should satisfy the following non-functional requirements:

1. **Accessibility:** that the system should be easy to access. This achieved by,
□ the application will be web-based and can be accessed from any device that has internet connection.

2. **Accuracy:** that the system should provide accurate results.

This achieved by,

□Our application provides accurate information as it depends on the rules of the faculty and The Undergraduate schedule of Computers and Information Faculty

Of EL-Fayoum University 2015.

□Our application depends on international and structured repository of Meta information.

3. **Usability:** that the system should be easy to use. This achieved by,
□ Use friendly GUI (Graphical User Interface).
□ Retrieve information in a user friendly way.

4. **Real-time System:** that the system should provide solutions in suitable time. This achieved by:
□ Retrieve information in a fast way.

5. **Scalability:** that the system should be scalable. This achieved by:
□ Every time new data will be added to the system. This presents a challenge in how to make the system scalable in a way that can accommodate the expected increase in data without significantly affecting the system performance.

5. **Security (Explained in details on chapter 6):**
□ prevent unauthorized access from other people.

4.2.2: Analysis of system functions.

We make list of functions that we will do on the system as shown next:

4.2.2.1: Functions of online system registration for Faculty of computer sciences and information due to El-fayoum University.

1. about FCI.

1.1. Mission && objectives

1.2. Vision.

1.3. Study plan.

1.4. Academic degree.

1.5. Calendar.

1.6. Visitors Guide.

1.7. Documents (login as "administrator or instructor").

1.8. Building.

1.9. Faculty history.

1.10. Code of ethics.

2. Academics.

2.1. Academic programs.

2.2. Course schedules (course time table).

2.3.Academic courses.

2.4.Academic services.

2.5.Faculty&& Staff.

2.6.Academic rules.

2.6.1. College level plan.

2.6.2. How to calculate CGPA.

2.7. List of degrees.

3. Prospective students

3.1. Admission policy .

3.2. Policies and procedures.

3.3. Faculty fee.

3.4. Rules of enrollment and registration.

3.5. About doctors && assistants.

3.6. Career after faculty.

3.7. Student guide “download”.

4. Current students.

4.1. Registration.

4.2. Exams scheduling.

4.3. Results of exams (student id, student password).

4.4.Regulations for Medical Excuses.

4.5. Online fee payment.

5. Resources.

5.1.Soft skill stations &&scientific events.

5.2**.** Student affairs**.**

5.3.Writing center.

5.4. Disability services.

5.5. Computer services.

5.6. Libraries.

5.7.Student and Alumni Affairs Office.

5.8. Online services.

5.9.Crisis management center and disaster.
5.10. Unit to communicate with alumni.

6. Graduation projects.

That contains all graduation projects for last years and description for it.

6.1.proposal.

6.2.marks.

6.3.GraduationProject Weekly Report.

7. Researches.

System have list of functions that the user not need to interact and do any operations with the system, the system will display them directly to the user

such as:

Mission && objectives, Vision, Internal law, Academic degree, Calendar,

Visitors Guide……………….and so on.

On the other hand, we have list of functions that the user need to interact and do operations with the system to display them such as:

- Documents (staff Id, staff Password).
- Academic schedules (course time table).
- College level plan.
- How to calculate CGPA.
- Registration.
- Results of exams.

4.2.2.2: Use case diagrams.

We will illustrate the previous six functions on detail and show the analysis of them by using Use Case Diagrams.

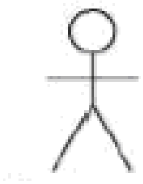
Use Case Diagrams:

- Use cases represent system functionality from the user's perspective
- Use Case diagrams describe who will use the system and in what ways the user expects to interact with the system.
- Use Case diagrams represent the interactions between use cases and actors
- Use Case diagrams should be defined during the initial stage of a project.

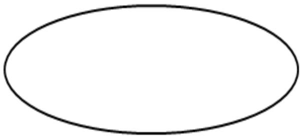
The Components of Use Case Diagram :

- The two main components of use Case diagrams are:

1. Actor



2. Use Case



The Analysis of Use Cases :

- It is necessary to have more specific details to build the system.
- These details are known as the analysis of use case.
- The analysis of use case will describe the flow of events
- The analysis is needed as a document to describe in detail what the user of the system will do and what the system itself will do.
- The scenario in the analysis of use case is as follows:
  - ✓ The user does an action, then
  - ✓ The system does an action as a response,
  - ✓ Then the user does,
  - ✓ Then the system responds,
  - ✓ And so on.

➢ *We will make the use case for each function:*

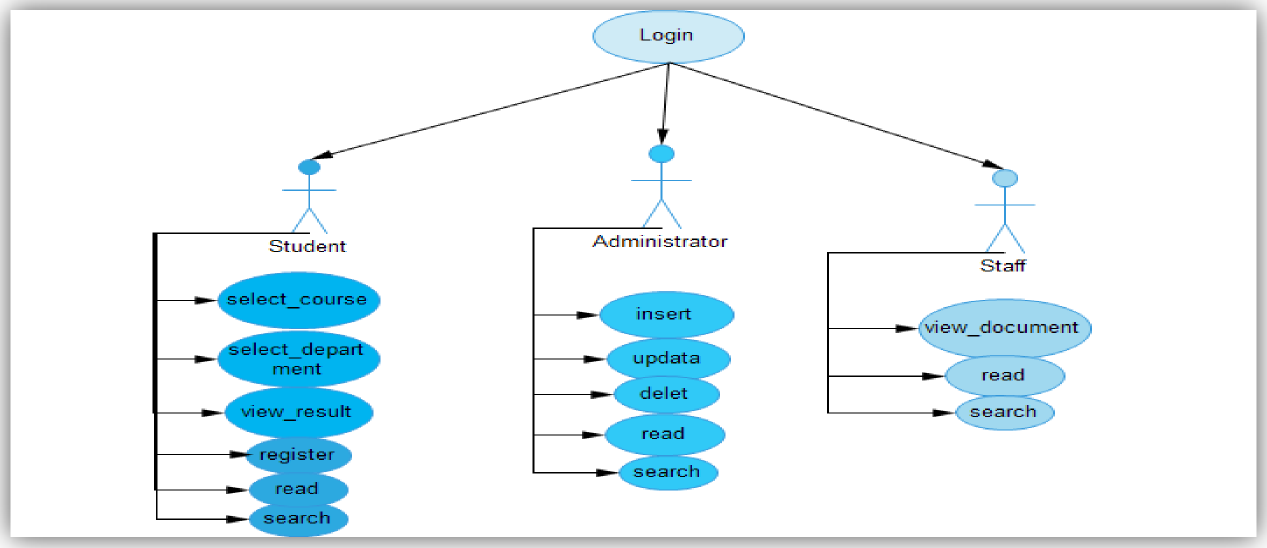


Figure (4.2)

1. Login

Actor: Students.

Use case: Login.

Use Case Name:	Login	
Actor(s):	Students.	
Description:	This function is performed by students . So it requires user name and password to enable them to make some operations such as register, view result.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: This use case is initiated when a student enter the user name and password that belong to him. Step2: student click on the login button.	Step3: The student's user name and password are validated with what is current on database. Step4: the system will open the page if the user name and password are correct.
Alternate Courses:	Step5: if The student's user name and password are not correct, the system will send a message (try again ()).	
Preconditions	Student must be already has user name and password and already exist on data base.	

post conditions	The student is logged in and then the required page is open.	

Table (4.1)

1. Documents (staff Id, staff Password).

Actor: Employees (staff & administrator).

Use case: Login.

Use Case Name:	Login for documents	
Actor(s):	Employees.	
Description:	This function is private for only Employees. So it requires user name and password to enable them to make some operations.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: This use case is initiated when a staff enter the user name and password that belong to him. Step2:staff click on the login button.	Step3: The staff's user name and password are validated with what is current on database. Step4: the system will open the page if the user name and password are correct.

Alternate Courses:	Step5: if The staff's user name and password are not correct, the system will sent a message (try again ()).	
Preconditions	-The page is visible for Only staff. -staff must be already have user name and password and already exist on data base.	
post conditions	Any changes are recorded in the database.	

Table (4.2)

Administrator’s functions:

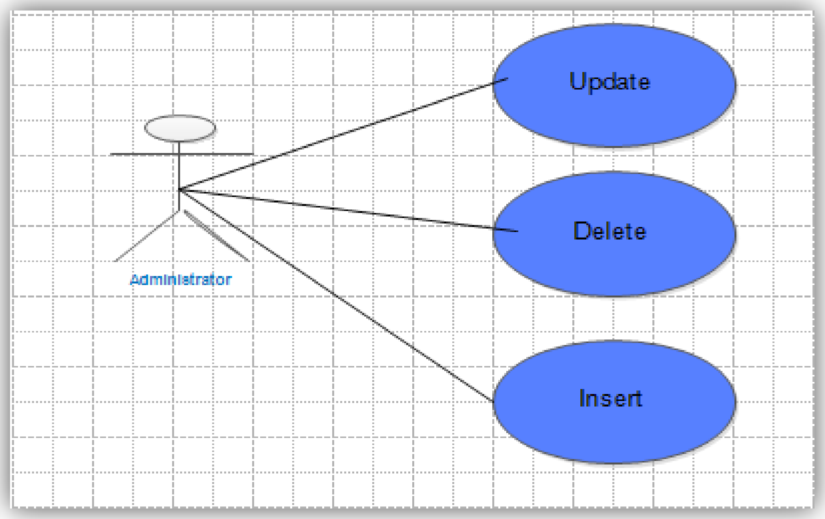


Figure (4.3)

1. Update

Actor: Administrator.

Use case: update data.

Use Case Name:	Update data.	
Actor(s):	Administrator.	
Description:	This function is performed by administrator when he wants to change data on database.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: This use case is initiated when the administrator enter the user name and password that belong to him on login function. Step2: he click on the login button.	Step3: The Administrator’s user name and password are validated against what is current on database.

	Step4: administrator determine data he wants to update. Step5: administrator enters new that data to make updates and then click on save button.	Step6: the system will check data entered and if it is correct, system makes these changes on database.
Alternate Courses:	Step3: if The administrator's user name and password are not correct, the system will send a message (try again ()). Step4: if new data he entered isn't correct , system will show an error message (" not valid data").	
Preconditions	The administrator must log in the system.	
post conditions	System will show an message("data is inserted a correctly way")	

Table(4.3)

2. Insert data

Actor: Administrator.

Use case: insert data.

Use Case Name:	Insert data.	
Actor(s):	Administrator.	
Description:	This function is performing by administrator when he wants to insert new data on database.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: This use case is initiated when the administrator enter the user name and password that belong to him on login function. Step2: he click on the login button.	Step3: The Administrator's user name and password are validated with what is current on database.
	Step4:administrator choose tables he wants to insert in the new data. Step5: administrator enters that new data into that selected table.	Step6: the system will check data entered and if it is correct, systeminsert these additions into the database.
Alternate Courses:	Step3: if The administrator's user name and password are not correct, the system will send a message (try again ()). Step4: if new data he	

	entered isn't correct, system will show an error message ("not valid data").	
Preconditions	The administrator must log in the system.	
post conditions	System will show an message("data inserted in a correctly way")	

Table (4.4)

3. Delete data

Actor: Administrator.

Use case: Delete data.

Use Case Name:	Delete data.	
Actor(s):	Administrator.	
Description:	This function is performing by administrator when he wants to delete a specific data on the database.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: This use case is initiated when the administrator enters the user name and password that belong to him on login function. Step2: he click on the login	Step3: The Administrator's user name and password are validated with what is current on database.

	button.	
	Step4: administrator chooses data (value, table, attribute, all data base) he wants to delete. Step5: administrator deletes that data which he selected.	Step6: Action of deletion will happen.
Alternate Courses:	Step3: if The administrator's user name and password are not correct, the system will send a message (try again ()).	
Preconditions	The administrator must log in the system.	
post conditions	System will show an message("data deleted in a correctly way")	

Table(4.5)

2. Academic schedules (course time table).

Actor: student.

Use case: select course name &view

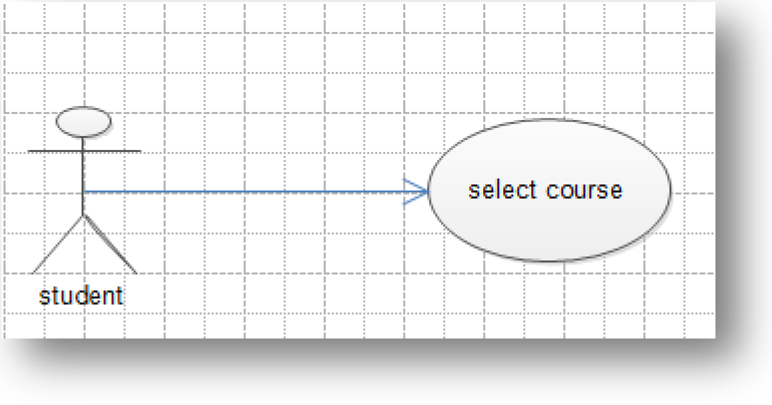


Figure (4.4)

Use Case Name:	select courses time table	
Actor(s):	Student.	
Description:	This function show the time table for each course the user will select from the list of courses, the system will be shown.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: The student will choose a course from list of available courses in the system. Step2: the student will click on the submit button.	Step3: The system will call the time table from the database for the selected courses and show it to the student.

Alternate Courses:	Step3: if The course does not have a time table that not found on the database , the system will show a message(not available time table)	
Preconditions	The student must select a course.	
post conditions	Non at this time	

Table (4.6)

3. College level plan.

Actor: student.

Use case: select department.

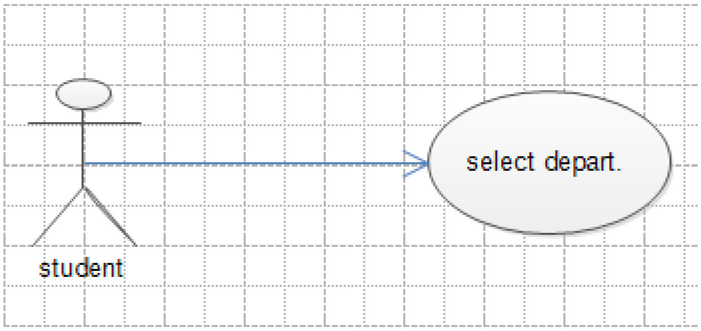


Figure (4.5)

Use Case Name:	select department	
Actor(s):	Student .	
Description:	This function show the courses for each department the user will select from the list of departments, the system will be shown.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: The student will choose a department from list of available departments in the system. Step2: the student will click on the submit button.	Step3: The system will call the level And credit hours for each level from the database for the selected departments and show it to the student.
Alternate Courses:	Step3: if The department does not have a course that not found on the database , the system will show a message(not available course)	
Preconditions	-The student must select a department.	
post conditions	Non at this time	

Table (4.7)

4. How to calculate CGPA (explain).

You need to calculate your GPA (Grade Point Average, School use GPA's as a means of ranking students based on their academic performance.

The cumulative grade point average of the student account (GPA) as follows:

1. The estimate is multiplied by the value of each course (the points outlined in the table) in the number of credit hours for this decision to get the number of points for each Course.

2. Adding points are all courses in which the student record.

3. Divide the total points on the total hours recorded for the student to get a cumulative grade point average as follows:

CGPA = (grades * credits) / credits. (Equation 4.1)

4. The overall rating account for the student based on grade point average according to the following schedule:

How GPA was calculated?

Overall rating	**Cumulative average**
Excellent	*3.4 And more*
Very good	2.8 to less than 3.4
Good	2.4 to less than 2.8
Acceptable	2 to less than 2.4
Weak	1.4 to less than 2
Very weak	Less than 1.4

Table(4.8)

5. Registration.

Actor: student.

Use case: register.

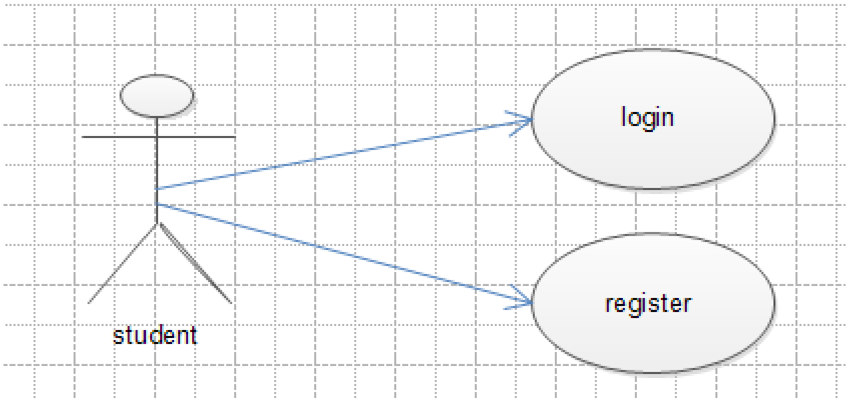


Figure (4.6)

Use Case Name:	Register the courses.	
Actor(s):	Student.	
Description:	This function is shown when the student will choose courses to be recorded.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: The student enters the user name and password that must entered in the login function. Step2: the student will click on the login button.	Step3: The system will check for the user name and password that were entered and check for the validation of them and check if the student was recorded in the database. Step4: the system will open the registration page for student to register the courses that the student wants. Step5: The system will check if student is new or already registered before. Step6:assume that he is already

	Step 8: the student will choose the courses that he wanted by click on the check box. Step 9: after student click on the desired courses from both mandatory and elective courses, he click on submit button.	registered before, the system will check GPA For student must be > 1.4 (Thus, he passes the semester) then,system checks the grade of each course, then it must be >1.4 (passed successfully.) Step7: after system check for the student 's GPA, it will open Mandatory and elective courses that student not registered before (Allowed courses). Step 10: system checks on the chosen courses selected by student and they if they correct and don't have prerequisite courses that student don't studied before (checking the student's schedule time table data and prerequisite courses table on data base.) Step 11: after checking, system will show a time table for student to ensure what he chosen.
Alternate Courses:	Step12: if The student user name and password are not correct, the system will send a message (try again ()). Step13: - If student is new comer so it doesn't have a GPA, system will not check failing on any courses because there isn't exist any courses registered before it was the first level	

	For him,System will open all mandatory and elective courses. Step14: -If student’s grade < 1.4 for one or two courses (Fail), system will add that courses on the time table for student and indicates that student must take these courses with the semester courses that student selected. - If student’s GPA < 1.4 (Fail),this means that there are more than two courses grades<1.4, then the Student must renter the whole semester again. -If the student is so weak (means that student has so weak GPA), student must register Under the care of his academic guide. -If student’s GPA for courses is more than 3.4 (Excellent), system will give student three credit hours bonus (over) on elective courses. -if student is incomplete (means that student enter all exams for specific courses but not enter final only),	

	system will calculate the student degrees for that course as same as the main course degree. -if student is withdraw (means that student do not enter any exam of specific Course), system will calculate the student degree for that course from the half of the main course degree. Step14: if selected courses Have prerequisite courses that student not pass or register before, system will not accept these courses and show an error message and let student choose another ones.	
Preconditions	-This functions must be initiated by a student And login function ().	
post conditions	1. The system doesn't accept from the student to record courses that have passed it. 2. The system doesn't accept courses that have prerequisite. 3. The selected courses must record in the database.	

Table (4.9)

6. Results of exams.

Actor: student.

Use case: view results.

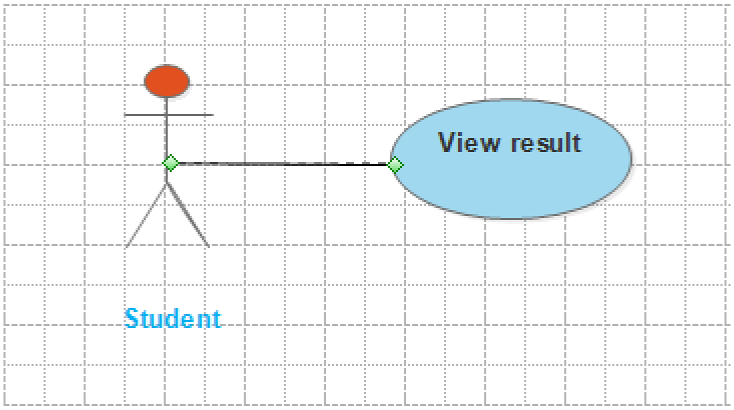


Figure (4.7)

Use Case Name:	Show results.	
Actor(s):	Student.	
Description:	This function is required when the student wanted to view the results of exams.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step1: The student enters the user name and password that must entered in login function. Step2: student click on the submit button.	Step3: The system will check for the user name and password that was entered and check if the student was recorded in the database. Step4: the system will validate the user name and password that was entered. Step5: The system shown the result

Alternate Courses:	Step6: if The student user name and password are not correct, the system will send message (try again ()).	
Preconditions	The actor must be a student in the faculty.	
post conditions	The system shown the result according to particular student.	

Table (4.10)

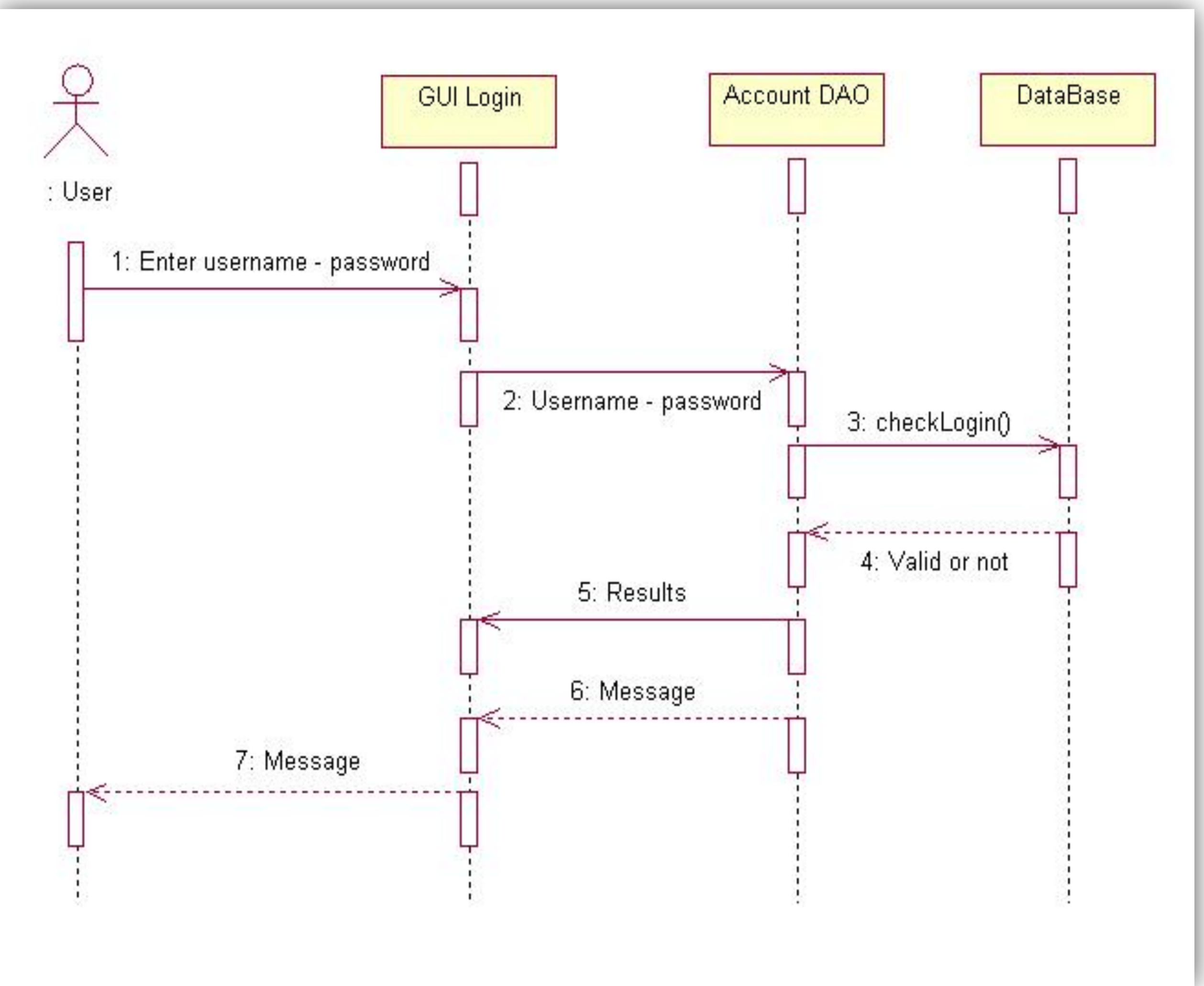
4.2.2.5: Sequence analysis diagrams:

- **Sequence** diagram represents one use case (For each use case there is the corresponding sequence diagram)
- **Sequence** diagram shows step-by-step the flows through a use case, so we need to know:
    - What **objects** are needed for the flow: (Examine the nouns in the flow of events)

- What **messages** the objects send to each other during the flow (Communication between objects)
- What **actor initiates** the flow (External user (who or what) that starts flow of events)
- What **order the messages** are sent (Activities or Events order)

We will discuss the sequence diagram for the following functions:

1. Login



Figure(4.8)

2. Select course time table.

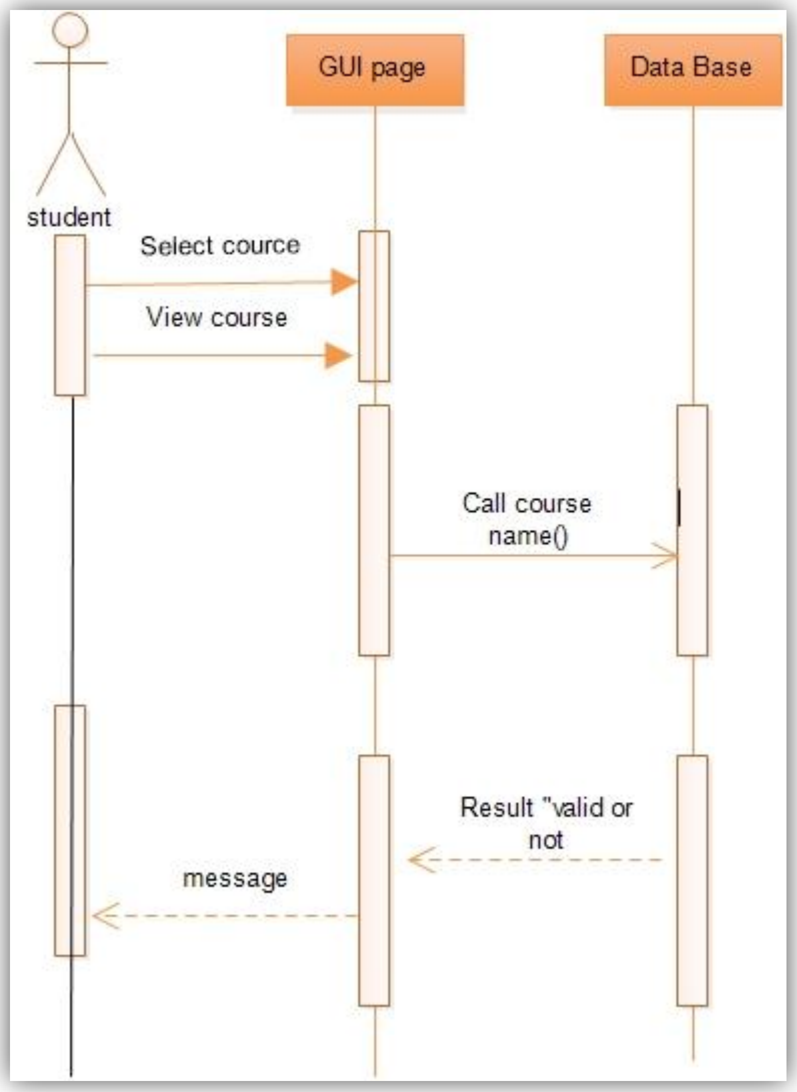


Figure (4.9)

3. View college level plan.

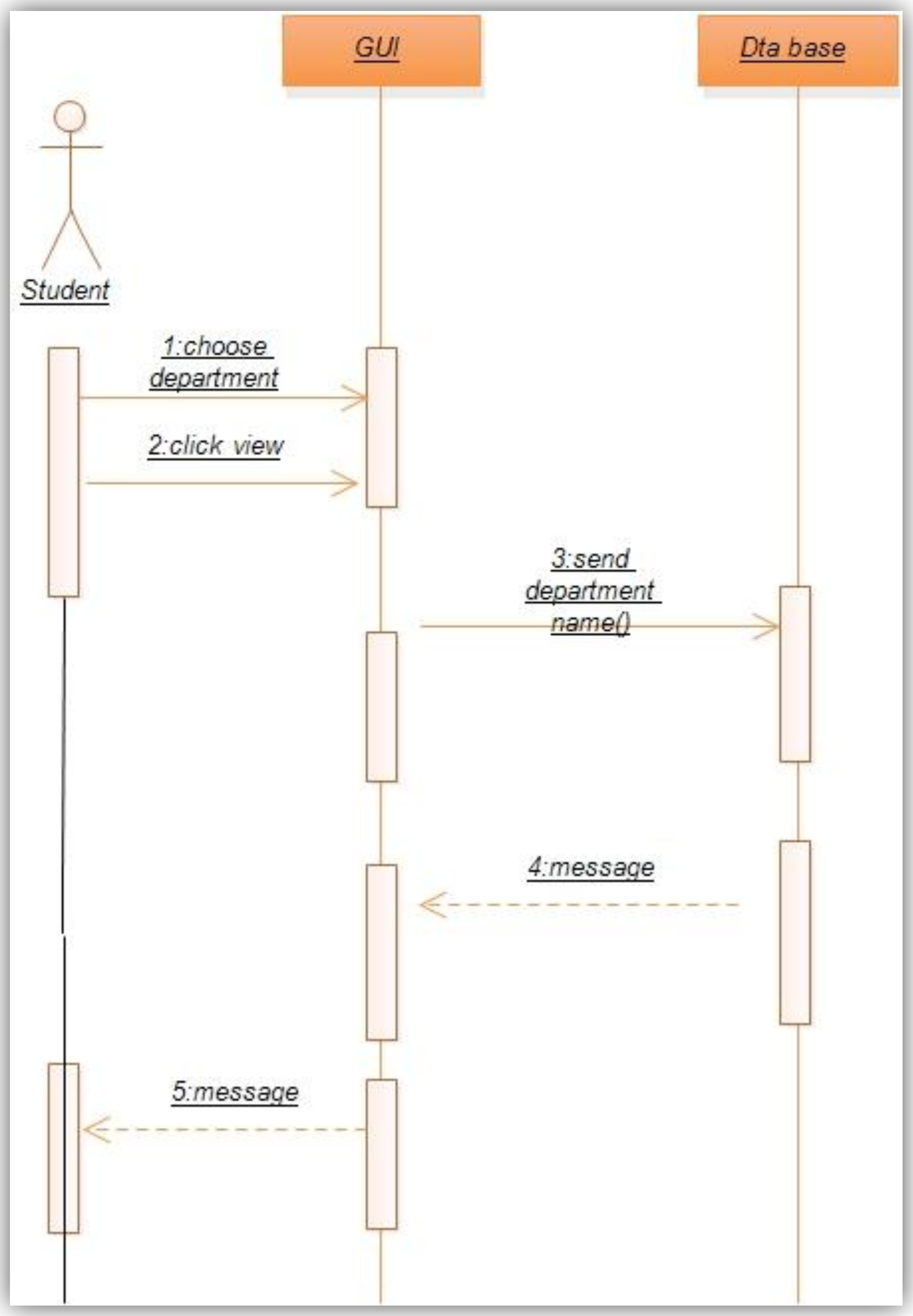
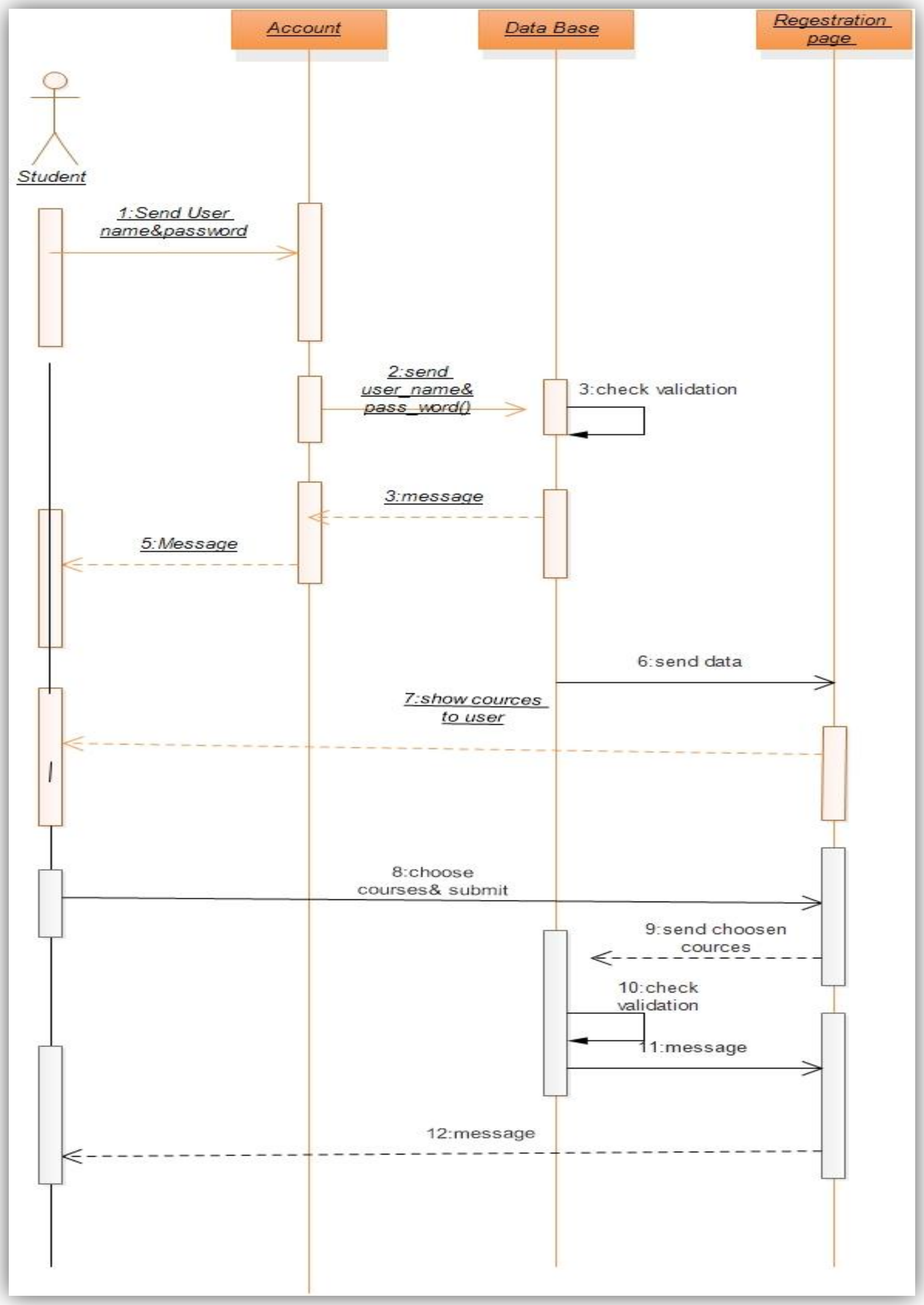


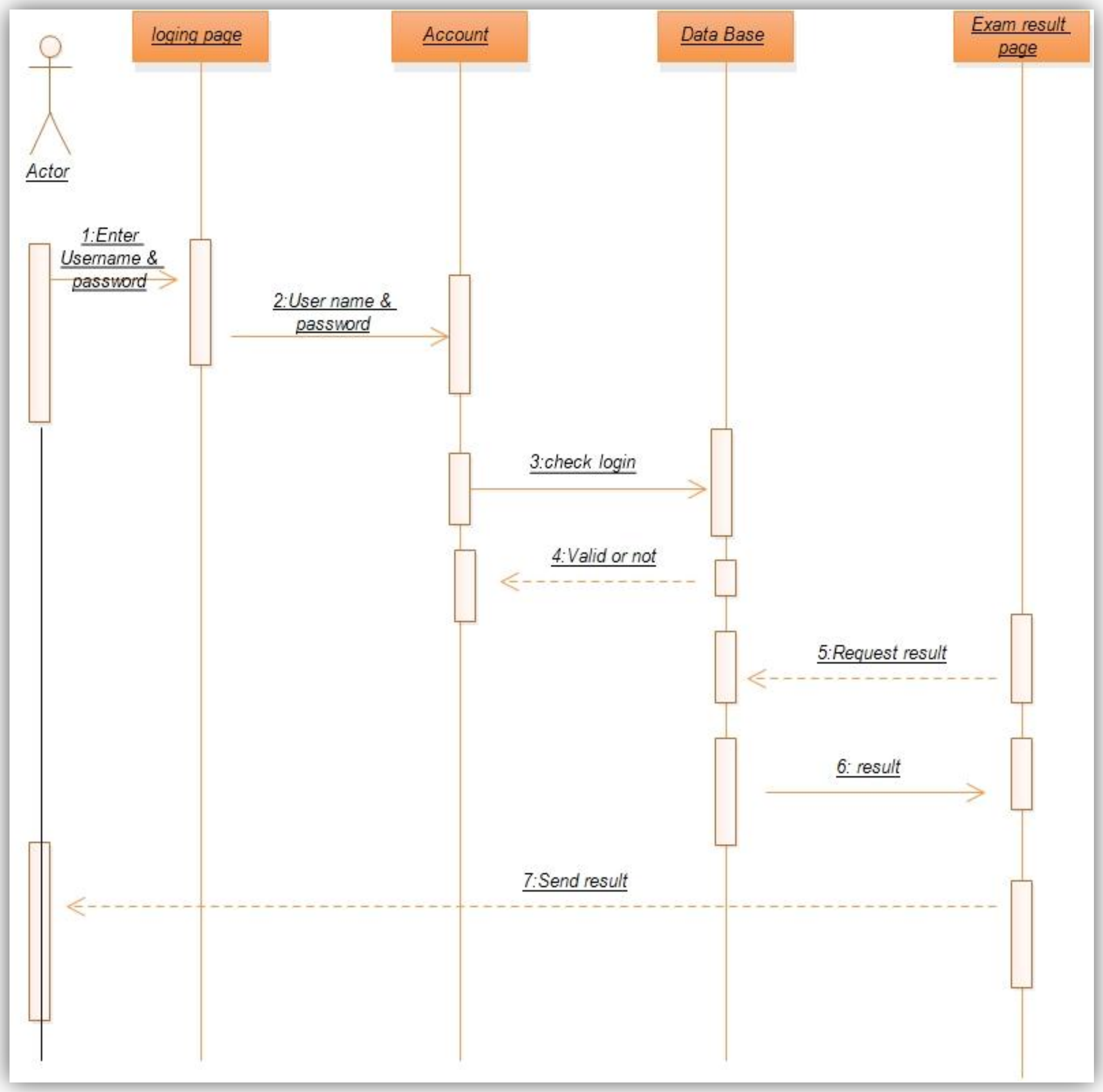
Figure (4.10)

4. Register.



Figure(4.11)

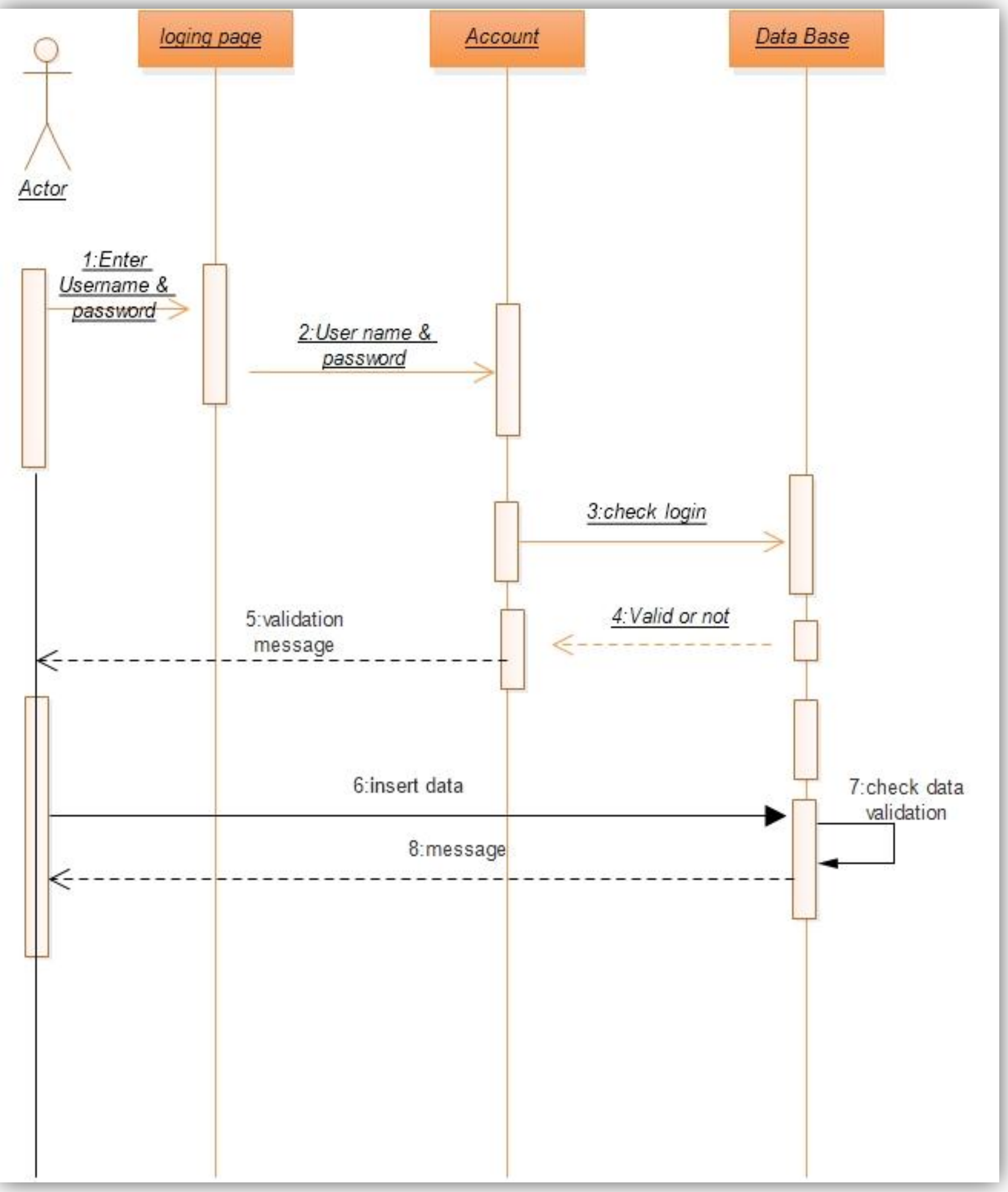
5. View Results of exams.



Figure(4.12)

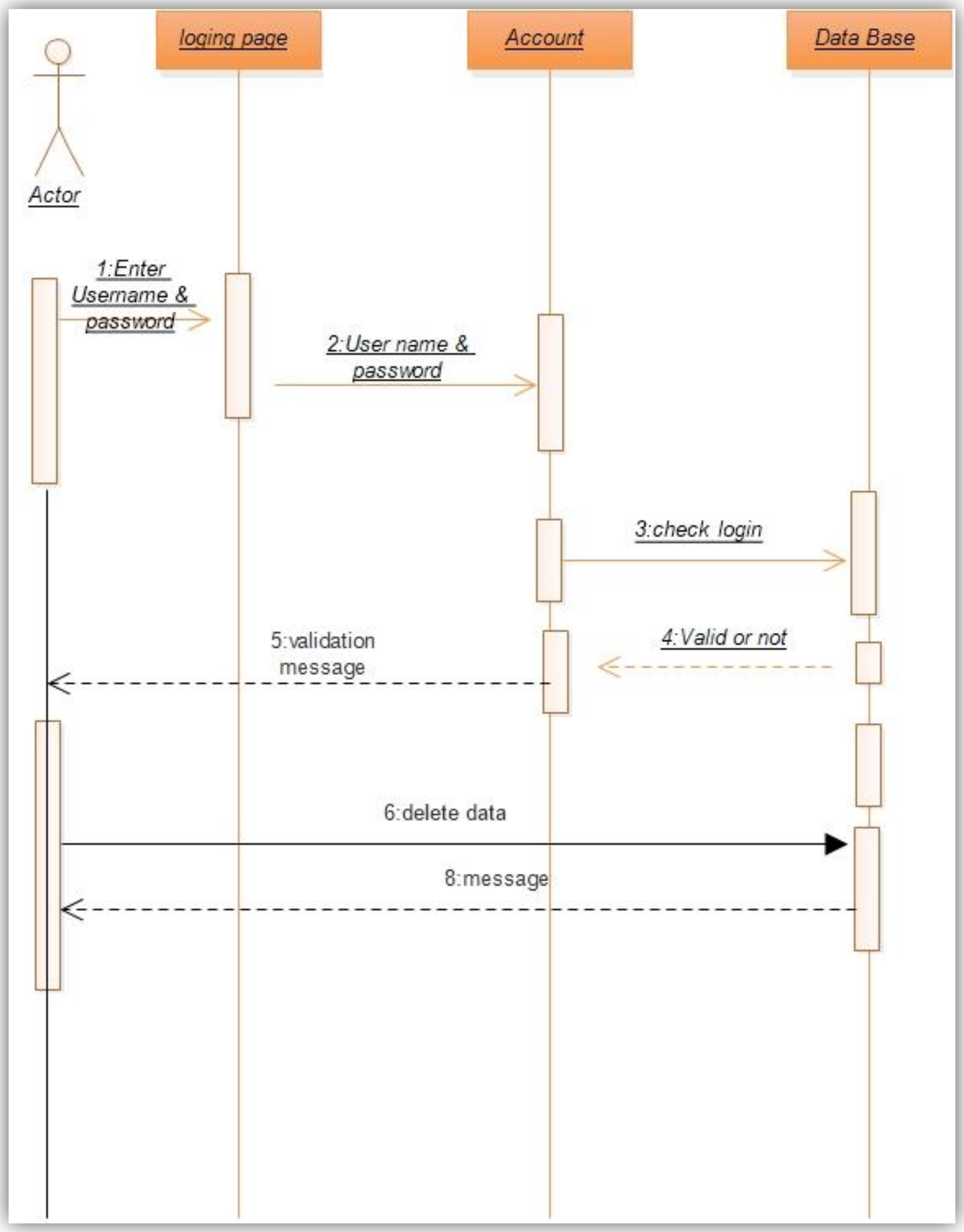
6. Administrator's functions:-

1. Insert



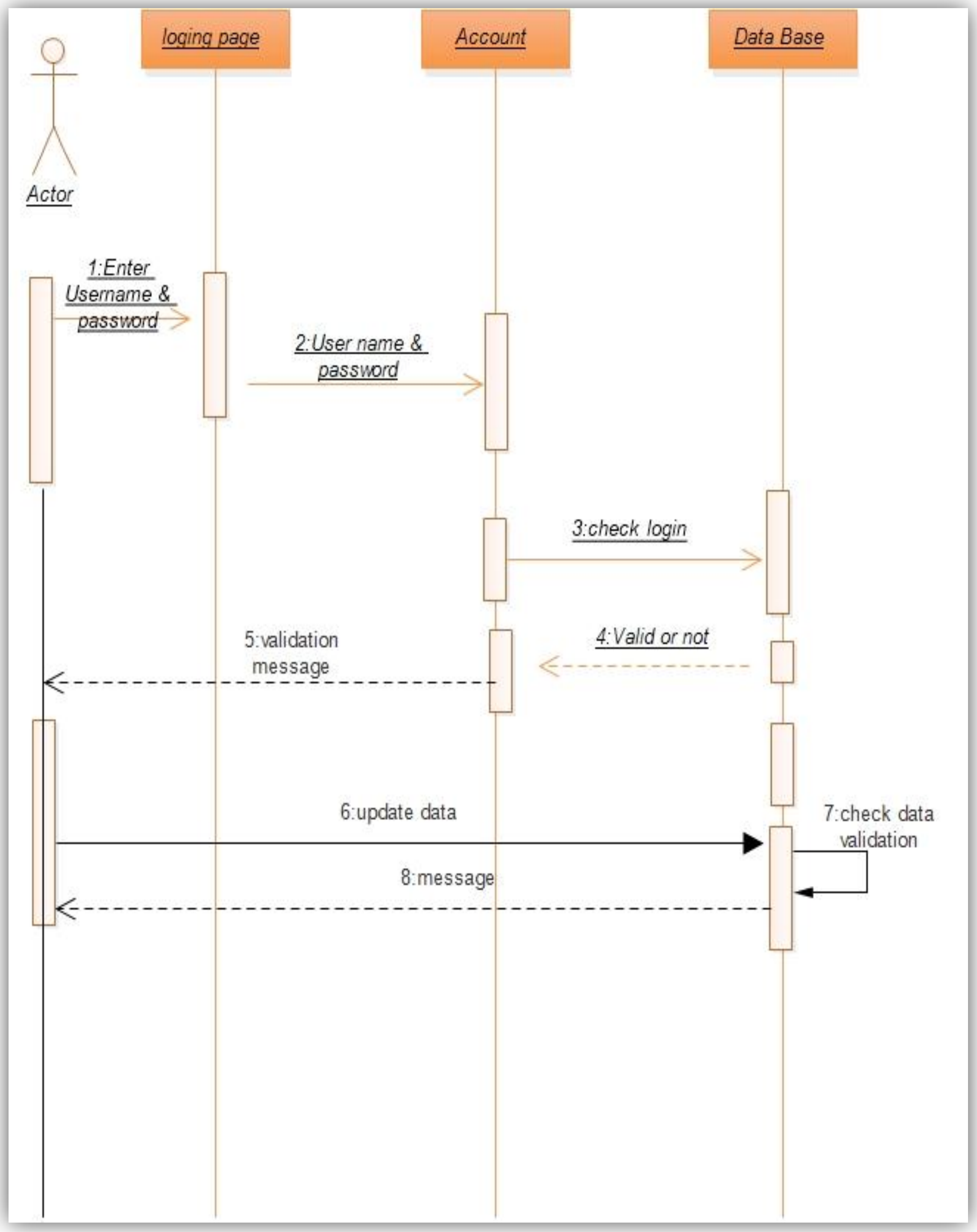
Figure(4.13)

2.Delete



Figure(4.14)

3.Update



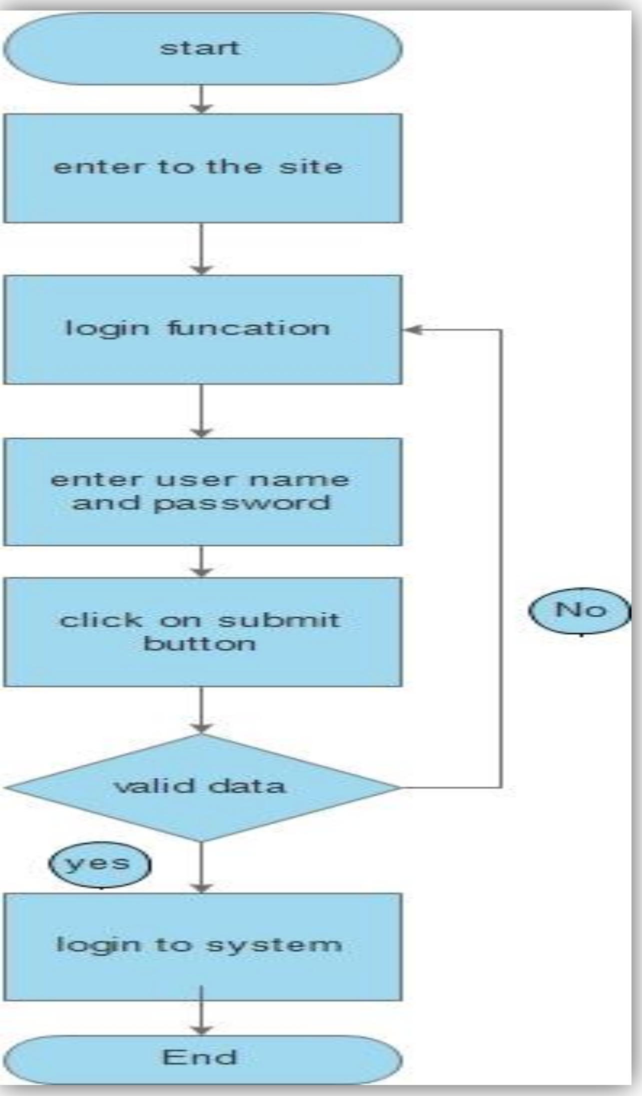
Figure(4.15)

4.2.2.6: Flow chart analysis diagrams:

A flow chart is a type of diagram representing a process using different symbols containing information about steps or a sequence of events. Each of these symbols is linked with arrows to illustrate the flow direction of the process.

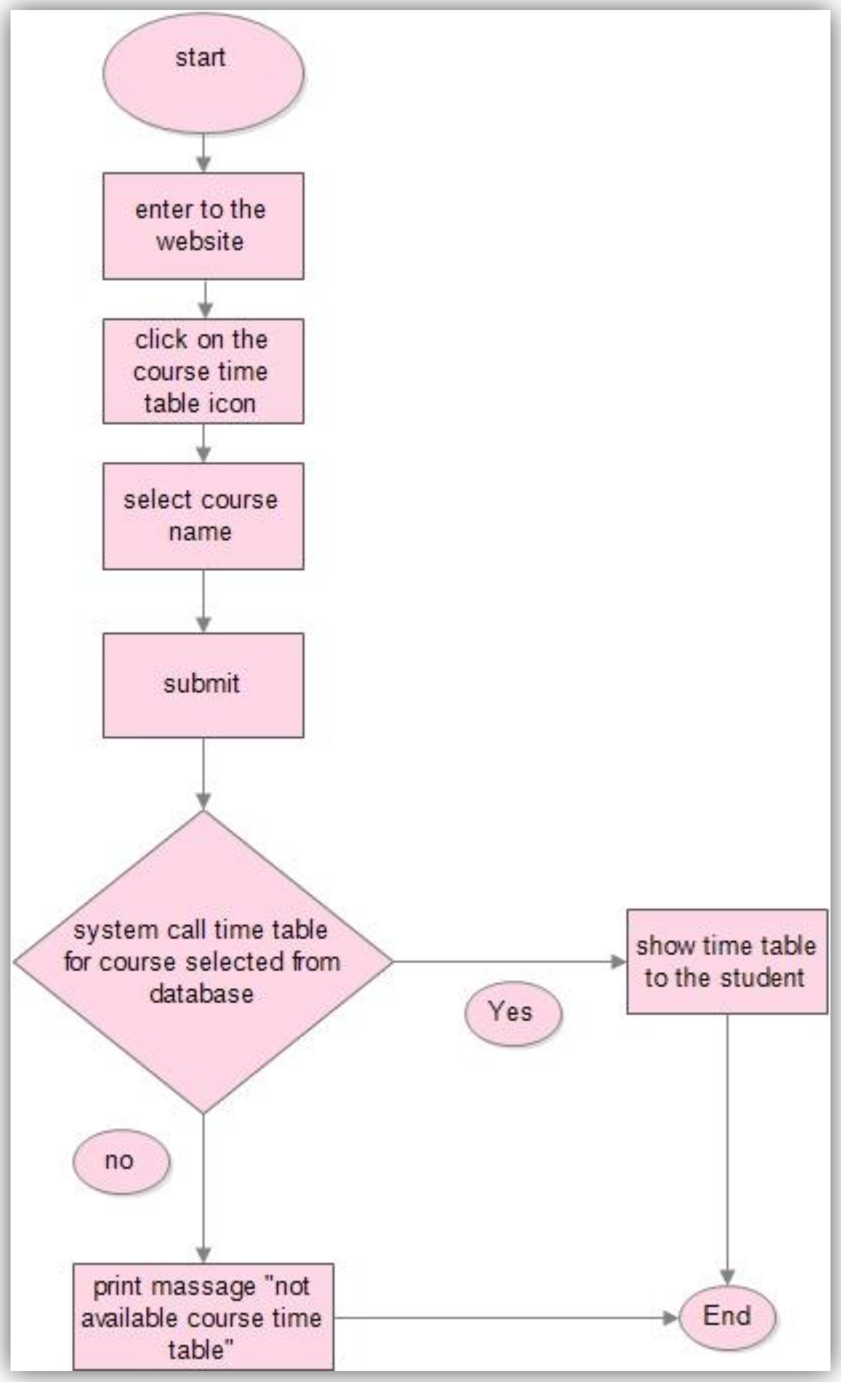
Flowcharts are a methodology used to analyze, improve, document and manage a process or program.

1. **Login**



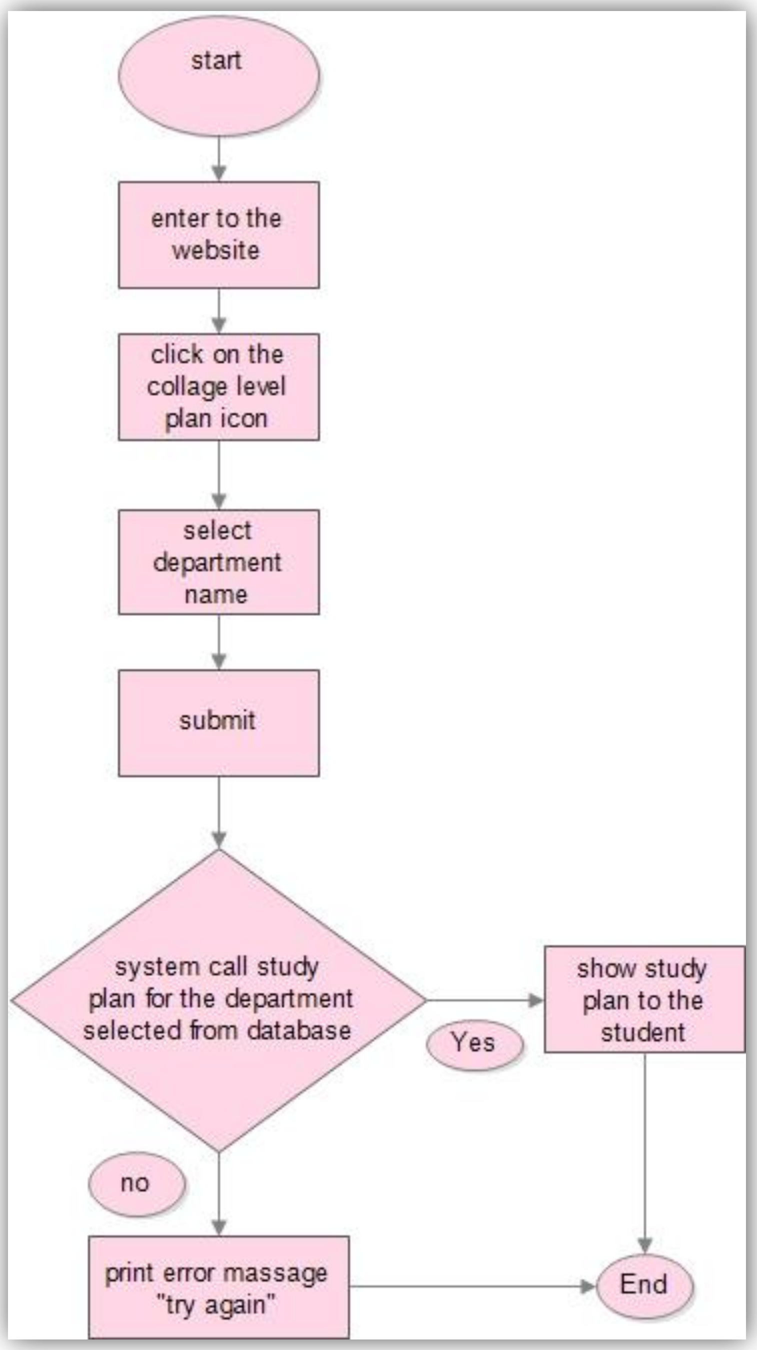
Figure(4.16)

6. Select course time table.


start
enter to the website
click on the course time table icon
select course name
submit
system call time table for course selected from database
Yes
show time table to the student
no
print massage "not available course time table"
End

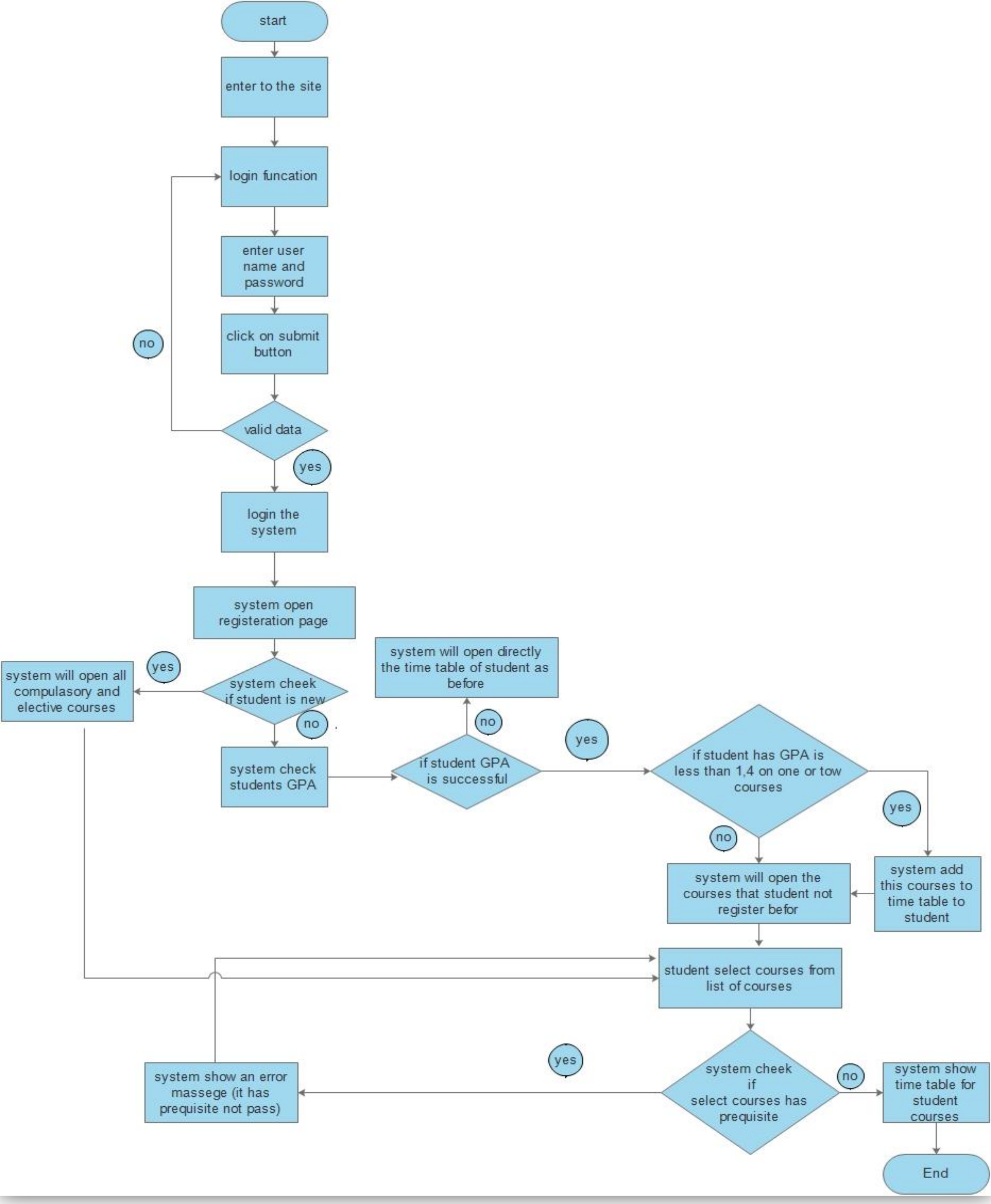
Figure(4.17)

3.View college level plan.


start
enter to the website
click on the collage level plan icon
select department name
submit
system call study plan for the department selected from database
Yes
show study plan to the student
no
print error massage "try again"
End

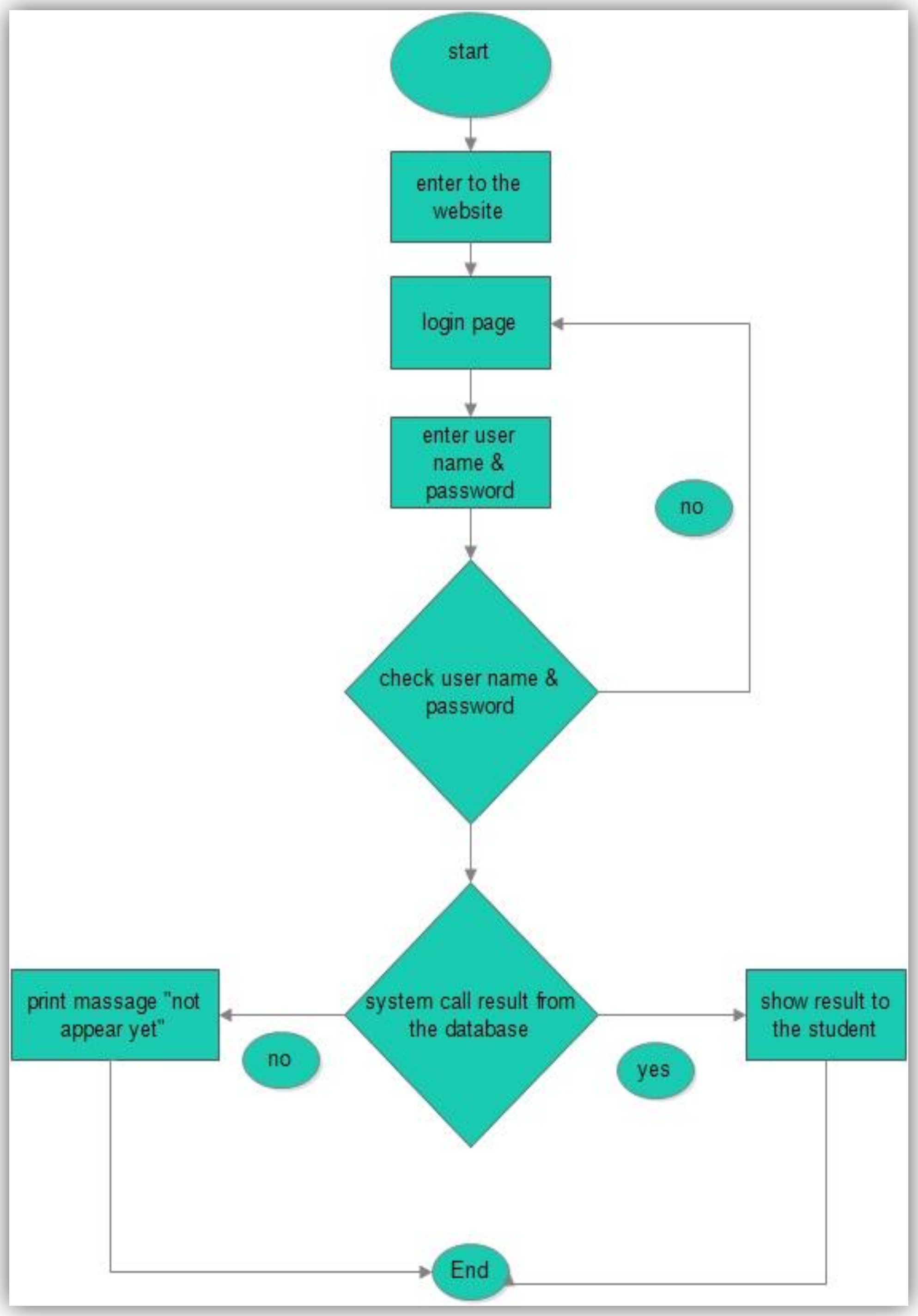
Figure(4.18)

4. Register.


start
enter to the site
login funcation
enter user name and password
click on submit button
no
valid data
yes
login the system
system open registeration page
yes
system will open all compulasory and elective courses
system cheek if student is new
no
system check students GPA
system will open directly the time table of student as before
no
if student GPA is successful
yes
if student has GPA is less than 1,4 on one or tow courses
yes
no
system will open the courses that student not register befor
system add this courses to time table to student
student select courses from list of courses
yes
system show an error massege (it has prequisite not pass)
system cheek if select courses has prequisite
no
system show time table for student courses
End

Figure(4.19)

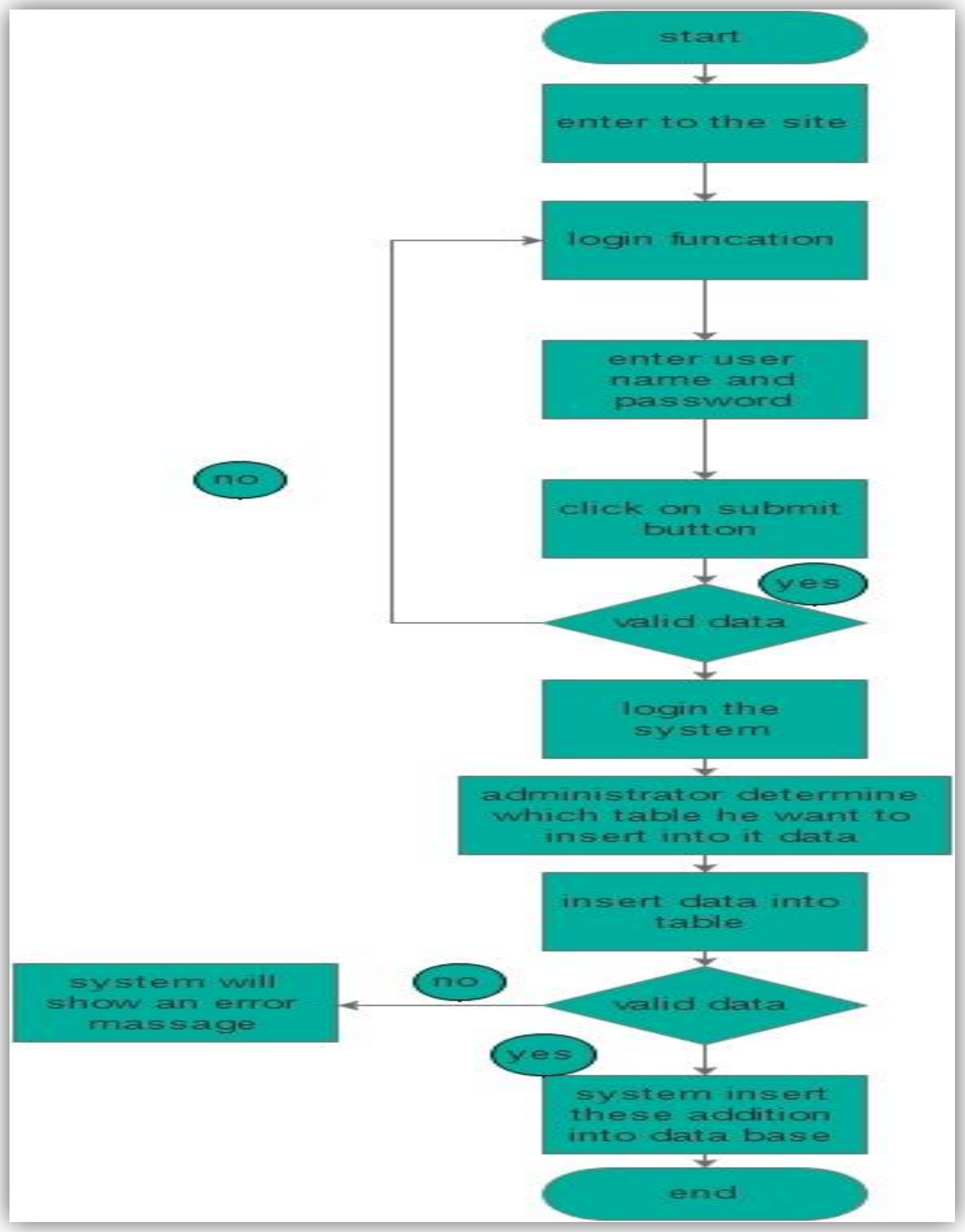
5.View Results of exams


start
enter to the website
login page
enter user name & password
no
check user name & password
print massage "not appear yet"
no
system call result from the database
yes
show result to the student
End

Figure(4.20)

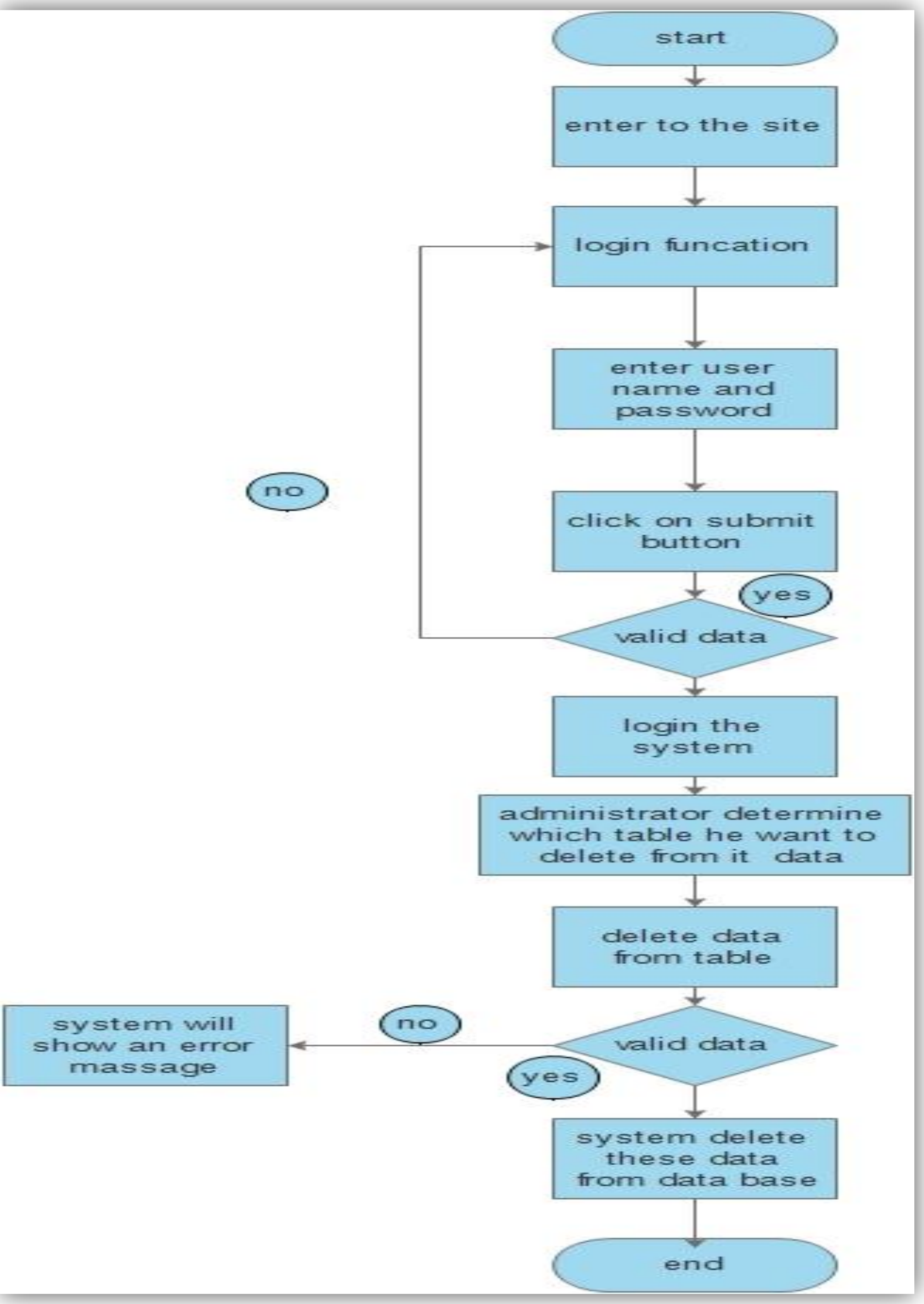
6. Administrator’s functions:

1. Insert


start
enter to the site
login funcation
enter user name and password
click on submit button
yes
valid data
no
login the system
administrator determine which table he want to insert into it data
insert data into table
valid data
no
system will show an error massage
yes
system insert these addition into data base
end

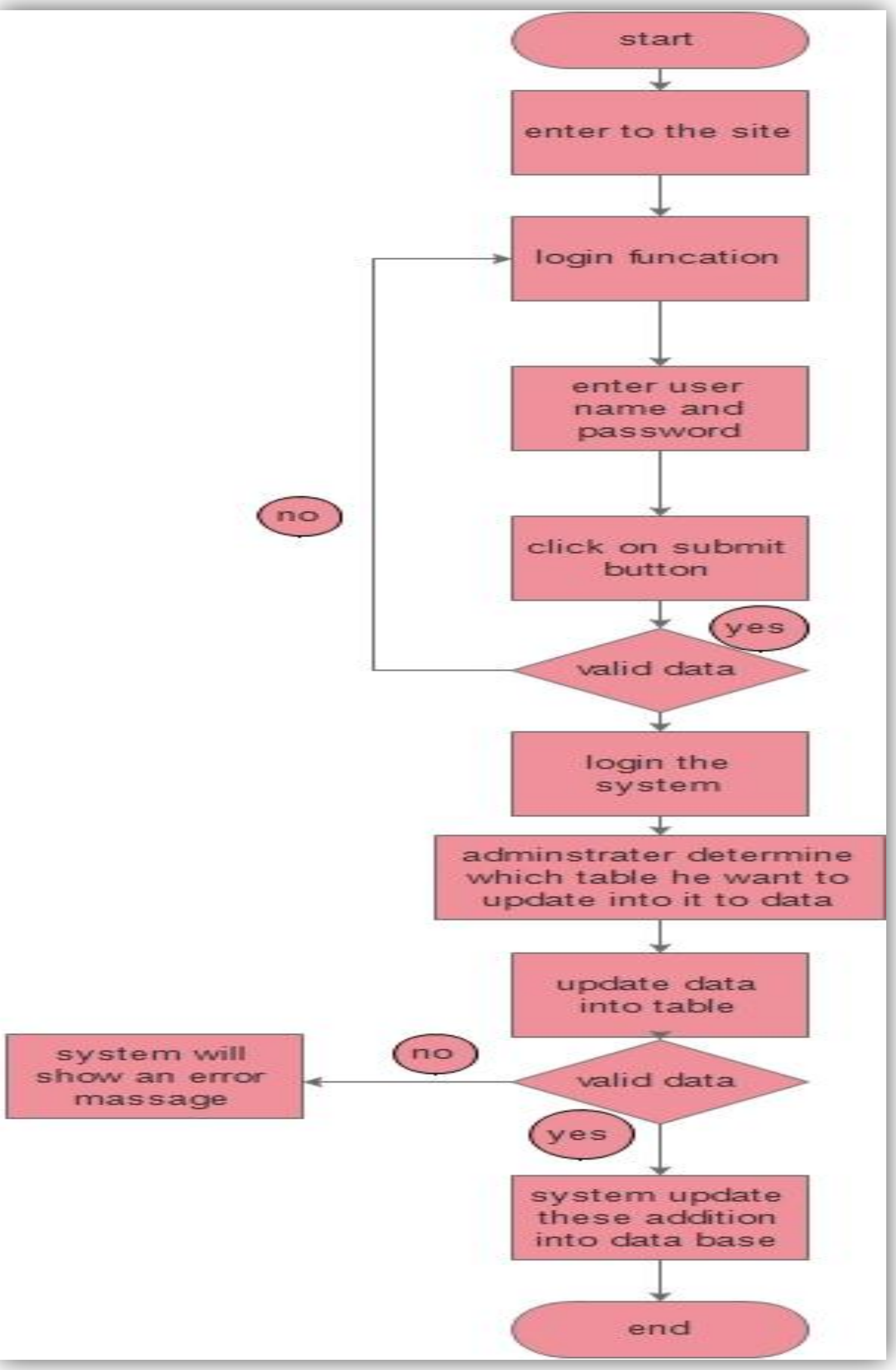
Figure(4.21)

2-Delet


start
enter to the site
login funcation
enter user name and password
no
click on submit button
yes
valid data
login the system
administrator determine which table he want to delete from it data
delete data from table
system will show an error massage
no
valid data
yes
system delete these data from data base
end

Figure(4.22)

3-update


start
enter to the site
login funcation
enter user name and password
click on submit button
no
yes
valid data
login the system
adminstrater determine which table he want to update into it to data
update data into table
system will show an error massage
no
valid data
yes
system update these addition into data base
end

Figure(4.23)

Chapter 5

“Architecture and Design”

5.1 Overview

5.2 Design of system.

5.3 Design of database.

5.3.1 ERD Diagram
5.3.2 Description of ERD Tables

5.1Overview

In this chapter we will present credit hours systems architecture. The system architecture defines the main system components and how they will be used to satisfy the functional and non-functional requirements.

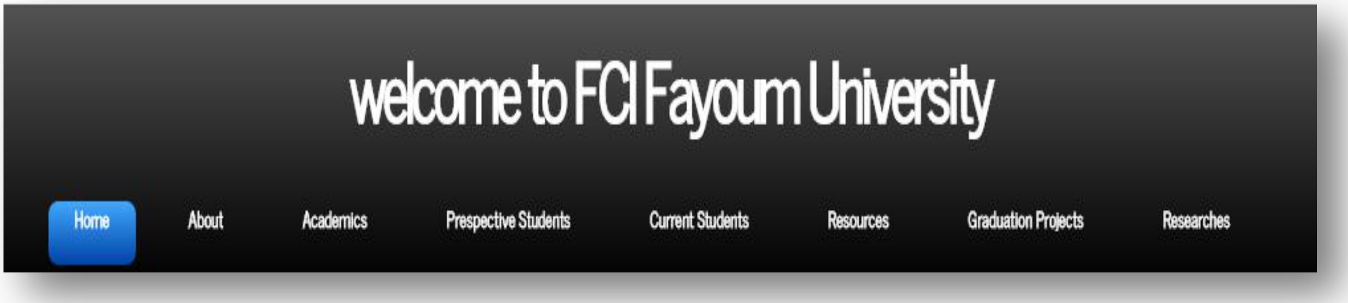
- The Design stage describes how the proposed solution is to be developed. The solution design is specific to the system's technical environment and the tools to be used in constructing the system. The results of this stage will be inputs to the Execute and Implement stages.

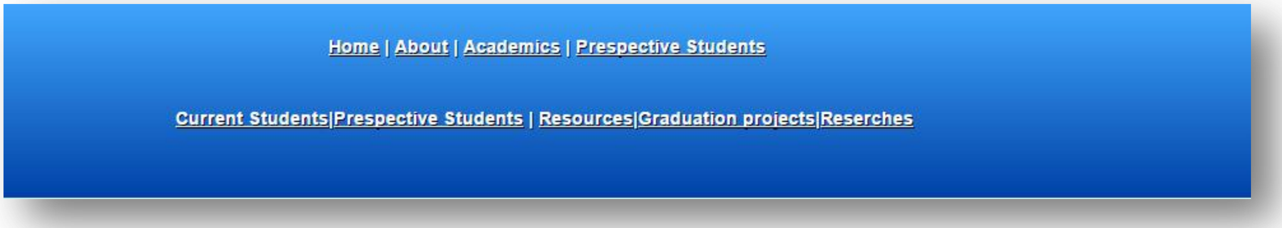
- Detailed requirements are transformed into detailed specifications for the system to guide the work of the Execute stage. For complex projects there may be an iterative relationship between the Design and Execute stages. Infrastructure architecture decisions are made to address how the system will meet the defined functional, physical, interface, information protection and data requirements

- In systems design, the design functions and operations are described in detail, including the design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts.

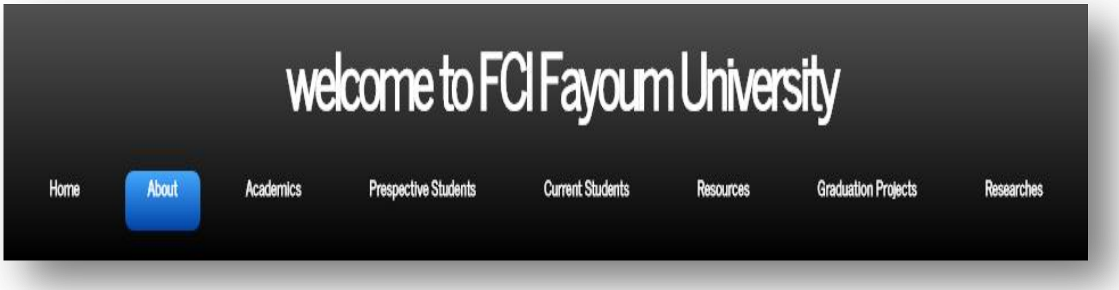
5.2 Design of system.

- System main functions.

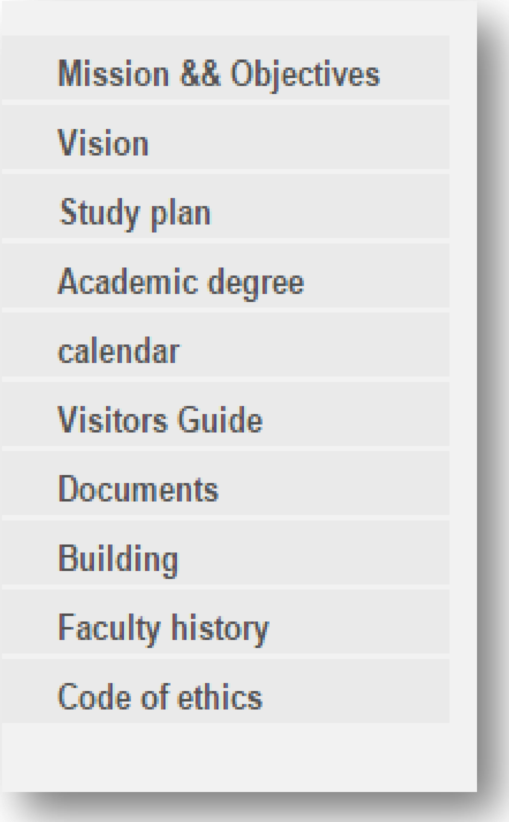



Home | About | Academics | Prespective Students
Current Students|Prespective Students | Resources|Graduation projects|Reserches

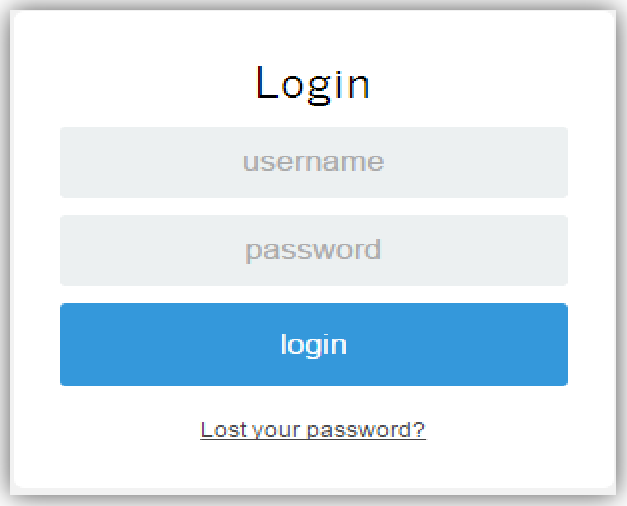
- About Function



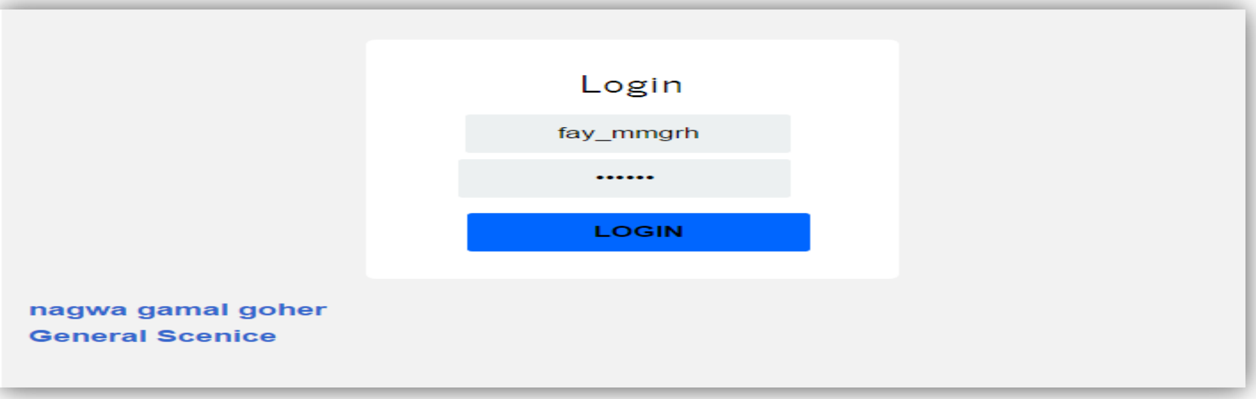
Main Functions inside About Function



Login Function



Registration function

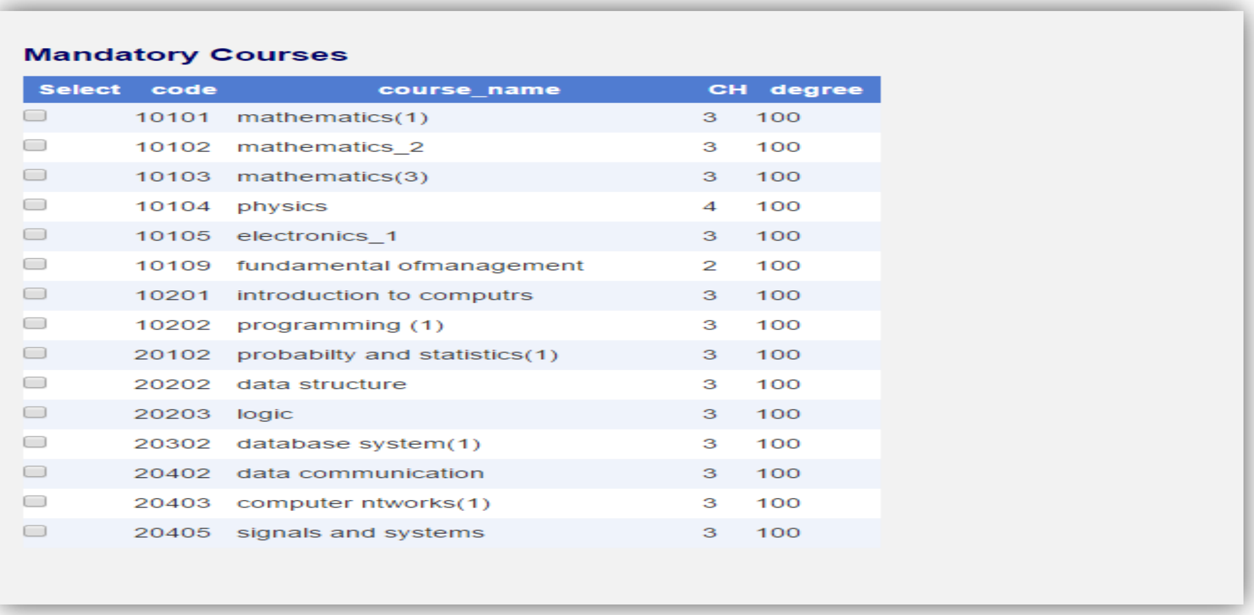


Login

fay_mmgrh

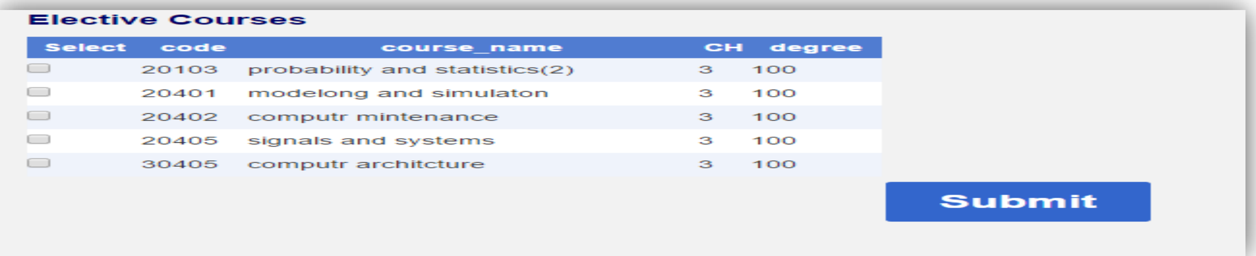
••••••

nagwa gamal goher
General Scenice



Mandatory Courses

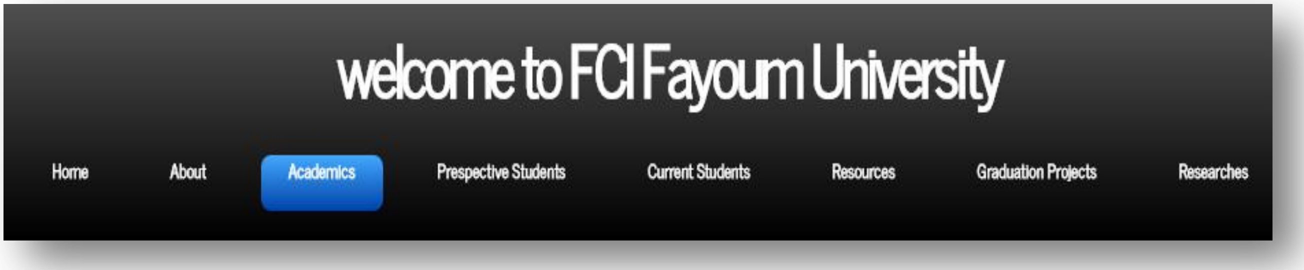
Select	code	course_name	CH	degree
□	10101	mathematics(1)	3	100
□	10102	mathematics_2	3	100
□	10103	mathematics(3)	3	100
□	10104	physics	4	100
□	10105	electronics_1	3	100
□	10109	fundamental ofmanagement	2	100
□	10201	introduction to computrs	3	100
□	10202	programming (1)	3	100
□	20102	probabilty and statistics(1)	3	100
□	20202	data structure	3	100
□	20203	logic	3	100
□	20302	database system(1)	3	100
□	20402	data communication	3	100
□	20403	computer ntworks(1)	3	100
□	20405	signals and systems	3	100



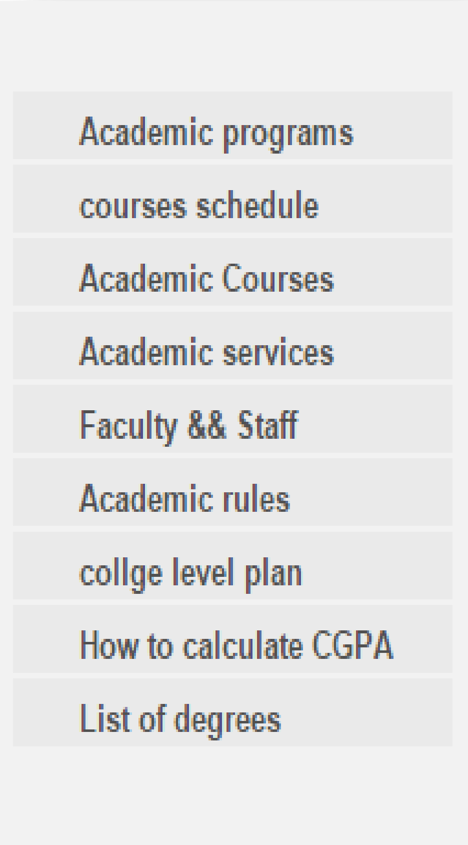
Elective Courses

Select	code	course_name	CH	degree
□	20103	probability and statistics(2)	3	100
□	20401	modelong and simulaton	3	100
□	20402	computr mintenance	3	100
□	20405	signals and systems	3	100
□	30405	computr architcture	3	100

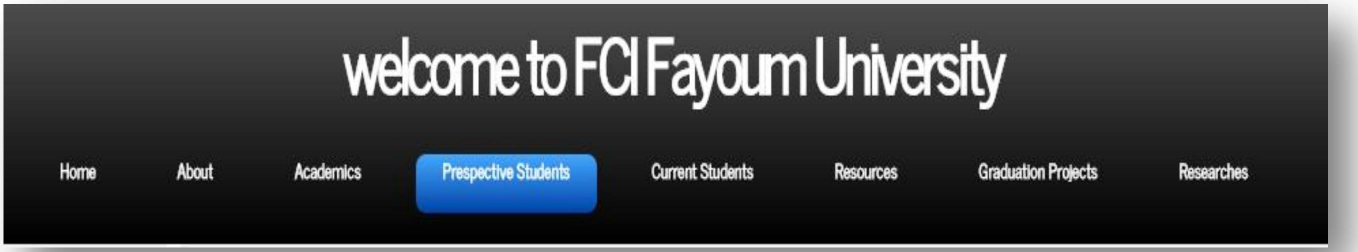
Academics function



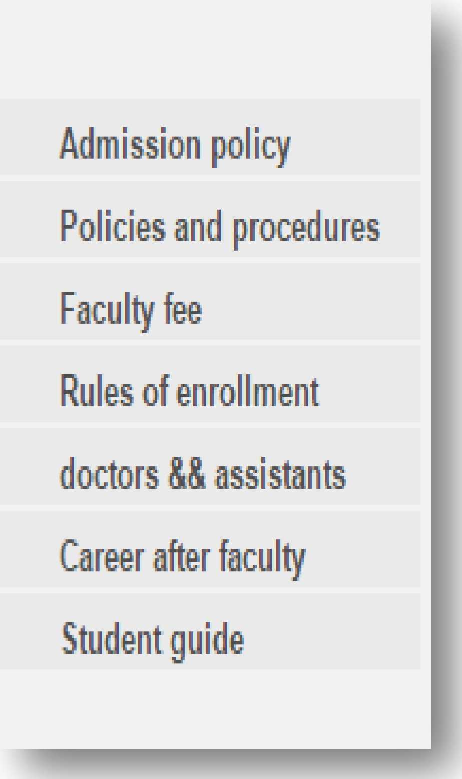
Main functions inside Academics



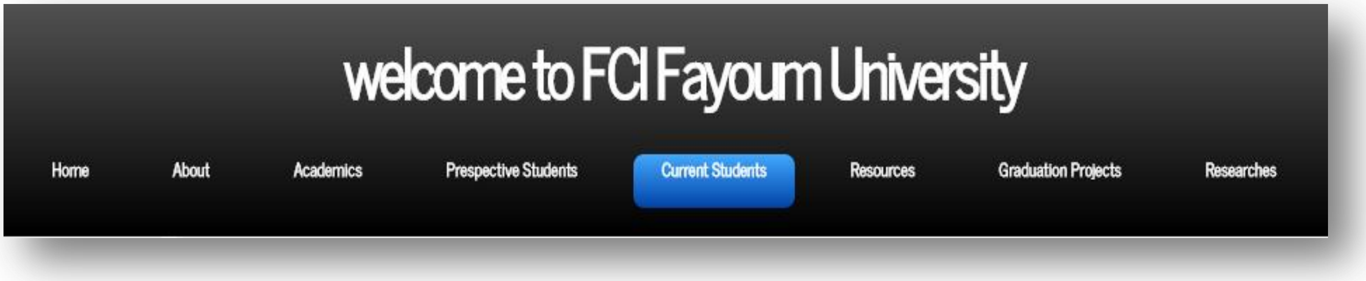
Perspective Students



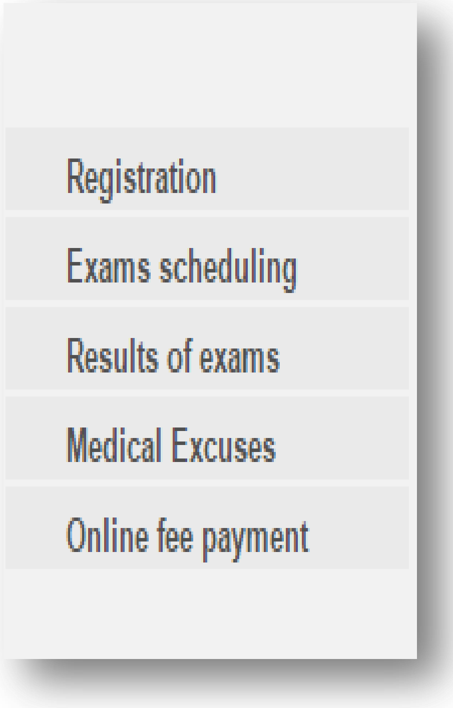
Main Functions inside Perspective Students



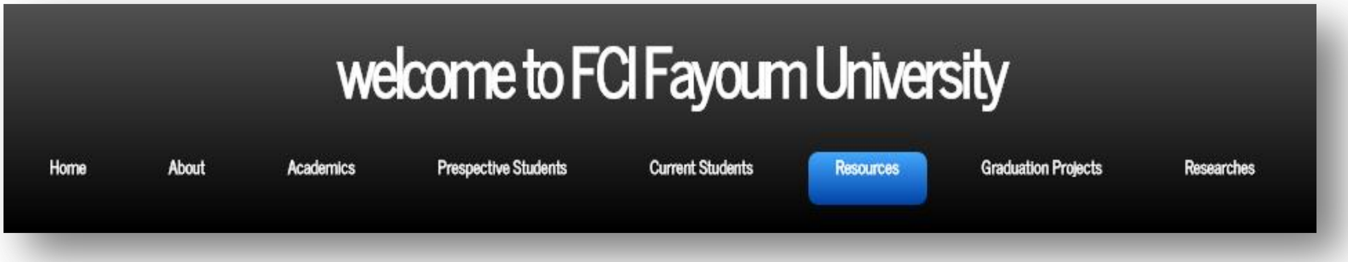
Current Student



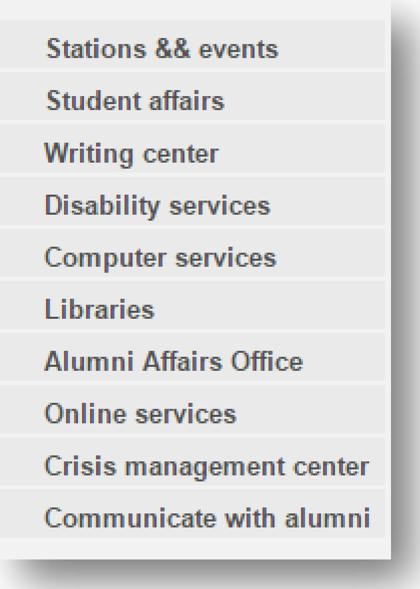
Main Functions inside Current Student



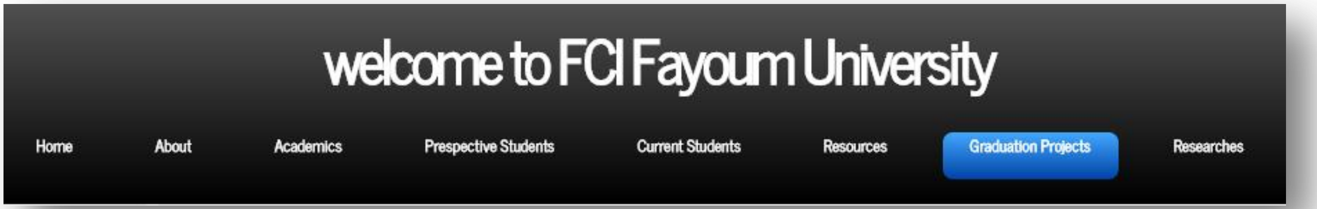
Resources



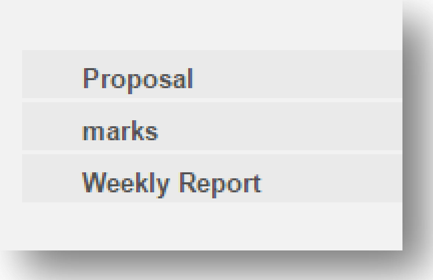
Main Functions inside Resources



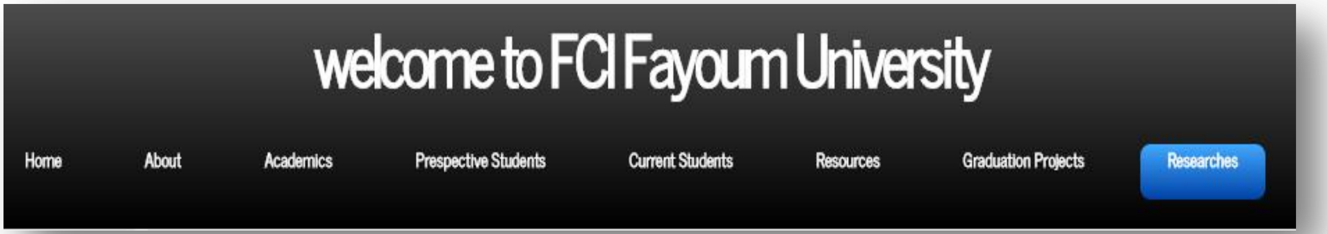
Graduation Project



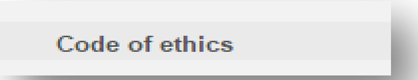
Main Functions inside Graduation Project



Researches



Main Functions inside Researches



5.3 Design of database.

5.3.1 ERD Diagram

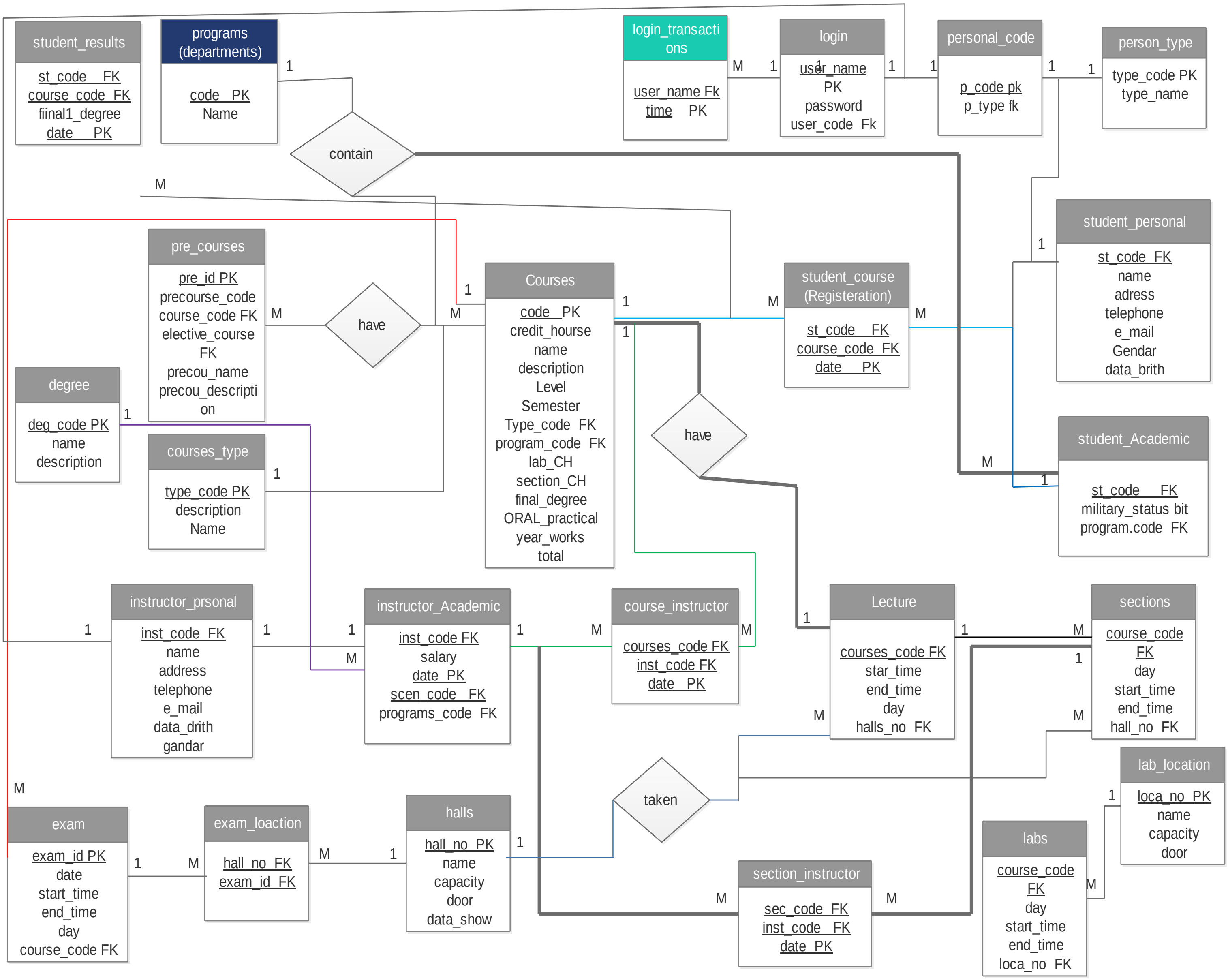


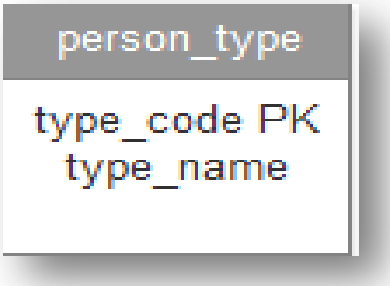
Figure (5.1)

5.3.2 Description of ERD Tables:

The first step is determination what tables we need on our system' data base to satisfy the objectives and functions of the system, Also determine which attributes tables need and it's criteria.

On next sections we show these tables.

1. **Personal type Table**

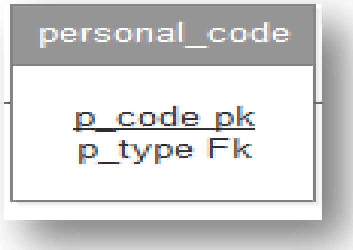


person_type

type_code PK
type_name

- Personal type consists of one primary key (type code).
- Column to hold another data (type name)

2. **Personal code Table**

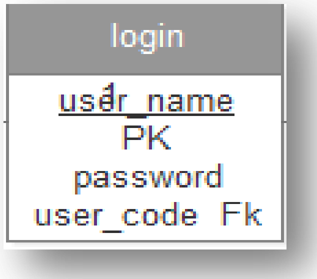


personal_code

p_code pk
p_type Fk

- Personal code consists of one primary key (p_code).
- Table consists of several of foreign key like (p_type)

3. Login Table

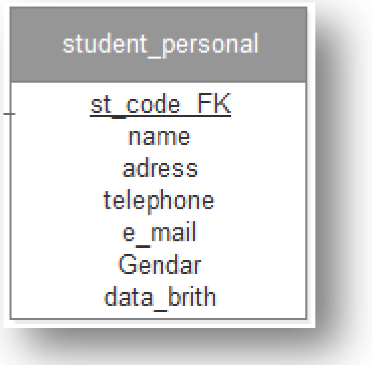


login

user_name
PK
password
user_code Fk

- Login consists of one primary key (user_name).
- Column to hold another data (password).
- Table consists of several of foreign key like (user_code)

4. Student_personal Table

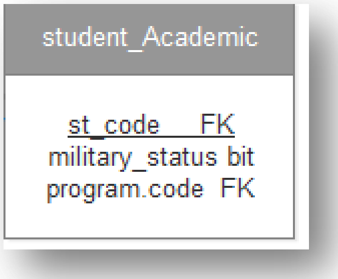


student_personal

st_code FK
name
adress
telephone
e_mail
Gendar
data_brith

- Student Table consists of several of foreign key like (st_code)
- Column to hold another data (Name,Address,Telephone,E_mail,Gender,Date_birth)

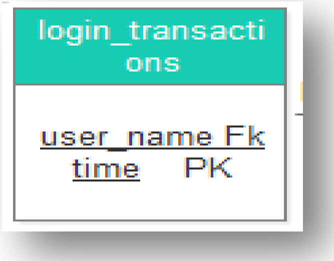
5. Student_Academic Table


student_Academic

st_code FK
military_status bit
program.code FK

- Table consists of two of foreign **key** like (St_code, program. Code).
- Column to hold another data (military_status).

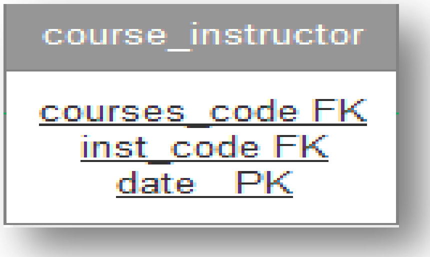
6. Login_transactions Table


login_transactions

user_name Fk
time PK

- Table t consists of one primary key (time).
- Table consists of several of foreign key like (user_name)

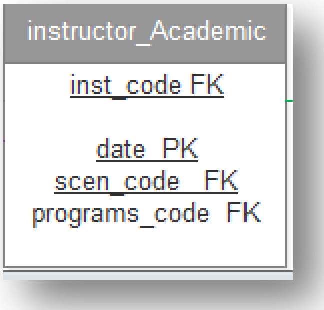
7. Course_Instructor Table


course_instructor

courses_code FK
inst_code FK
date PK

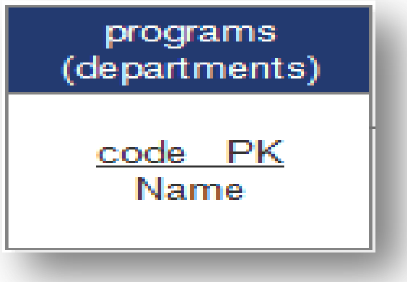
- Table consists of one primary key (date).
- Table consists of several of foreign key like (inst_code, courses_code).

8. Instructor_Academic Table



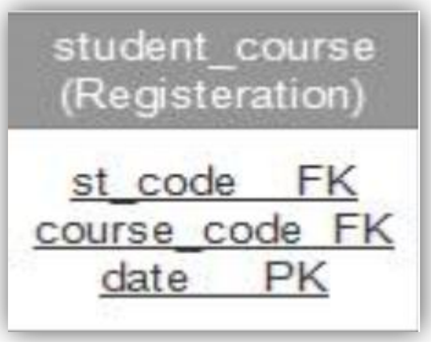
- Table consists of one primary key (date).
- Table consists of several of foreign key like (inst_code, scen_code, programs_code).

9. Programs (Department) Table



- Table consists of one primary key (code).
- Column to hold another data (Name).

10. Student_course (registration) table

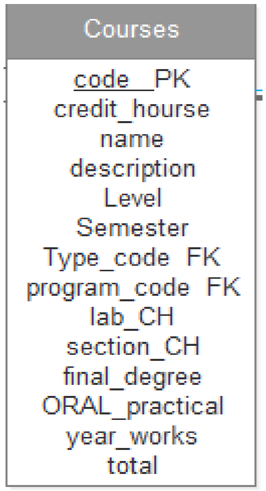


student_course
(Registeration)

st_code FK
course_code FK
date PK

- Table consists of one primary key (Date).
- table consists of several of foreign key like (st_code, course_code ,elective_course).

11. Courses Table

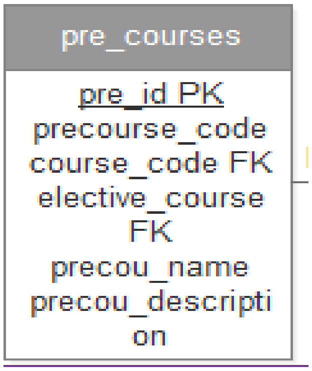


Courses

code PK
credit_hourse
name
description
Level
Semester
Type_code FK
program_code FK
lab_CH
section_CH
final_degree
ORAL_practical
year_works
total

- Table consists of one primary key (code).
- Table consists of several of foreign key like (Type_code, program_code).
- Column to hold another data (name, credit_hours, name, description, Level,Semester, final_degree,ORAL_practical,section_CH,year_works,total, lab_CH).

12. Pre_courses Table

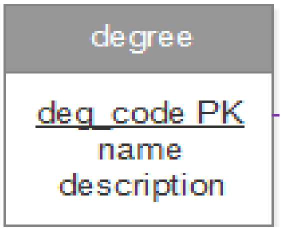


pre_courses

pre_id PK
precourse_code
course_code FK
elective_course FK
precou_name
precou_description

- Table consists of one primary key (pre_id).
- Table consists of several of foreign key like (course_code, elective_course,).
- Column to hold another data (precourse_code, precou_name,precou_description).

13. Degree Table

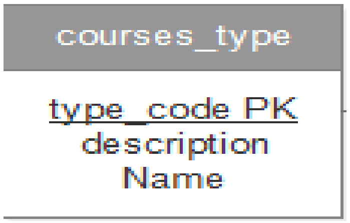


degree

deg_code PK
name
description

- Table consists of one primary key (deg_code).
- Column to hold another data (Name, Description).

14. Courses_TypeTable

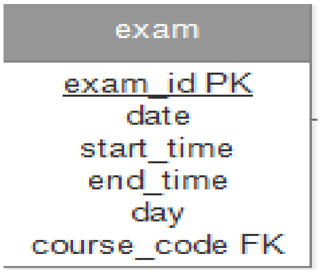


courses_type

type_code PK
description
Name

- Table consists of one primary key (Type_code).
- Column to hold another data (Name, Description).

15. Exam Table

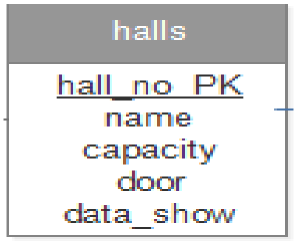


exam

exam_id PK
date
start_time
end_time
day
course_code FK

- Table consists of one primary key (exam_id).
- Column to hold another data (start_time,data , end_time , day).
- Consists of one foreign key like (Course_code).

16. Halls Table

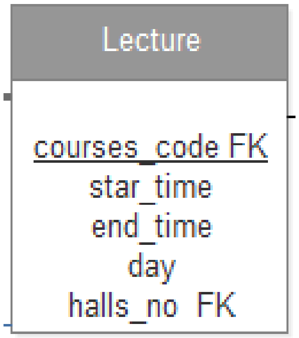


halls

hall_no PK
name
capacity
door
data_show

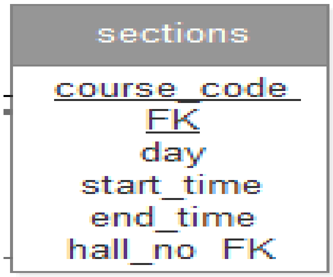
- Table consists of one primary key (hall_no).
- Column to hold another data (name,capacity , door, data show).

17. Lecture Table



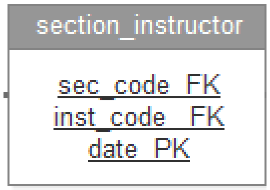
- Table has Column to hold another data (day, start_time, end_time).
- Consists of several of foreign key like (courses_code, halls no).

18. Sections Table



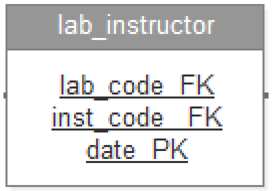
- Column to hold another data (Name, date, day, start_time, end_time).
- Table consists of several of foreign key like (course_code, hall_no).

19. Section_instructor Table



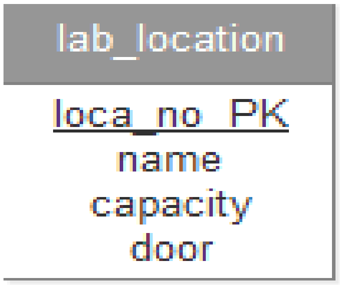
- Table consists of one primary key (date).
- Consists of several of foreign key like (sec_code , inst_code).

20. Lab_instructor Table



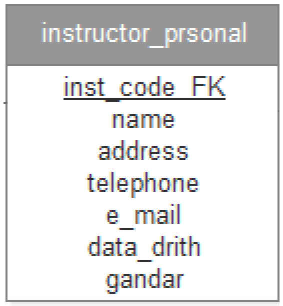
- Table consists of one primary key (date).
- Consists of several of foreign key like (lab_code, inst_code).

21. Lab Location Table



- Table consists of one primary key (location_no).
- Table has Column to hold another data (name,capacity,door).

22. Instructor_personal Table

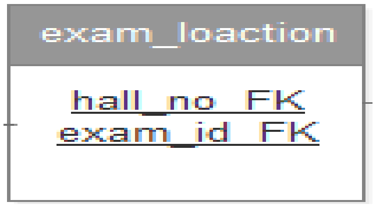


instructor_prsonal

inst_code FK
name
address
telephone
e_mail
data_drith
gandar

- Table consists of one primary key (inst_code).
- Table has Column to hold another data (name,address, telephone,e-mail, date_birth,gender).

23. Exam_location Table

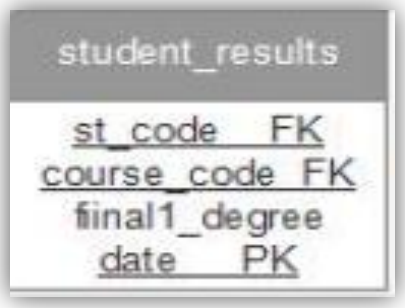


exam_loaction

hall_no FK
exam_id FK

- Consists of several of foreign key like (hall_no, exam_id).

24. Student_results Table

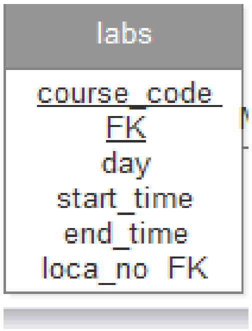


student_results

st_code FK
course_code FK
final1_degree
date PK

- Table consists of one primary key (data).
- Column to hold another data (final1_degree, final2_degree).
- Consists of several of foreign key like (st_code, course_code, elective_course).

25. Labs Table



labs

course_code FK
day
start_time
end_time
loca_no FK

- Column to hold another data (day, start_time, end_time).
- Consists of several of foreign key like (course_code, loca_no).

Chapter 6

"Implementation"

6.1Overview

6.2 Implementation of data base.

6.2.1 Creating data base tables

6.3 Implementation of website.

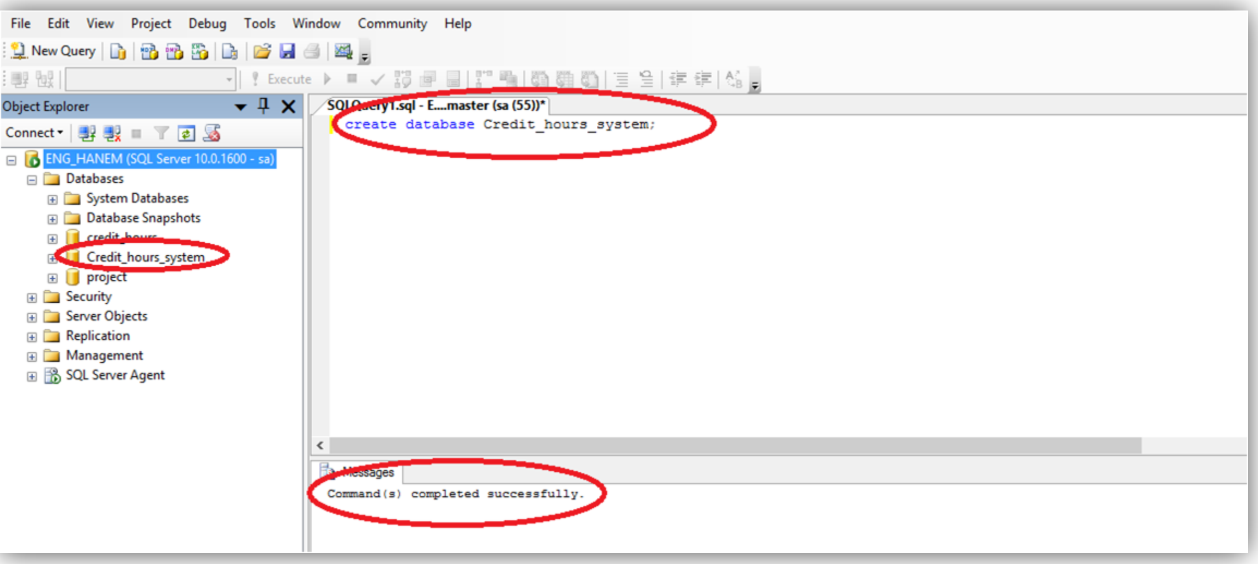
6.1Overview

The implementation phase comes after analysis and design phase in waterfall model. So, this chapter focuses on the implementation phase .The implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment.

6.2 Implementation of data base.

Implementation of data base by using SQL server 2008.

- First creating data base Credit_Hours_System.



```
create database Credit_hours_system;
```

```
Command(s) completed successfully.
```

Figure(6.1)

This shows that data base was created successfully and that appear in the previous figure (6.1)

The code is used to do that is :

Create database Credit_Hours_System ;

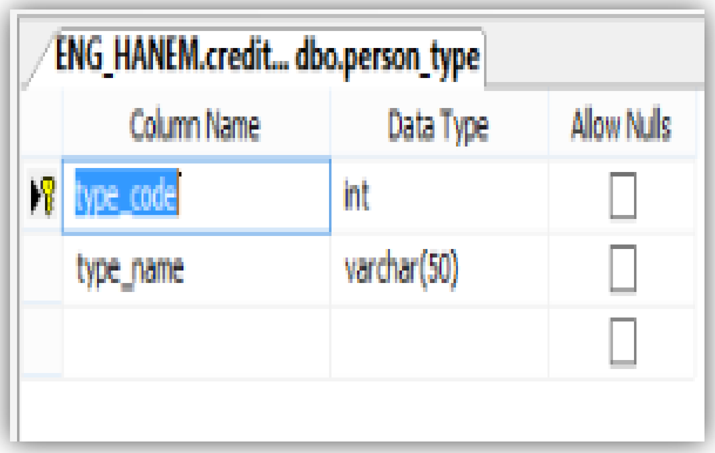
6.2.1 :Creating data base tables

1-create table person_type :

Code:

```
use credit_hours;

go

create table person_type

(

  type_code int not null unique,

  type_name varchar(50) not null,

  primary key(type_code)

)
```

Figure that shown that is correct:



Column Name	Data Type	Allow Nulls
type_code	int	□
type_name	varchar(50)	□
		□

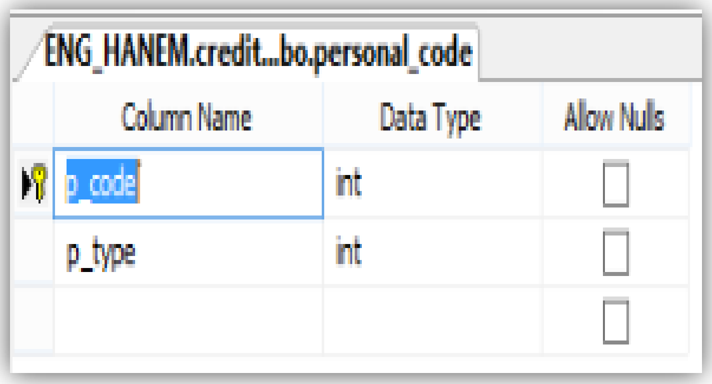
Figure(6.2)

2-create personal_code table

Code:

```
create table personal_code
(
  p_code int not null unique,
  p_type int not null,
  primary key(p_code),
  foreign key (p_type) references person_type(type_code)
)
```

Figure that shown that is correct:



Column Name	Data Type	Allow Nulls
p_code	int	□
p_type	int	□
		□

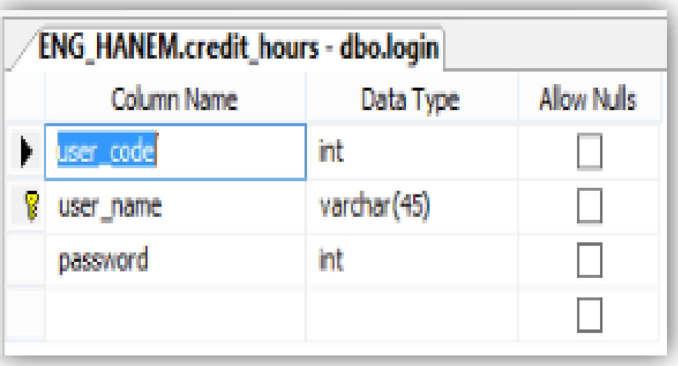
Figure (6.3)

3-create login table:

Code:

```
create table login
(
  user_code int not null unique,
  user_name varchar(45) not null unique,
  password int not null unique,
  primary key(user_name),
 foreign key (user_code) references personal_code(p_code)
)
```

Figure that shown that is correct:



ENG_HANEM.credit_hours - dbo.login

Column Name	Data Type	Allow Nulls
user_code	int	□
user_name	varchar(45)	□
password	int	□
		□

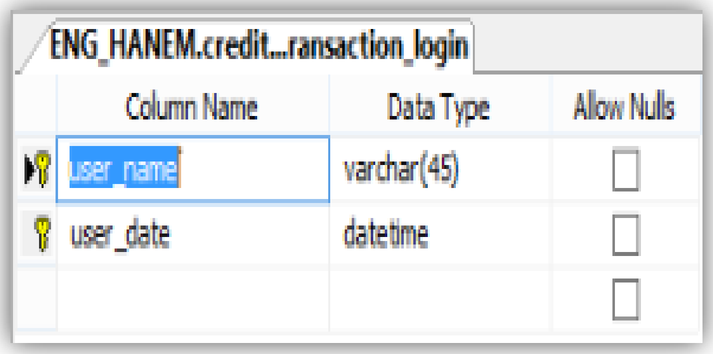
Figure (6.4)

4-create transaction_login table:

Code:

```
create table transaction_login
(
user_name varchar(45) not null unique,
user_date datetime default(getdate()),

primary key(user_name,user_date),

foreign key (user_name) references login(user_name)

)
```

Figure that shown that is correct:



ENG_HANEM.credit...ransaction_login

Column Name	Data Type	Allow Nulls
user_name	varchar(45)	□
user_date	datetime	□
		□

Figure (6.5)

5-create student_personaltable:

Code:

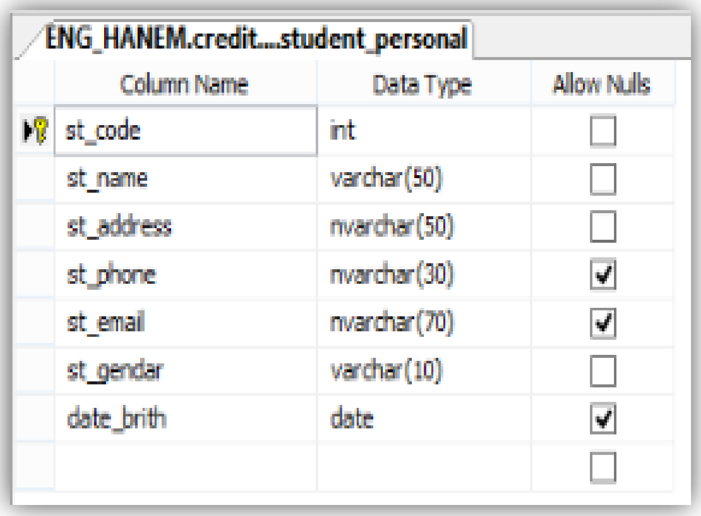
```
create table student_personal

(

st_code int not null unique,

st_name varchar(50) not null ,

st_address nvarchar(50) not null,

st_phone nvarchar(30) null,

st_email nvarchar(70) null,

st_gendar varchar(10) not null,

date_brith date,
```

primary key (st_code),

foreign key (st_code) references personal_code(p_code)

)

Figure that shown that is correct:



	Column Name	Data Type	Allow Nulls
▶🔑	st_code	int	□
	st_name	varchar(50)	□
	st_address	nvarchar(50)	□
	st_phone	nvarchar(30)	☑
	st_email	nvarchar(70)	☑
	st_gendar	varchar(10)	□
	date_brith	date	☑
			□

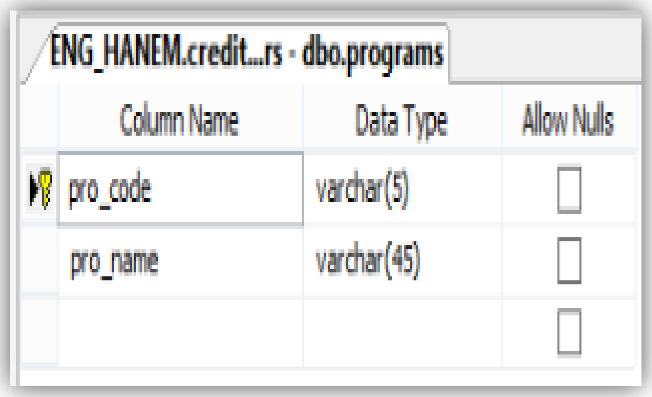
Figure (6.6)

6-create programs tables:

Code:

```
create table programs

(

pro_code varchar(5) not null unique,

pro_name varchar(45) not null,

primary key(pro_code)

)
```

Figure that shown that is correct:



	Column Name	Data Type	Allow Nulls
	pro_code	varchar(5)	☐
	pro_name	varchar(45)	☐
			☐

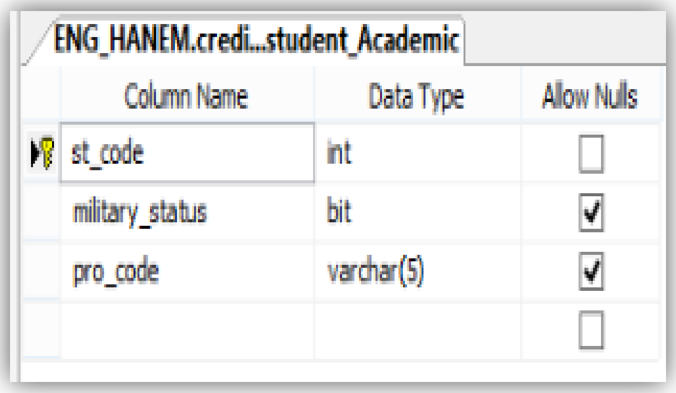
Figure (6.7)

7-create student_Academictable:

Code:

```
create table student_Academic

(

st_code int not null unique,

pro_code varchar(5) not null ,

military_status bit null,

primary key(st_code),

foreign key (st_code) references student_personal(st_code),

foreign key (pro_code) references programs(pro_code)

)
```

Figure that shown that is correct:



ENG_HANEM.credi...student_Academic

Column Name	Data Type	Allow Nulls
st_code	int	□
military_status	bit	☑
pro_code	varchar(5)	☑
		□

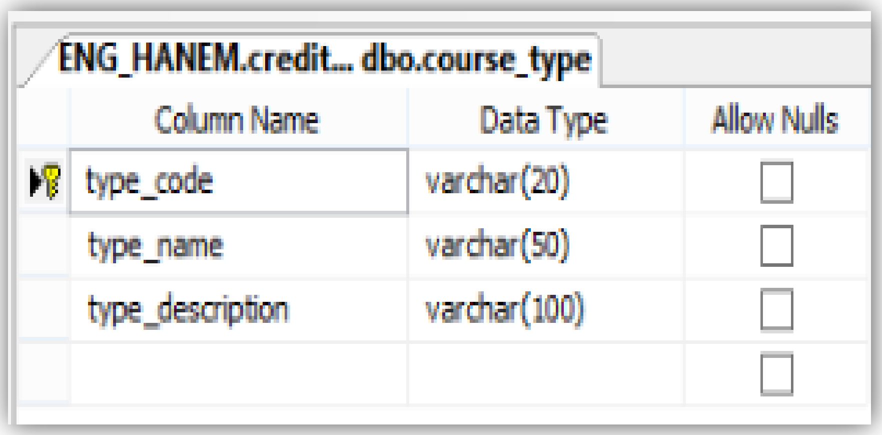
Figure (6.8)

8-create course_type table:

Code:

```
create table course_type
(
type_code varchar(20) not null unique,
type_name varchar(50) not null,
type_description varchar(100)not null,
primary key(type_code)
)
```

Figure that shown that is correct:



ENG_HANEM.credit... dbo.course_type

Column Name	Data Type	Allow Nulls
type_code	varchar(20)	□
type_name	varchar(50)	□
type_description	varchar(100)	□
		□

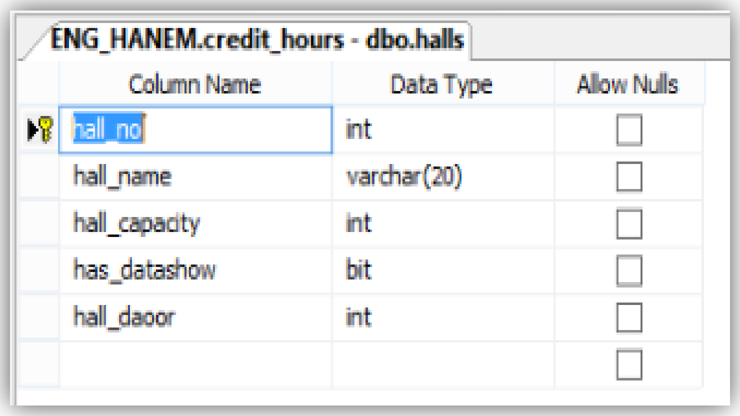
Figure (6.9)

9-create hallstable:

Code:

```
create table halls

(

hall_no int not null unique,

hall_name varchar(20) not null,

hall_capacity int not null,

has_datashow bit not null,

hall_daoor int not null,

primary key(hall_no)

)
```

Figure that shown that is correct:



ENG_HANEM.credit_hours - dbo.halls

Column Name	Data Type	Allow Nulls
hall_no	int	☐
hall_name	varchar(20)	☐
hall_capacity	int	☐
has_datashow	bit	☐
hall_daoor	int	☐
		☐

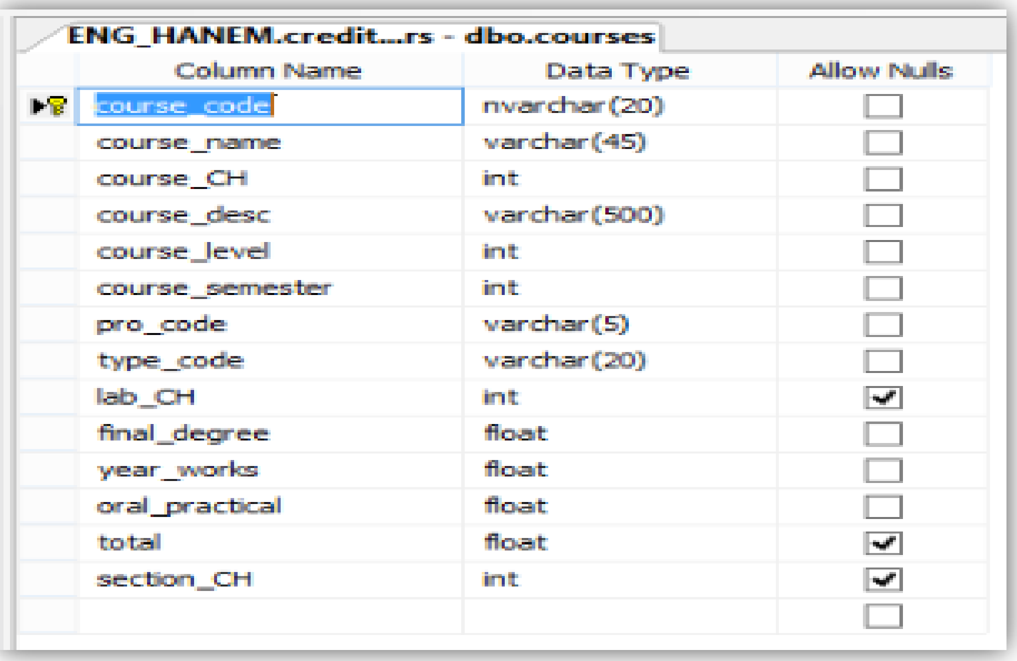
Figure (6.10)

10-create courses table:

Code:

```
create table courses

(

course_code nvarchar(20) not null unique,

course_name varchar(45) not null,

course_CH int not null,

course_desc varchar(500) not null,

course_level int not null,

course_semester int not null,

pro_code varchar(5) not null ,

type_code varchar(20) not null ,

lab_CH int null,

final_degree float not null,
year_works float not null,
oral_practical float not null,
section_CH int null,
primary key(course_code),
foreign key (pro_code) references programs(pro_code),
foreign key (type_code) references course_type(type_code)
)
```

Figure that shown that is correct:



ENG_HANEM.credit...rs - dbo.courses

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
course_name	varchar(45)	□
course_CH	int	□
course_desc	varchar(500)	□
course_level	int	□
course_semester	int	□
pro_code	varchar(5)	□
type_code	varchar(20)	□
lab_CH	int	☑
final_degree	float	□
year_works	float	□
oral_practical	float	□
total	float	☑
section_CH	int	☑
		□

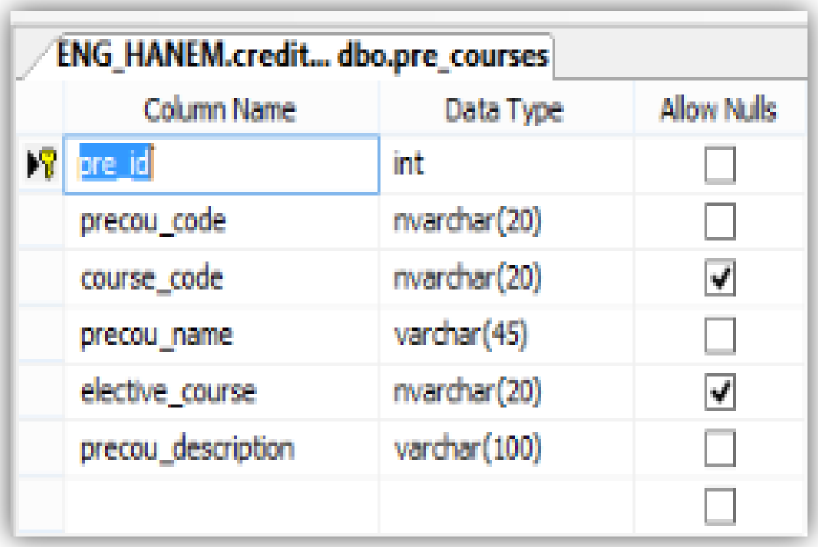
Figure (6.11)

11-create pre_coursestable:

Code:

```
create table pre_courses
(pre_id int not null unique,
precou_code nvarchar(20) not null ,
course_code nvarchar(20)  null ,
precou_name varchar(45) not null,
precou_description varchar(100) not null,
elective_course  nvarchar(20)  null ,
primary key(pre_id),
foreign key (course_code) references courses(course_code),
foreign key (elective_course) references elective_courses(course_code)
)
```

Figure that shown that is correct:



ENG_HANEM.credit... dbo.pre_courses

Column Name	Data Type	Allow Nulls
pre_id	int	□
precou_code	nvarchar(20)	□
course_code	nvarchar(20)	☑
precou_name	varchar(45)	□
elective_course	nvarchar(20)	☑
precou_description	varchar(100)	□
		□

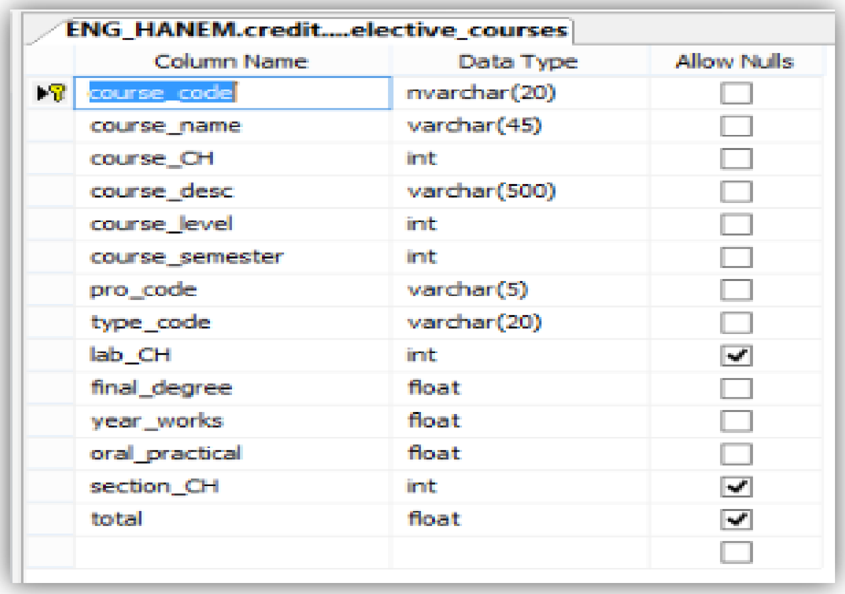
Figure (6.12)

12-create elective_courses table:

Code:

```
create table elective_courses

(

course_code nvarchar(20) not null unique,

course_name varchar(45) not null,

course_CH int not null,

course_desc varchar(500) not null,

course_level int not null,

course_semester int not null,

pro_code varchar(5) not null ,

type_code varchar(20) not null ,

lab_CH int null,

final_degree float not null,

year_works float not null,

oral_practical float not null,

section_CH int null,

primary key(course_code),

foreign key (pro_code) references programs(pro_code),

foreign key (type_code) references course_type(type_code)

)
```

Figure that shown that is correct:



ENG_HANEM.credit....elective_courses

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
course_name	varchar(45)	□
course_CH	int	□
course_desc	varchar(500)	□
course_level	int	□
course_semester	int	□
pro_code	varchar(5)	□
type_code	varchar(20)	□
lab_CH	int	☑
final_degree	float	□
year_works	float	□
oral_practical	float	□
section_CH	int	☑
total	float	☑
		□

Figure (6.13)

13-create Registerationtable:

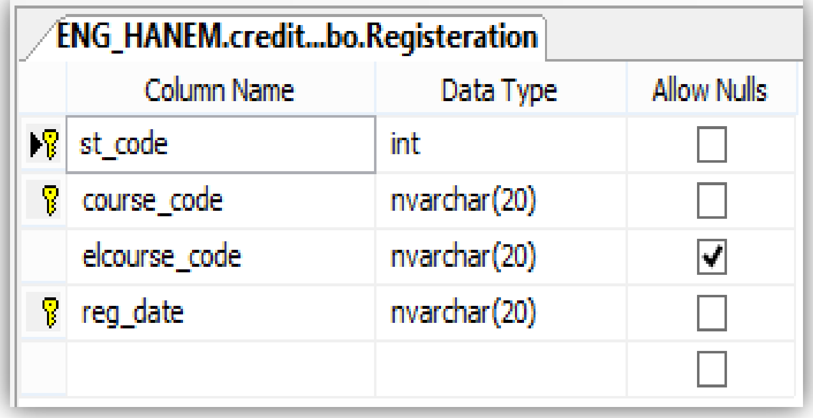
Code:

```
create table Registeration

(

st_code int not null unique,

course_code nvarchar(20) not null,

elcourse_code nvarchar(20) null,

reg_date nvarchar(20) not null ,

primary key(st_code,course_code,reg_date),

foreign key (st_code) references student_Academic(st_code),

foreign key (course_code) references courses(course_code),

foreign key (elcourse_code) references elective_courses(course_code),

constraint academic_year_R
```

check(reg_date like '[2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][-][2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9]')

)

Figure that shown that is correct:



ENG_HANEM.credit...bo.Registeration

Column Name	Data Type	Allow Nulls
st_code	int	□
course_code	nvarchar(20)	□
elcourse_code	nvarchar(20)	☑
reg_date	nvarchar(20)	□
		□

Figure (6.14)

14-create instructor_personaltable:

Code:

create table instructor_personal

(

inst_code int not null unique,

inst_name varchar(50) not null ,

inst_address nvarchar(50) not null,

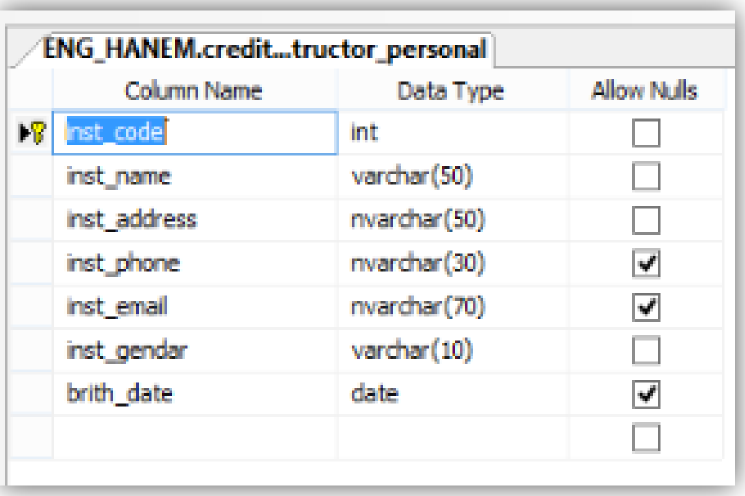
inst_phone nvarchar(30) null,

inst_email nvarchar(70) null,

inst_gendar varchar(10) not null,

```
brith_date date,

primary key (inst_code),

foreign key (inst_code) references personal_code(p_code)

)
```

Figure that shown that is correct:



ENG_HANEM.credit...tructor_personal

Column Name	Data Type	Allow Nulls
inst_code	int	☐
inst_name	varchar(50)	☐
inst_address	nvarchar(50)	☐
inst_phone	nvarchar(30)	☑
inst_email	nvarchar(70)	☑
inst_gendar	varchar(10)	☐
brith_date	date	☑
		☐

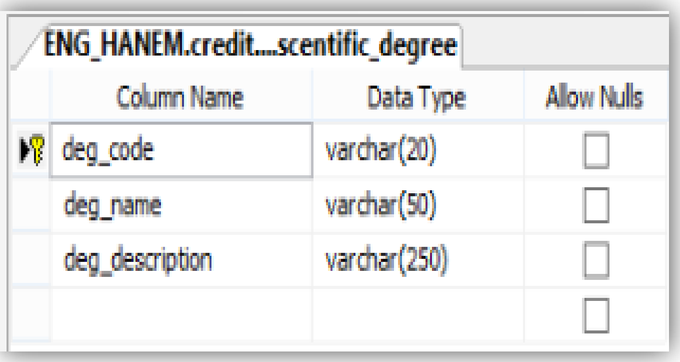
Figure (6.15)

15-create scentific_degree table:

Code:

```
create table scentific_degree

(

deg_code varchar(20) not null unique,

deg_name varchar(50) not null unique,

deg_description varchar(250) not null,

primary key(deg_code)

)
```

Figure that shown that is correct:



ENG_HANEM.credit....scentific_degree

	Column Name	Data Type	Allow Nulls
	deg_code	varchar(20)	□
	deg_name	varchar(50)	□
	deg_description	varchar(250)	□
			□

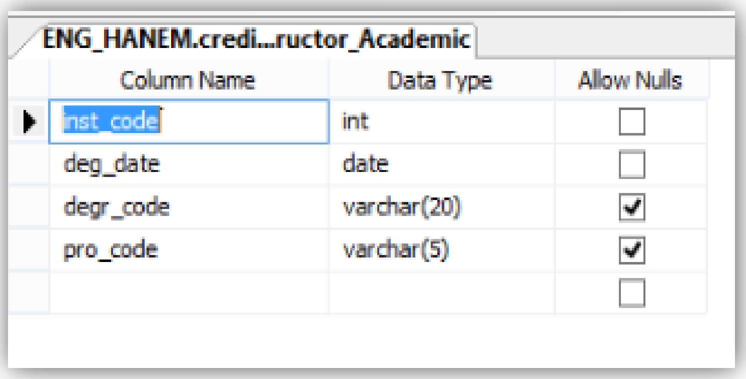
Figure (6.16)

16-create instructor_Academictable:

Code:

```
create table instructor_Academic

(

inst_code int not null unique,

deg_date date not null,

degr_code varchar(20) not null,

pro_code varchar(5) not null ,

primary key(inst_code,degr_code),

foreign key (inst_code) references instructor_personal(inst_code),

foreign key (degr_code) references scentific_degree(deg_code),

foreign key (pro_code) references programs(pro_code)

)
```

Figure that shown that is correct:



ENG_HANEM.credi...ructor_Academic

	Column Name	Data Type	Allow Nulls
▶	inst_code	int	□
	deg_date	date	□
	degr_code	varchar(20)	☑
	pro_code	varchar(5)	☑
			□

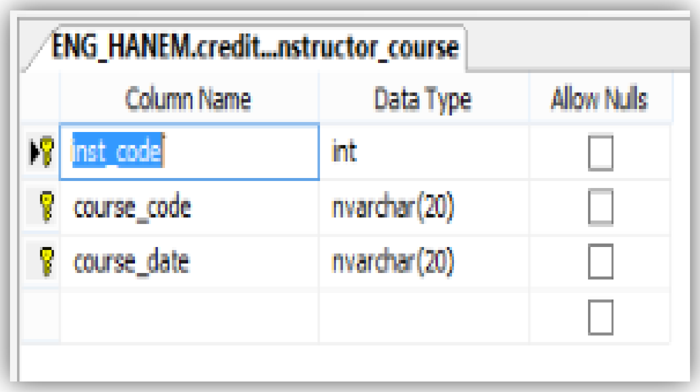
Figure (6.17)

17-create instructor_course table:

Code:

```
create table instructor_course
(
inst_code int not null unique,
course_code nvarchar(20) not null,
course_date nvarchar(20) not null ,
primary key(inst_code,course_code,course_date),
foreign key (inst_code) references instructor_Academic(inst_code),
foreign key (course_code) references courses(course_code),
constraint academic_year
check(course_date like '[2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][-][2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9]')
)
```

Figure that shown that is correct:



ENG_HANEM.credit...nstructor_course

	Column Name	Data Type	Allow Nulls
▶🔑	inst_code	int	□
🔑	course_code	nvarchar(20)	□
🔑	course_date	nvarchar(20)	□
			□

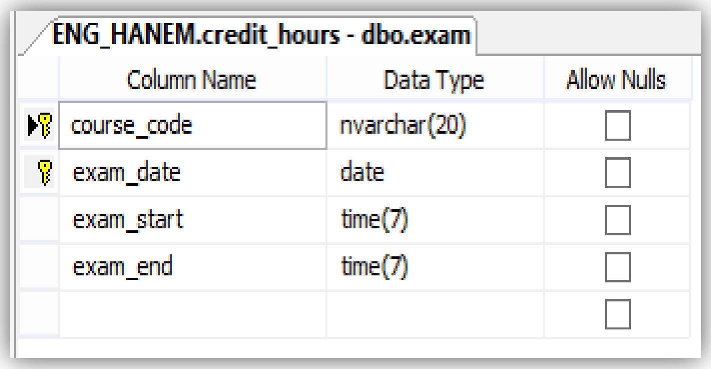
Figure(6.18)

18-create exam table:

Code:

```
create table exam

(

course_code nvarchar(20) not null unique,

exam_date date not null,

exam_start time not null,

exam_end time not null,

foreign key (course_code) references courses(course_code),

primary key(course_code ,exam_date)

)
```

Figure that shown that is correct:



ENG_HANEM.credit_hours - dbo.exam

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
exam_date	date	□
exam_start	time(7)	□
exam_end	time(7)	□
		□

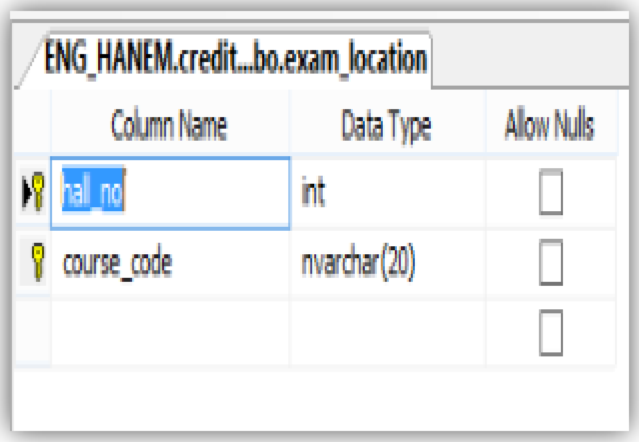
Figure (6.19)

19-create exam_locationtable:

Code:

```
create table exam_location
(
hall_no int not null unique,
course_code nvarchar(20) not null unique,
primary key(hall_no,course_code),
foreign key (hall_no) references halls(hall_no),
foreign key (course_code) references courses(course_code)
)
```

Figure that shown that is correct:



ENG_HANEM.credit...bo.exam_location

Column Name	Data Type	Allow Nulls
hall_no	int	□
course_code	nvarchar(20)	□
		□

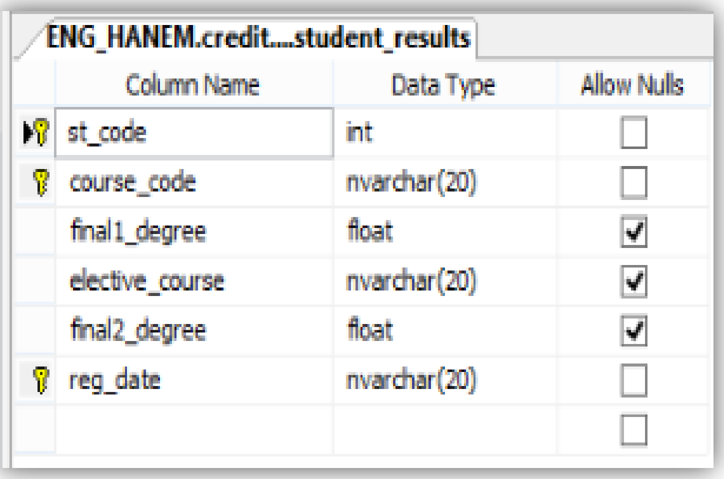
Figure (6.20)

20-create student_resultstable:

Code:

```
create table student_results

(

st_code int not null ,

course_code nvarchar(20)not null,

final1_degree float  null,

elective_course nvarchar(20) null,

final2_degree float  null,

reg_date nvarchar(20) not null ,

foreign key (course_code) references courses(course_code),

foreign key (st_code) references student_Academic(st_code),

foreign key (elective_course) references elective_courses(course_code),

primary key(st_code,course_code,reg_date),
constraint academic_year_d

check(reg_date like '[2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][-][2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9]')

)
```

Figure that shown that is correct:



ENG_HANEM.credit....student_results

Column Name	Data Type	Allow Nulls
st_code	int	☐
course_code	nvarchar(20)	☐
final1_degree	float	☑
elective_course	nvarchar(20)	☑
final2_degree	float	☑
reg_date	nvarchar(20)	☐
		☐

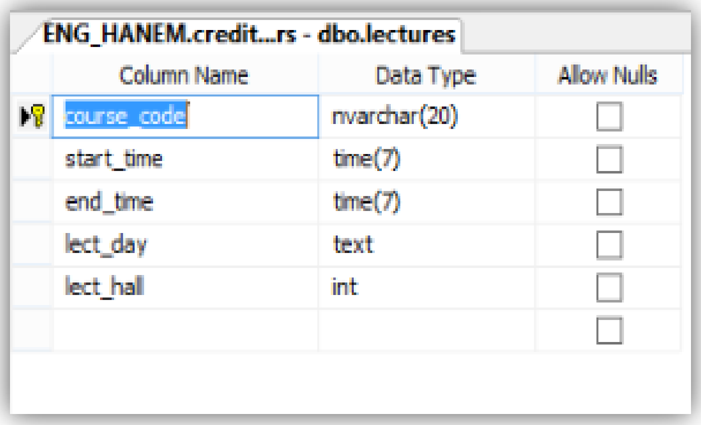
Figure (6.21)

21-create lecturestable:

Code:

```
create table lectures

(

course_code nvarchar(20) not null ,

start_time time not null,

end_time time not null,

lect_day text not null,

lect_hall int not null ,

primary key(course_code),
foreign key (course_code) references courses(course_code),

foreign key (lect_hall) references halls(hall_no)

)
```

Figure that shown that is correct:



ENG_HANEM.credit...rs - dbo.lectures

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
start_time	time(7)	□
end_time	time(7)	□
lect_day	text	□
lect_hall	int	□
		□

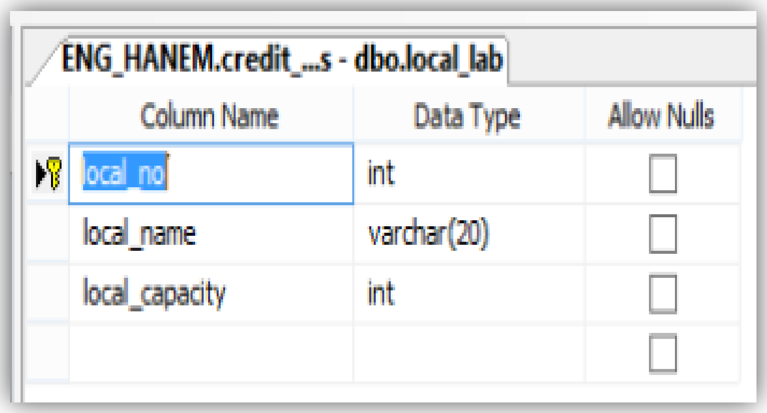
Figure (6.22)

22-create local_labtable:

Code:

```
create table local_lab

(

local_no int not null unique,

local_name varchar(20) not null,

local_capacity int not null,

primary key(local_no)

)
```

Figure that shown that is correct:



ENG_HANEM.credit_...s - dbo.local_lab

	Column Name	Data Type	Allow Nulls
	local_no	int	□
	local_name	varchar(20)	□
	local_capacity	int	□
			□

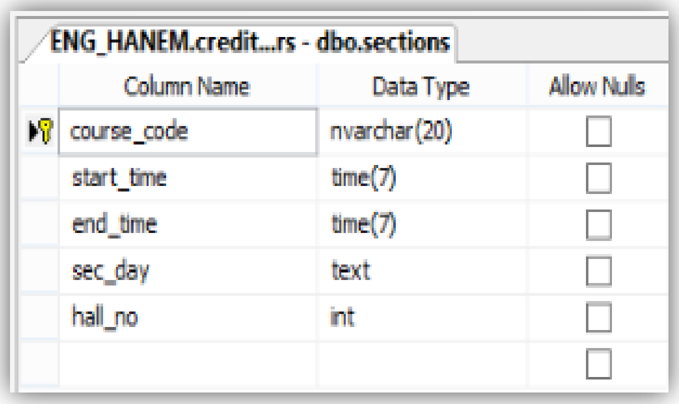
Figure (6.23)

23-create sectionstable:

Code:

```
create table sections

(

course_code nvarchar(20) not null unique,

start_time time not null,

end_time time not null,

sec_day text not null,

hall_no int not null,

primary key(course_code),

foreign key (course_code) references courses(course_code),

foreign key (hall_no) references halls(hall_no)

)
```

Figure that shown that is correct:



ENG_HANEM.credit...rs - dbo.sections

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
start_time	time(7)	□
end_time	time(7)	□
sec_day	text	□
hall_no	int	□
		□

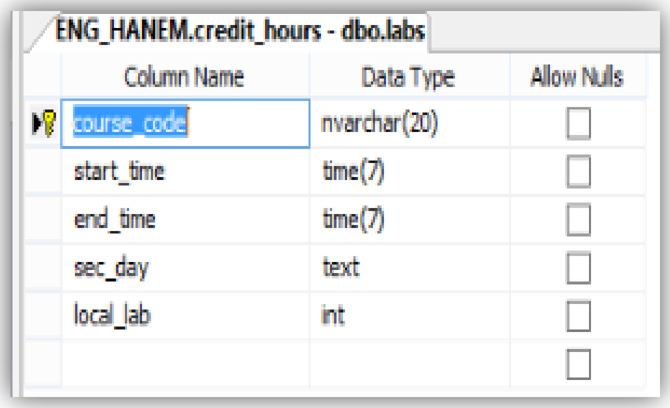
Figure (6.24)

24-create labs table

Code:

```
create table labs

(

course_code nvarchar(20) not null unique,

start_time time not null,

end_time time not null,

sec_day text not null,

local_lab int not null,

primary key(course_code),

foreign key (course_code) references courses(course_code),

foreign key (local_lab) references local_lab(local_no)

)
```

Figure that shown that is correct:



ENG_HANEM.credit_hours - dbo.labs

Column Name	Data Type	Allow Nulls
course_code	nvarchar(20)	□
start_time	time(7)	□
end_time	time(7)	□
sec_day	text	□
local_lab	int	□
		□

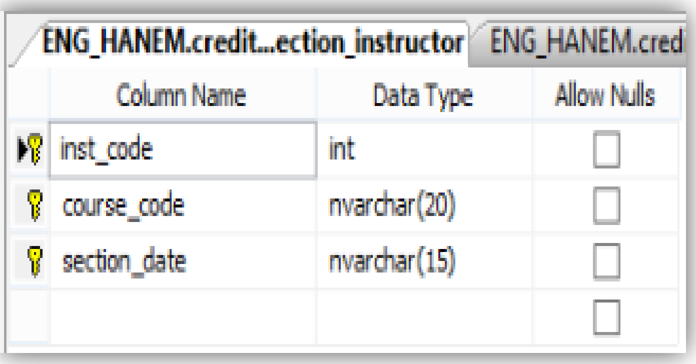
Figure (6.25)

25-create section_instructor table

Code:

```
create table section_instructor

(

inst_code int not null unique,

course_code nvarchar(20) not null ,

section_date nvarchar(15) not null ,

primary key(inst_code,course_code,section_date),

foreign key (inst_code) references instructor_Academic(inst_code),

foreign key (course_code) references courses(course_code),

constraint academic_year_section

check(section_date like '[2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][-][2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9]')

)
```

Figure that shown that is correct:



ENG_HANEM.credit...ection_instructor | ENG_HANEM.cred

	Column Name	Data Type	Allow Nulls
🔑	inst_code	int	☐
🔑	course_code	nvarchar(20)	☐
🔑	section_date	nvarchar(15)	☐
			☐

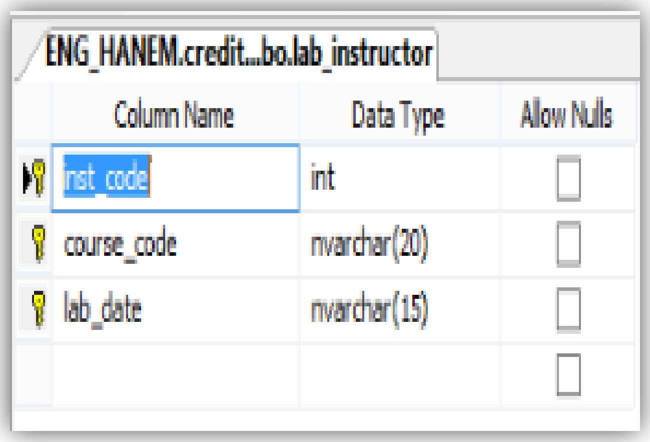
Figure (6.26)

26-create lab_instructor table

Code:

```
create table lab_instructor
(
inst_code int not null unique,
course_code nvarchar(20) not null ,
lab_date nvarchar(15) not null ,
primary key(inst_code,course_code,lab_date),
foreign key (inst_code) references instructor_Academic(inst_code),
foreign key (course_code) references courses(course_code),
constraint academic_year_lab
check(lab_date like '[2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][-][2 3 4][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9][0 1 2 3 4 5 6 7 8 9]')
)
```

Figure that shown that is correct:



ENG_HANEM.credit...bo.lab_instructor

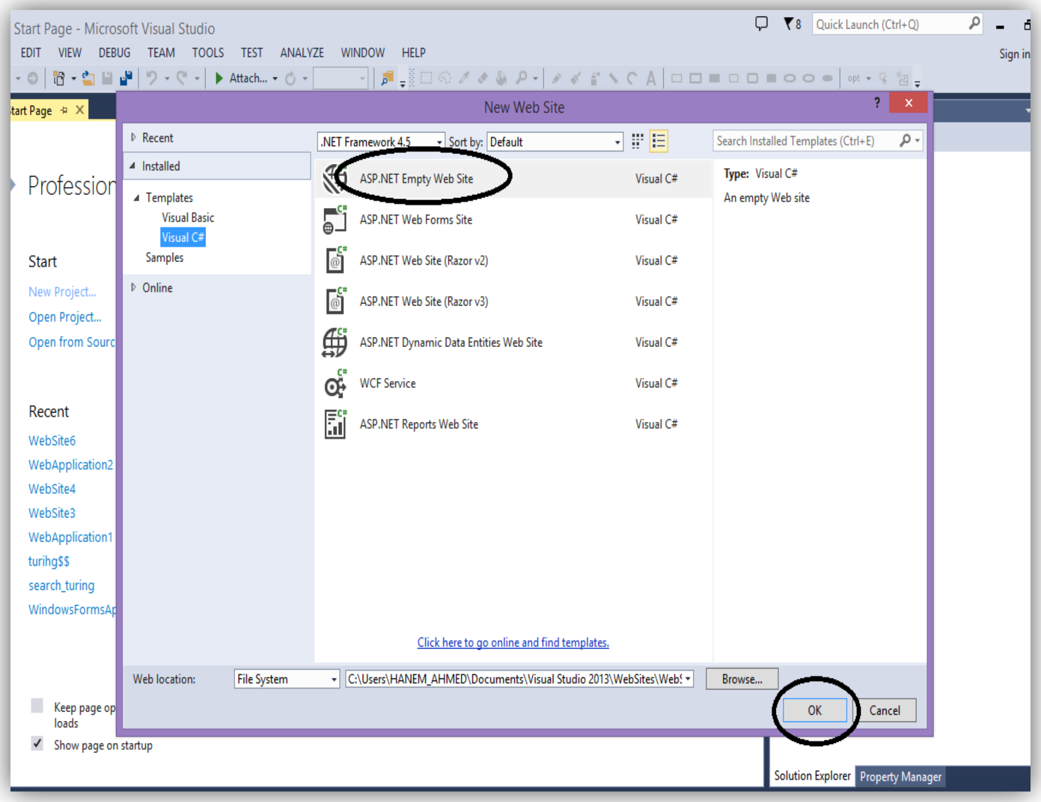
Column Name	Data Type	Allow Nulls
inst_code	int	□
course_code	nvarchar(20)	□
lab_date	nvarchar(15)	□
		□

Figure (6.27)

6.3 Implementation of website.

The site implemented by ASP.NET, C# in visual studio 2013.

First: creating ASP project web site:



Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
EDIT VIEW DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Sign in
Attach...
tart Page
Profession
Start
New Project...
Open Project...
Open from Sourc
Recent
WebSite6
WebApplication2
WebSite4
WebSite3
WebApplication1
turihg$$
search_turing
WindowsFormsAp
Keep page op loads
Show page on startup
New Web Site
Recent
Installed
Templates
Visual Basic
Visual C#
Samples
Online
Search Installed Templates (Ctrl+E)
ASP.NET Empty Web Site Visual C#
ASP.NET Web Forms Site Visual C#
ASP.NET Web Site (Razor v2) Visual C#
ASP.NET Web Site (Razor v3) Visual C#
ASP.NET Dynamic Data Entities Web Site Visual C#
WCF Service Visual C#
ASP.NET Reports Web Site Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel
Solution Explorer
Property Manager

.NET Framework 4.5 Sort by: Default

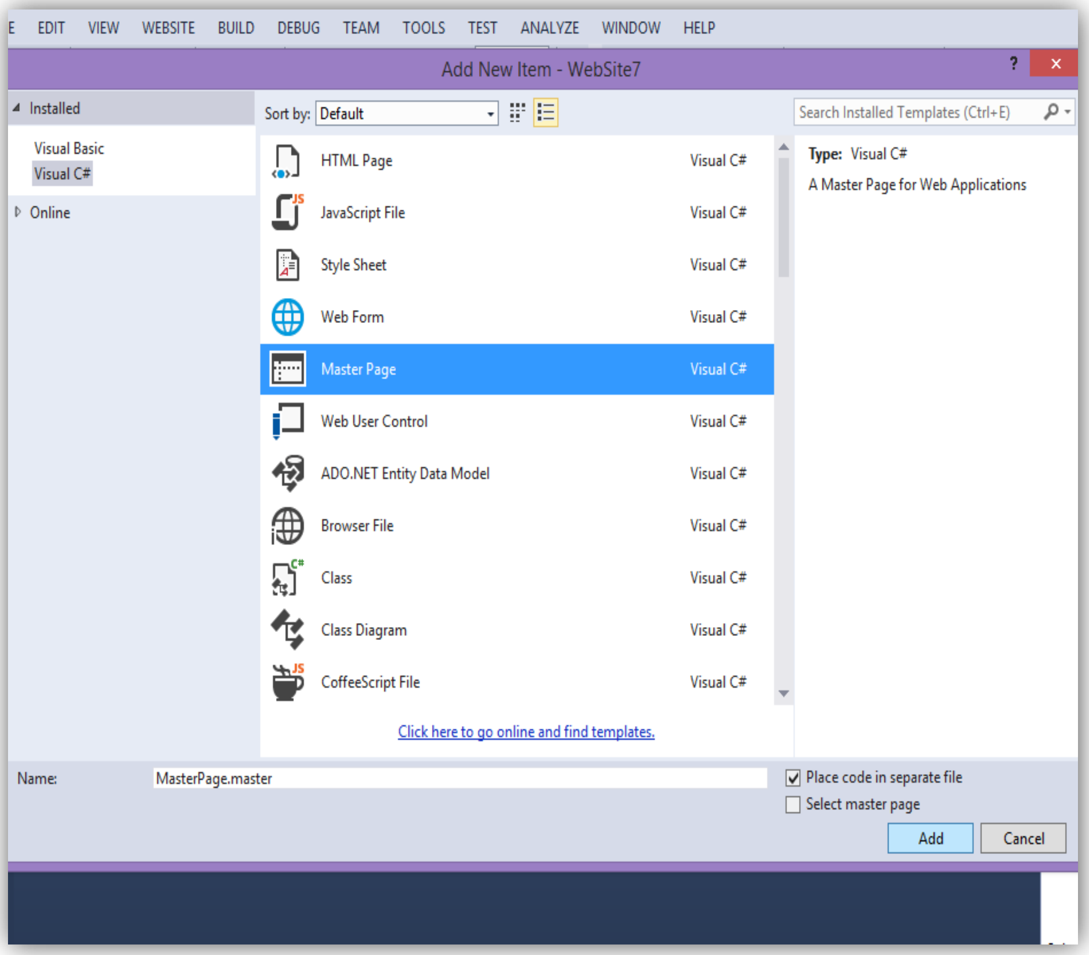
Type: Visual C#

An empty Web site

Web location: File System C:\Users\HANEM_AHMED\Documents\Visual Studio 2013\WebSites\Web!

Figure (6.28)

Second: creating master page:


E EDIT VIEW WEBSITE BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Add New Item - WebSite7
?
×
Installed
Visual Basic
Visual C#
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
HTML Page Visual C#
JavaScript File Visual C#
Style Sheet Visual C#
Web Form Visual C#
Master Page Visual C#
Web User Control Visual C#
ADO.NET Entity Data Model Visual C#
Browser File Visual C#
Class Visual C#
Class Diagram Visual C#
CoffeeScript File Visual C#
Click here to go online and find templates.
Add
Cancel

Type: Visual C#

A Master Page for Web Applications

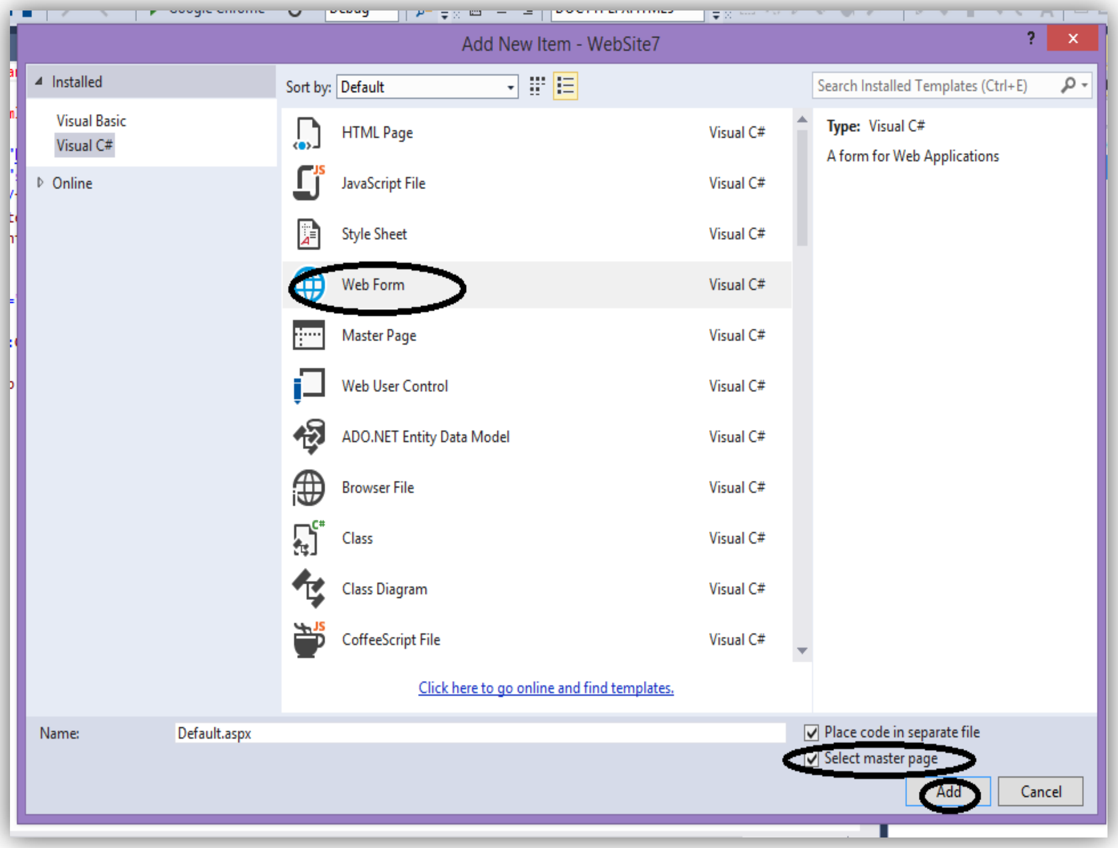
Name: MasterPage.master

☑ Place code in separate file

□ Select master page

Figure (6.29)

Third: creating web form page inherit from master page:



Type: Visual C#

A form for Web Applications

Name: Default.aspx

☑ Place code in separate file

☑ Select master page

Figure (6.30)

Fourth: implementing web site functions:

1. Code of main functions and general form of the system:-

```
<divid="main">

<header>
     <divid="strapline">
     <divid="welcome_slogan">
     <h3>welcome to FCI Fayoum   University</h3>
     </div><!--close welcome_slogan-->
</div><!--close strapline-->
     <nav>
     <divid="menubar">
```

```
<ulid="nav">
<liclass="current"><ahref="index.aspx">Home</a></li>
<li><ahref="About.aspx">About</a></li>
<li><ahref="Academics.aspx">Academics</a></li>
<li><ahref="Prospective students.aspx">Prespective Students</a></li>
<li><ahref="Current students.aspx">Current Students</a></li>
<li><ahref="Resources.aspx">Resources</a></li>
<li><ahref="Graduation Projects.aspx">Graduation Projects</a></li>
<li><ahref="Researches.aspx">Researches</a></li>
</ul>
</div><!--close menubar-->
</nav>
</header>

<div>
<asp:ContentPlaceHolderid="ContentPlaceHolder1"runat="server">

</asp:ContentPlaceHolder>
</div>



<divid="footer">
<ahref="index.aspx">Home</a> | <ahref="About.aspx">About</a> |
<ahref="Academics.aspx">Academics</a> | <ahref="Prespective
Students.aspx">Prespective Students</a><br/><br/>
<ahref="Current Students.aspx">Current
Students</a>|<ahref="Prespective Students.aspx">Prespective
Students</a> |
<ahref="Resources.aspx">Resources</a>|<ahref="Graduation
projects.aspx">Graduation
projects</a>|<ahref="Reserches.aspx">Reserches</a><br/><br/>

</div>

</div><!--close main-->
```

2. Code of left menu of main functions:-

2.1. About function

```
<divid="leftnav">
<h2></h2>
<ulid="leftnavmenu">
<li><ahref="About.aspx">Mission && Objectives</a></li>
<li><ahref="vision.aspx">Vision</a></li>
<li><ahref="study plan.aspx">Study plan</a></li>
```

```
<li><ahref="academic degree.aspx">Academic degree</a></li>
<li><ahref="#"> calendar </a></li>
<li><ahref="#">Visitors Guide</a></li>
<li><ahref="documents.aspx"> Documents</a></li>
<li><ahref="#"> Building</a></li>
<li><ahref="#"> Faculty history </a></li>
<li><ahref="#"> Code of ethics</a></li>
</ul>

<div/>
```

2.1.1. Study plan functions:

First: make database connection to display departments of faculty.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
publicpartialclassstudy_plan : System.Web.UI.Page
{
protectedvoid Page_Load(object sender, EventArgs e)
    {
SqlConnection conn = newSqlConnection();
        conn.ConnectionString = "data source=.;initial
catalog=credit_hours ; password=root;integrated security=true";
SqlCommand comm = newSqlCommand();
        comm.CommandText = "select pro_code as Code ,pro_name as Name
from programs";
        comm.Connection = conn;
conn.Open();
SqlDataReader reader = comm.ExecuteReader();
        GridView1.DataSource = reader;
GridView1.DataBind();
conn.Close();
    }
}
```

Second: in this function write the rules of new system that apply in faculty.

2.1.2. Documents function (Login).

```
SqlConnection conn = newSqlConnection();
         conn.ConnectionString = 
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
SqlCommand comm = newSqlCommand();
         comm.CommandText = "Select user_name ,password from login 
where user_name='" +   TextBox2.Text + "' and  password ='" + 
TextBox1.Text + "'";
        comm.Connection = conn;
conn.Open();
SqlDataReader reader = comm.ExecuteReader();
if(reader.Read())
          {Response.Redirect("");}
else
          {Response.Write("<script>alert('Please enter valid Username 
and   Password')</script>");}

conn.Close();
```

2.2. Academic function.

2.2.1 .Academic course scheduling.

```
protectedvoid Page_Load(object sender, EventArgs e)
    {
SqlConnection conn = newSqlConnection();
        conn.ConnectionString = 
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
SqlCommand comm = newSqlCommand();
```

```
    comm.CommandText = "select courses.course_name as Name ,
courses.course_level as Level ,courses.course_semester as
Semester,courses.course_CH as CH, instructor_personal.inst_name as
prof_Name,instructor_course.course_date as YEAR from courses JOIN
instructor_course ON courses.course_code=instructor_course.course_code
JOIN instructor_Academic ON
instructor_course.inst_code=instructor_Academic.inst_code JOIN
instructor_personal ON
instructor_Academic.inst_code=instructor_personal.inst_code  ";
        comm.Connection = conn;
conn.Open();
SqlDataReader reader = comm.ExecuteReader();
        GridView1.DataSource = reader;
GridView1.DataBind();
conn.Close();
    }
```

2.2.2. Faculty &&Staff.

```
protectedvoid Page_Load(object sender, EventArgs e)
    {
SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
ConfigurationManager.ConnectionStrings["MyConnectionString"]
.ToString();
        conn.Open();
        SqlCommand comm = new SqlCommand();
        comm.CommandText = "Select
instructor_personal.inst_name as
Pro_Name,scentific_degree.deg_name as
degree,programs.pro_name as Department from
instructor_personal join instructor_Academic on
instructor_personal.inst_code=instructor_Academic.inst_code
join scentific_degree on
scentific_degree.deg_code=instructor_Academic.degr_code join
programs on instructor_Academic.pro_code=programs.pro_code";
        comm.Connection = conn;
        SqlDataReader reader = comm.ExecuteReader();
        GridView1.DataSource = reader;
        GridView1.DataBind();
        conn.Close();
}
```

2.3. Current Student Functions.

2.3.1. Registration after Student login to the system.

```
public partial class registration : System.Web.UI.Page
{
 int usercode=2015003;
  protected void Page_Load(object sender, EventArgs e)
  { }
  protected void Button1_Click(object sender, EventArgs e)
  {
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = 
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
    conn.Open();
    SqlCommand comm = new SqlCommand();
    comm.CommandText = "Select user_name ,password from login 
where user_name='" + TextBox2.Text + "' and  password ='" + 
TextBox1.Text + "'";
    comm.Connection = conn;
    SqlDataReader reader = comm.ExecuteReader();
    if (reader.Read())//for the student have user_name and 
password
    { // usercode = Request.QueryString["regest.aspx"];
      conn.Close();

      //conn.Open();
      //comm.CommandText = " INSERT INTO transaction_login 
(user_name) VALUES ('"+TextBox2.Text+"')";
      //reader = comm.ExecuteReader();
      //conn.Close();

      conn.Open();
      comm.CommandText = "Select user_code  from login where 
user_name='" + TextBox2.Text + "'";
      comm.Connection = conn;
      reader = comm.ExecuteReader();
      if (reader.Read())
      {
       @usercode = Int32.Parse(reader.GetValue(0).ToString());
      //Response.Write("<script>alert(@usercode)</script>");
      }
      else
```

```
            { Response.Write("<script>alert(' No data 
found')</script>"); }
                conn.Close();

            conn.Open();
            comm.CommandText = "select 
student_personal.st_name,programs.pro_name from student_personal  join 
student_Academic on student_personal.st_code=student_Academic.st_code 
join programs on student_Academic.pro_code= programs.pro_code AND 
student_personal.st_code=(Select user_code  from login where 
user_name='" + TextBox2.Text + "')";
            comm.Connection = conn;
           reader = comm.ExecuteReader();
           if (reader.Read())
           {    Label1.Text = reader.GetValue(0).ToString();
                Label2.Text = reader.GetValue(1).ToString();
                conn.Close();

                conn.Open();   //for check if the student exist or 
newcoming
                comm.CommandText = "select course_code from 
Registeration where st_code=(Select user_code  from login where 
user_name='" + TextBox2.Text + "')";
                comm.Connection = conn;
                reader = comm.ExecuteReader();

conn.Close();


//the new student coming


                    conn.Open();
                    comm.CommandText = "select course_code as code 
,course_name  ,course_CH as CH ,total as degree from courses ";
                    comm.Connection = conn;
                   reader = comm.ExecuteReader();
                    GridView1.DataSource = reader;
                    GridView1.DataBind();
                     conn.Close();

                    conn.Open();
                    comm.CommandText = "select course_code as code 
,course_name ,course_CH as CH ,total as degree from elective_courses 
";
                    comm.Connection = conn;
```

```
                     reader = comm.ExecuteReader();
                    GridView2.DataSource = reader;
                    GridView2.DataBind();
                    conn.Close();

}

else
           {
               Response.Write("<script>alert(' No data 
found')</script>");
               conn.Close();
           }

        }
        else
        { Response.Write("<script>alert('Please enter valid Username 
and Password')</script>");
        conn.Close();
        }
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
       string[] stringArray = new string[10];
       // List<string> str = new List<string>();
        SqlConnection myconn = new SqlConnection();
        myconn.ConnectionString = 
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
        SqlCommand comm1 = new SqlCommand();
        SqlDataReader reader;
        //int i=0;
        foreach (GridViewRow item in GridView1.Rows )
        {
            myconn.Open();
            if ((item.Cells[0].FindControl("cbSelect") as 
CheckBox).Checked)
            {
                comm1.CommandText = " INSERT INTO 
Registeration(st_code,course_code,reg_date) VALUES ( @usercode ,'" + 
item.Cells[1].Text + "','2016-2017')";
                comm1.Connection = myconn;
              reader = comm1.ExecuteReader();
            }
            myconn.Close();
       }

        foreach (GridViewRow item in GridView2.Rows)
        {
            myconn.Open();
            if ((item.Cells[0].FindControl("elSelect") as
CheckBox).Checked)
            {
                comm1.CommandText = " INSERT INTO
Registeration2(st_code,elcourse_code,reg_date) VALUES ( @usercode ,'"
+ item.Cells[1].Text + "','2016-2017')";
                comm1.Connection = myconn;
                reader = comm1.ExecuteReader();
            }
            myconn.Close();
        }
        myconn.Open();
        comm1.CommandText = "select Registeration.course_code as code
,courses.course_name ,courses.course_CH as CH
,instructor_personal.inst_name,halls.hall_name,lectures.start_time,lec
tures.end_time,lectures.lect_day from Registeration join courses on
Registeration.course_code=courses.course_code join instructor_course
on courses.course_code=instructor_course.course_code join
instructor_Academic on
instructor_course.inst_code=instructor_Academic.inst_code join
instructor_personal on
instructor_Academic.inst_code=instructor_personal.inst_code join
lectures on courses.course_code=lectures.course_code join halls on
lectures.lect_hall=halls.hall_no AND Registeration.st_code=2015002 ";
       reader = comm1.ExecuteReader();
       GridView3.DataSource = reader;
       GridView3.DataBind();
        myconn.Close();

        myconn.Open();
        comm1.CommandText = "select Registeration2.elcourse_code as
code ,elective_courses.course_name as elective_course
,elective_courses.course_CH as CH
,instructor_personal.inst_name,halls.hall_name,lectures_elective.start
_time,lectures_elective.end_time,lectures_elective.lect_day from
Registeration2 join elective_courses on
Registeration2.elcourse_code=elective_courses.course_code join
instructor_elcourse on
elective_courses.course_code=instructor_elcourse.course_code join
instructor_Academic on
instructor_elcourse.inst_code=instructor_Academic.inst_code join
```

```
instructor_personal on 
instructor_Academic.inst_code=instructor_personal.inst_code join 
lectures_elective on 
elective_courses.course_code=lectures_elective.course_code join halls 
on lectures_elective.lect_hall=halls.hall_no AND 
Registeration2.st_code=2015002 ";
        reader = comm1.ExecuteReader();
        GridView4.DataSource = reader;
        GridView4.DataBind();
        myconn.Close();

    }
}
```

2.3.2. Exams scheduling.

```
protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString = 
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
        conn.Open();
        SqlCommand comm = new SqlCommand();
        //(select st_code from login where user_name='" + 
TextBox2.Text + "')
        comm.CommandText = "select Registeration.course_code as code, 
courses.course_name 
,exam.exam_date,exam.exam_start,exam.exam_end,halls.hall_name  from 
Registeration join courses on 
Registeration.course_code=courses.course_code  join exam on 
courses.course_code = exam.course_code join  exam_location on 
exam.course_code=exam_location.course_code join halls on 
exam_location.hall_no=halls.hall_no AND 
Registeration.st_code=2015002";
        comm.Connection = conn;
       SqlDataReader reader = comm.ExecuteReader();
        GridView1.DataSource = reader;
        GridView1.DataBind();
        conn.Close();
        conn.Open();
        comm.CommandText = "select Registeration2.elcourse_code as 
code, elective_courses.course_name 
,elective_exam.exam_date,elective_exam.exam_start,elective_exam.exam_e
nd,halls.hall_name  from Registeration2 join elective_courses on 
Registeration2.elcourse_code=elective_courses.course_code  join 
elective_exam on elective_courses.course_code =
elective_exam.course_code join  elec_exam_location on
elective_exam.course_code=elec_exam_location.course_code join halls on
elec_exam_location.hall_no=halls.hall_no AND
Registeration2.st_code=(select st_code from login where user_name='" +
TextBox2.Text + "')";
        comm.Connection = conn;
         reader = comm.ExecuteReader();
        GridView2.DataSource = reader;
        GridView2.DataBind();
        conn.Close();


    }
```

2.3.3. Result of exams.

```
protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection();
        conn.ConnectionString =
ConfigurationManager.ConnectionStrings["MyConnectionString"].ToString(
);
        conn.Open();
        SqlCommand comm = new SqlCommand();
        comm.CommandText = "select  courses.course_name
,student_results.final1_degree as degree from student_results join
courses on student_results.course_code=courses.course_code and
student_results.st_code=(select st_code from login where user_name='"+
TextBox2.Text +"')";
        comm.Connection = conn;
        SqlDataReader   reader=comm.ExecuteReader();
        GridView1.DataSource = reader;
        GridView1.DataBind();
        conn.Close();
    }
```

Chapter 7

"Security and Tools"

7.1Security:

7.1.1 Security in four ways.

7.1.1.1 Using security browsers.

7.1.1.2 Stored Procedures (Database Engine).

7.1.1.3 Using Three-Tier Architecture.

7.1.1.4 Using SQL server authentication.

7.2 Tools and programming languages:

7.2.1Tools:

7.2.1.1 Analysis phase

7.2.1.2 Design phase:

7.2.1.3 Implementation phase

7.2.2 Programming languages used on Web development.

7.1 Security

What is Computer Security?

A generic name for the collection of tools designed to protect data and to thwart hackers.

Why is Computer Security Important?

Computer Security allows the University to carry out its mission by:

- ✓ Enabling people to carry out their jobs, education, and research
- ✓ Supporting critical business process.
- ✓ Protecting personal and sensitive information.

Security Goals:

- ✓ **Confidentiality** (Secrecy or Privacy): assets accessed only by authorized parties, Not only reading but viewing, printing or knowing about the asset.

- ✓ **Integrity**: assets modified only by authorized parties
  Includes writing, changing, changing the status, deleting or creating
- ✓ **Availability**: assets are accessible to authorized parties at appropriate times.

7.1.1Security in three ways :(we apply these three ways of security)

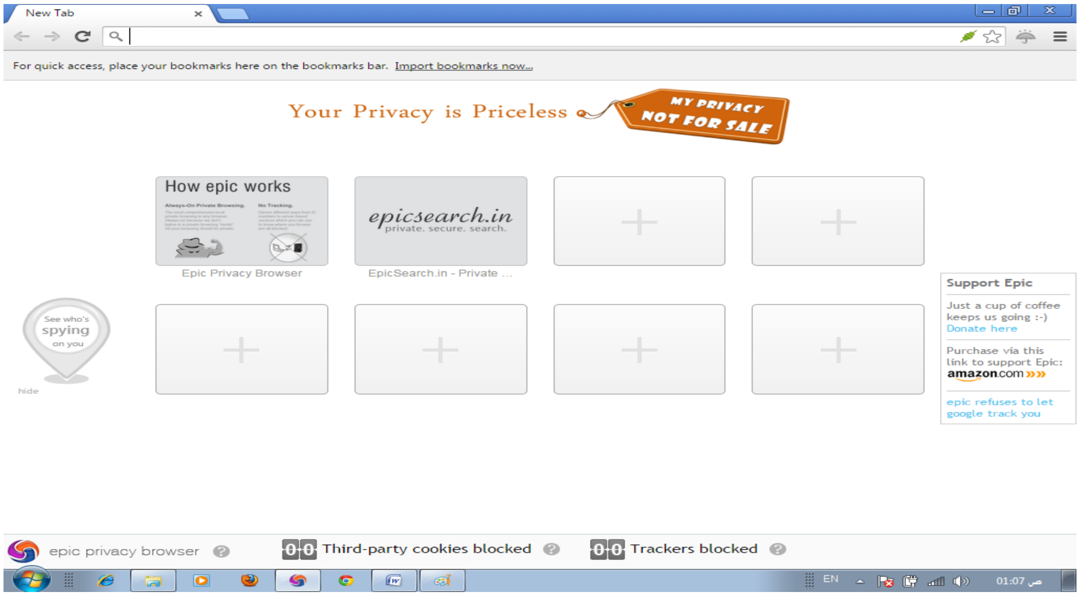
1. Using security browsers
2. Using Three-Tier Architecture**.**
3. Using SQL server authentication.

7.1.1.1Using security browsers

The best 5 secure browsers 2015

- The Best Secure Browser: Epic privacy browser.
- A good Secure Browser: Comodo Dragon/Ice Dragon.
- A good Secure Browser: Tor.
- A good Secure Browser: Dooble.
- A good Secure Browser: Maxthon Cloud Browser.

✓ But We used the best one "Epic"



For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Support Epic

Just a cup of coffee keeps us going :-)
Donate here

Purchase via this link to support Epic: amazon.com »»

epic refuses to let google track you

Figure(7.1)

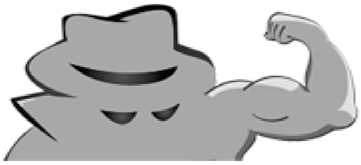
There are plenty of different ways you can ensure security and privacy in your Web browser. You can add a Do Not Track plug-in, or hide your IP address by going through a proxy....

You can simply download the free Epic Privacy Browser and get all the privacy features imaginable, all of them active by default.

How epic work?

Always-On Private Browsing

The most comprehensive local private browsing in any browser.Always-On because we don't believe in a private browsing "mode". All your browsing should be private.

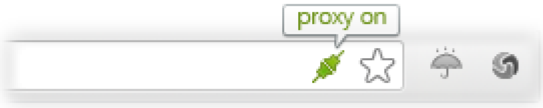


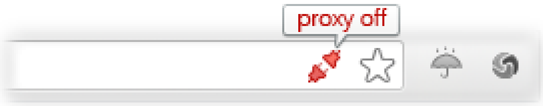
No Tracking

Eleven different ways from ID numbers to server-based services which can be used to know what you browsed

One-Click Encrypted Proxy

Click the red open socket icon to turn on Epic's built-in encrypted proxy. This hides your IP address (with a U.S -based IP) and encrypts all your data when you want to be very private.





Trackers Blocked

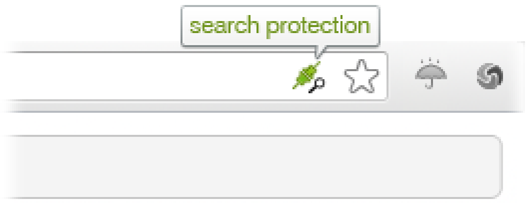
Trackers are companies that track where and what you browse on the internet. They may associate your online data with offline data and sell your data and or use it to show you targeted ads. Epic is the only browser that actively blocks thousands of trackers.

Third-party Cookies Blocked

A cookie is a small piece of data sent from a website and stored in your browser. Cookies from companies other than the website you're visiting (third-party cookies) are almost always used to track your browsing.

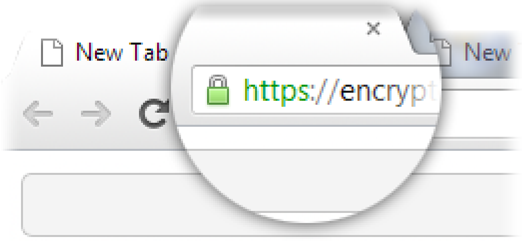
Search Protection

Epic protects your searches from being saved and tracked. Epic automatically loads search engines vis its built-in proxy because search engines track you by your IP address.



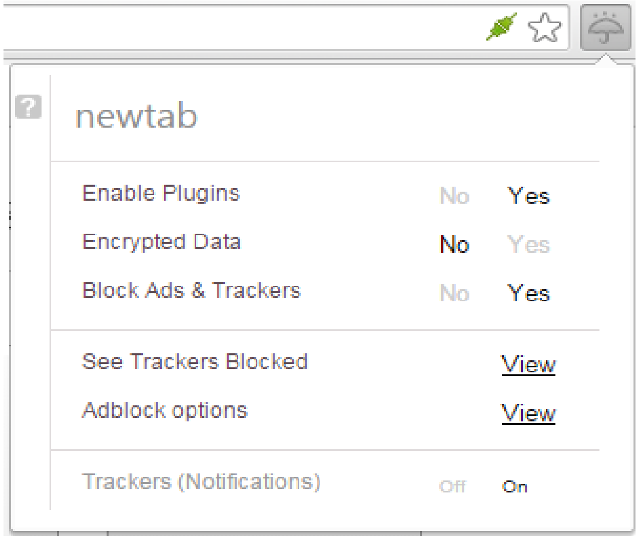
Encrypted Data Preference

If available, Epic will use an https version of the website(s) you visit so that data sent and received is encrypted.



Unified Button

In case a site you visit doesn't work, you may need to adjust Epic's settings for that particular website. Click on the umbrella icon and try turning off one or more of Epic's settings for that site to make the site load correctly.



newtab

Enable Plugins No [✓ Yes]

Encrypted Data [✓ No] Yes

Block Ads & Trackers No [✓ Yes]

See Trackers Blocked View

Adblock options View

Trackers (Notifications) Off [✓ On]

Because of previous features we used epic browser to protect the website and increase security.

7.1.1.2 Using Three-Tier Architecture:

A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture.

The three tiers in a three-tier architecture are:

- ✓ Presentation Tier: Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

- ✓ Application Tier: Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

- ✓ Data Tier: Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

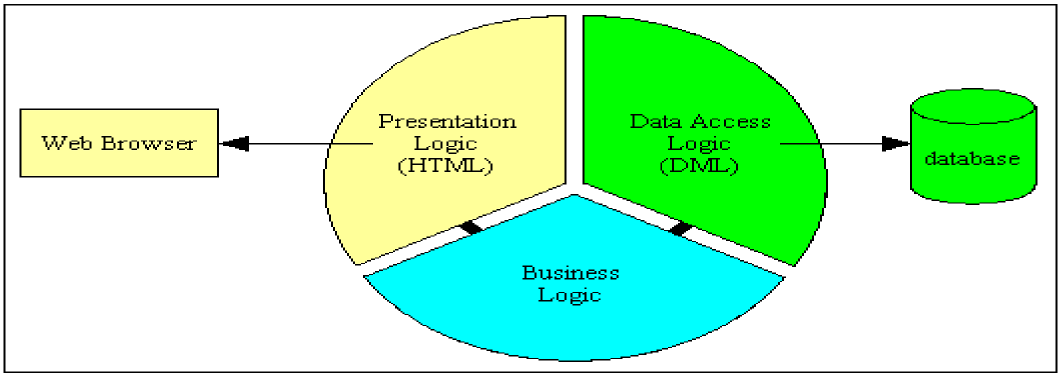


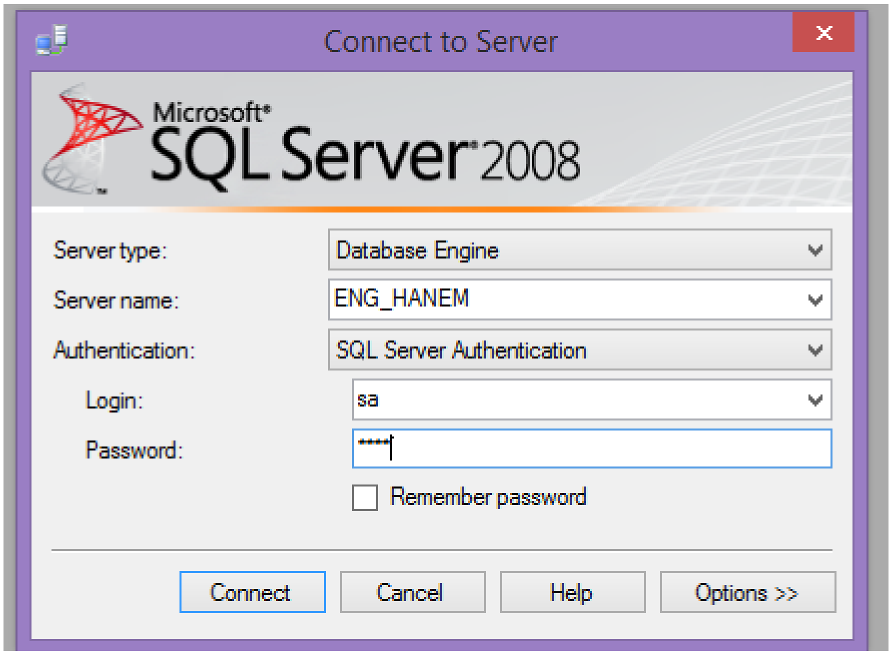
Figure (7.2)

Note here that the presentation layer has ***no direct communication*** with the data access layer - it can only talk to the business layer.

Three Tier Advantages:

- **Improved Security**
  - ✓ Since the client doesn't have direct access to the database, Data layer is more secure.
- Business Logic is generally more secure since it is placed on a secured central server.
- **Reduced Distribution**
  - ✓ Changes to business logic only need to be updated on application servers and need not to be distributed on clients.

7.1.1.3 Using SQL server authentication



Server type: Database Engine

Server name: ENG_HANEM

Authentication: SQL Server Authentication

Login: sa

Password: ****

□ Remember password

Figure (7.3)

Using SQL server authentication as shown on previous figure enables us to use user name and password to enter to system's data base only by enter right user name and password .this is helpful that prevent hackers to enter to data base because they don't know the user name and password.

Also on our data base we identify specific users and declare to them unique users name and password to access specific functions such as:

- Enabling only students that have right user name and password to access specific functions such as login and (view result, login, register).
- Enabling only specific staff that have right user name and password to access specific functions such as (view documents).
- Enabling only specific administrators that have right user name and password to access specific functions such as (update, delete, insert).

7.2 Tools and programming languages:

We used different tools and programming languages on different phases.

7.2.1 Tools:

- #### 7.2.1.1 Analysis phase

  ✓ **Edraw max 7:**

  Is a 2D business technical diagramming software which helps on creating many charts, Edraw Max enables students, teachers and business professionals to reliably create and publish various kinds of diagrams to represent any idea.

  Edraw enables us to draw use case diagrams, sequence diagrams, flow chart diagrams of our project.

7.2.1.2 Design phase:

Edraw max 7 enables us to draw ERD (Entity Relationship Diagram).

7.2.1.3 Implementation phase

- ✓ Microsoft SQL server 2008:

Is a relational database management system developed by Microsoft .As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft SQL server 2008 enables creating system data base and its tables.

- ✓ Microsoft Visual Studio 2013:

Is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites and web applications

Visual Studio enables us to:

- ✓ Create ASP application to create design and interface of our project
- ✓ Connecting between data base and system design.

7.2.2 Programming languages used on Web development:

C# programming language

Is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.

ASP.NET :

Is an open source[server-side Web application framework designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

Chapter 8

“Test Cases“

8.1 Overview.

8.2 Test case 1: login process.

8.3 Test case 2: Registration process.

8.4 Test case 3: View Exam table.

8.5 Test case 4: View result.

8.6 Test case 5: View Academic course schedule.

8.1 Overview

In the previous chapters, the design and development processes of the proposed system are mentioned in details. However, this chapter provides a different look to the proposed system as the implementation phase of it. This chapter will provide the proposed system as a real running system not only conceptual model of it. This is done by providing test cases to it with input and output screenshots to it.

Interface

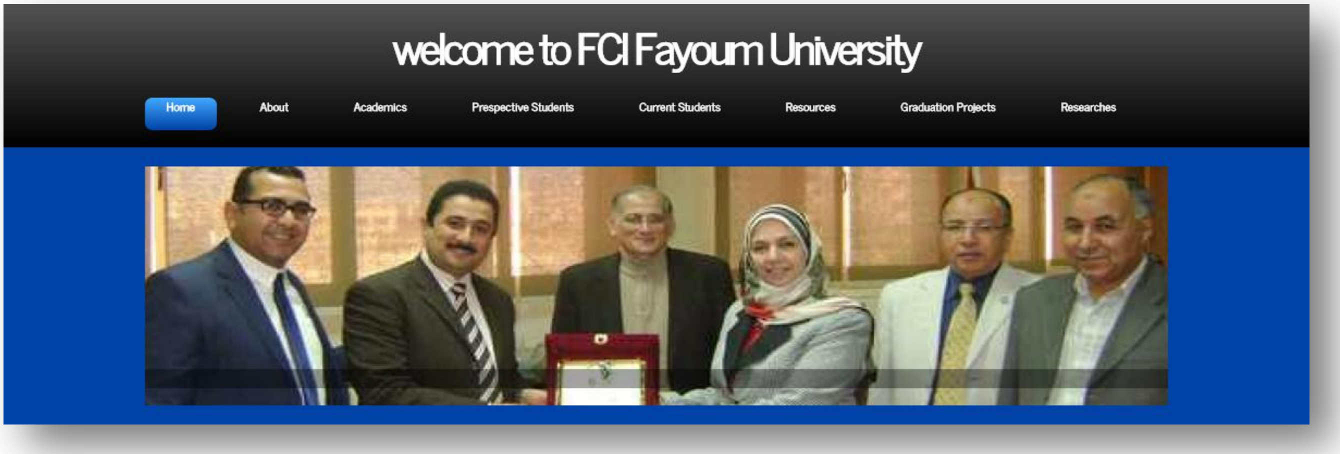


Figure (8.1)

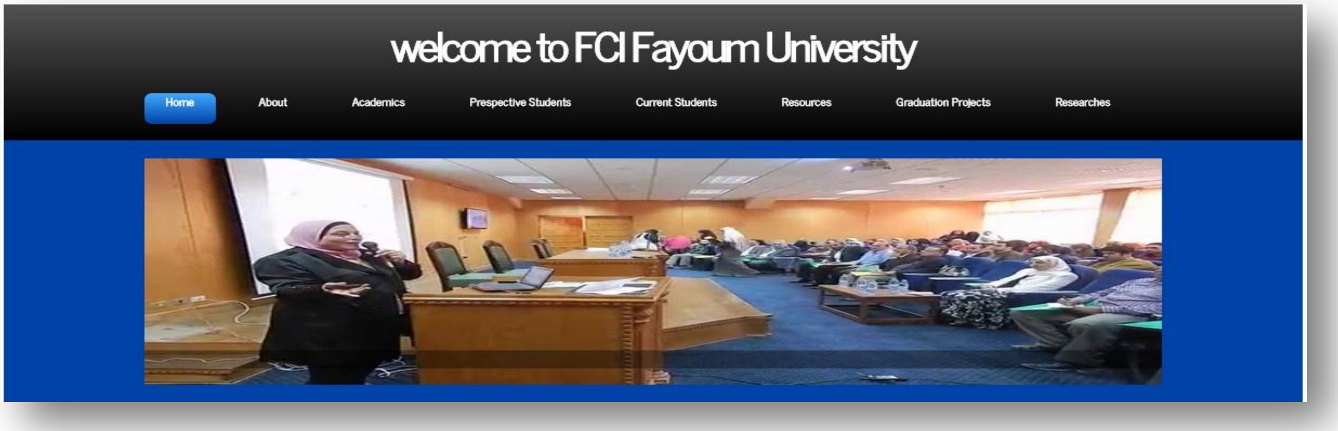
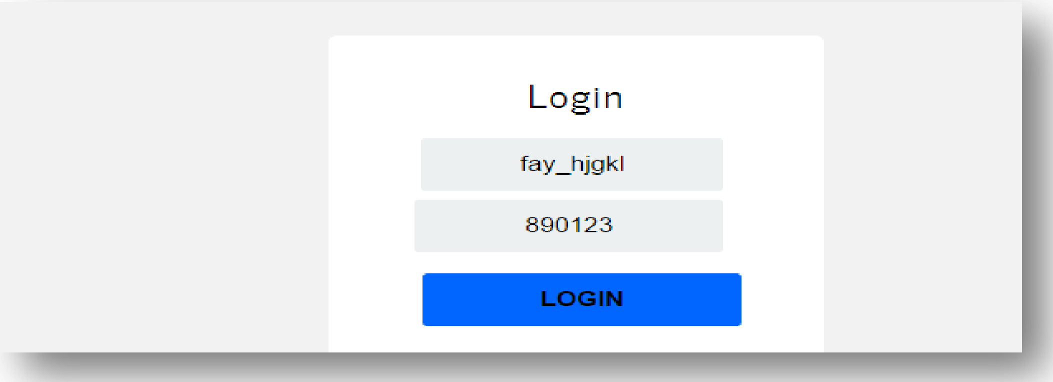


Figure (8.2)

The system must be tested in order to detect any error or mistake after the system was designed, in order to develop it. There are some of testing levels that have been executed which are unit testing, integration testing and user acceptance test.

8.2 Test case 1: login process

- ***Test Precondition:*** the student has accessed the internet.
- ***Input:*** Entering his password and user name.



Login

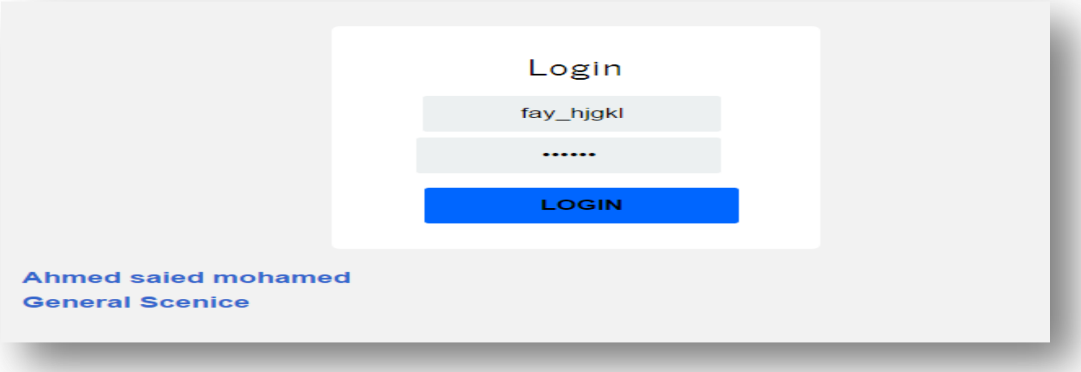
fay_hjgkl

890123

Figure (8.3)

- ***Output:*** Two probabilities.

1- If the password is true: This will appear.



Login

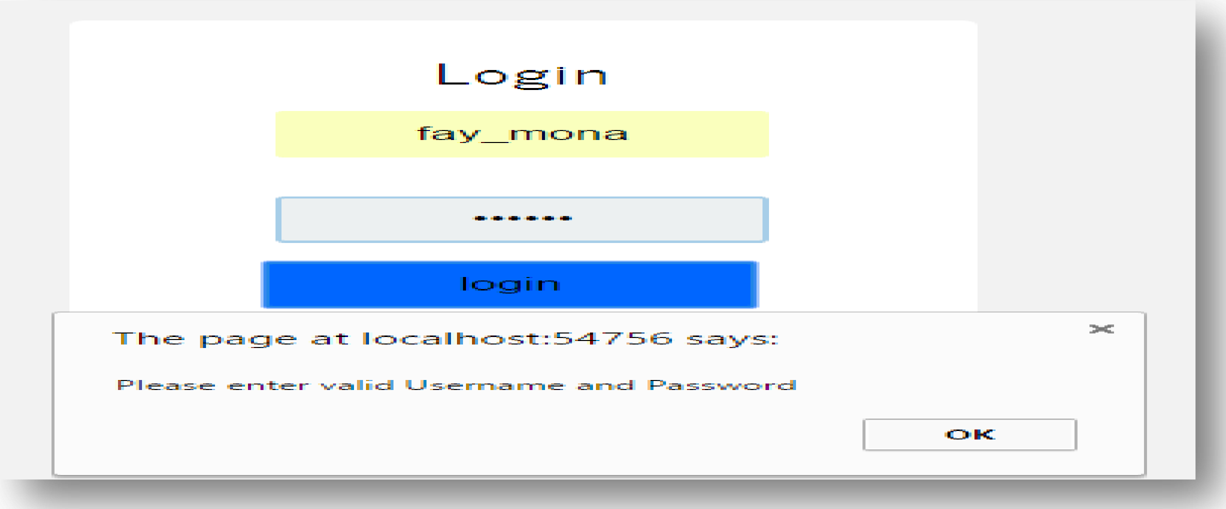
fay_hjgkl

••••••

Ahmed saied mohamed
General Scenice

Figure (8.4)

2- If the password or user name is false: error message will appear:



The page at localhost:54756 says:

Please enter valid Username and Password

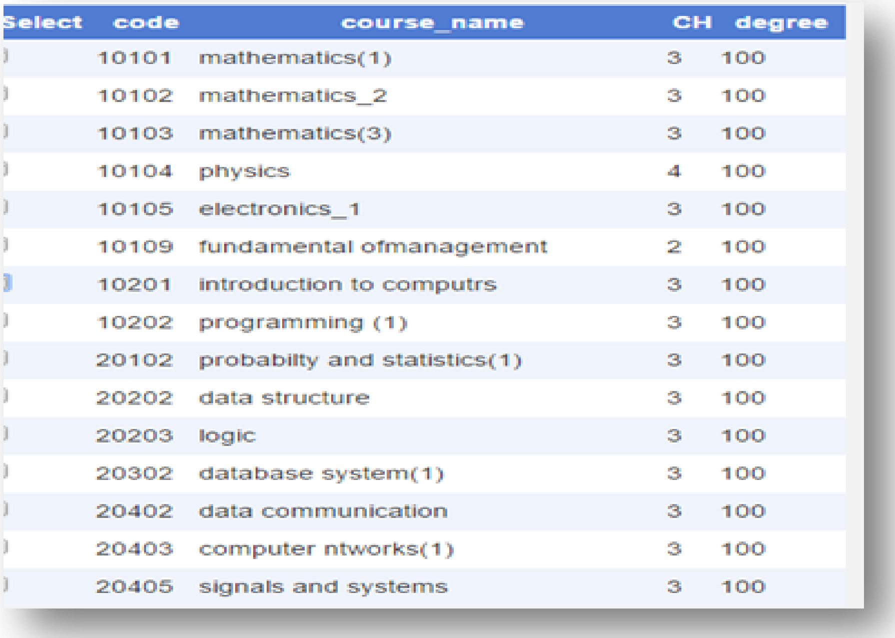
Figure (8.5)

8.3 Test case 2: Registration process

- ***Test Precondition:*** Entering password and user name in a correctly way at the registration page.
  - *After the system check his user name and password, system knows that:*
    - If he a new comer student : -

      System will make a grid view of all courses 'mandatory & elective' in our system.

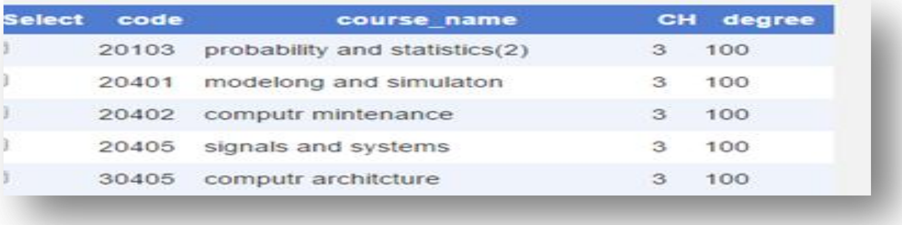
Mandatory courses:



Select	code	course_name	CH	degree
	10101	mathematics(1)	3	100
	10102	mathematics_2	3	100
	10103	mathematics(3)	3	100
	10104	physics	4	100
	10105	electronics_1	3	100
	10109	fundamental ofmanagement	2	100
	10201	introduction to computrs	3	100
	10202	programming (1)	3	100
	20102	probabilty and statistics(1)	3	100
	20202	data structure	3	100
	20203	logic	3	100
	20302	database system(1)	3	100
	20402	data communication	3	100
	20403	computer ntworks(1)	3	100
	20405	signals and systems	3	100

Figure (8.6)

Elective courses:



Select	code	course_name	CH	degree
	20103	probability and statistics(2)	3	100
	20401	modelong and simulaton	3	100
	20402	computr mintenance	3	100
	20405	signals and systems	3	100
	30405	computr architcture	3	100

figure (8.7)

- If he is a current student :

System will checks the GPA belongs to him, then the system will make a grid view of his allowed courses according to that:

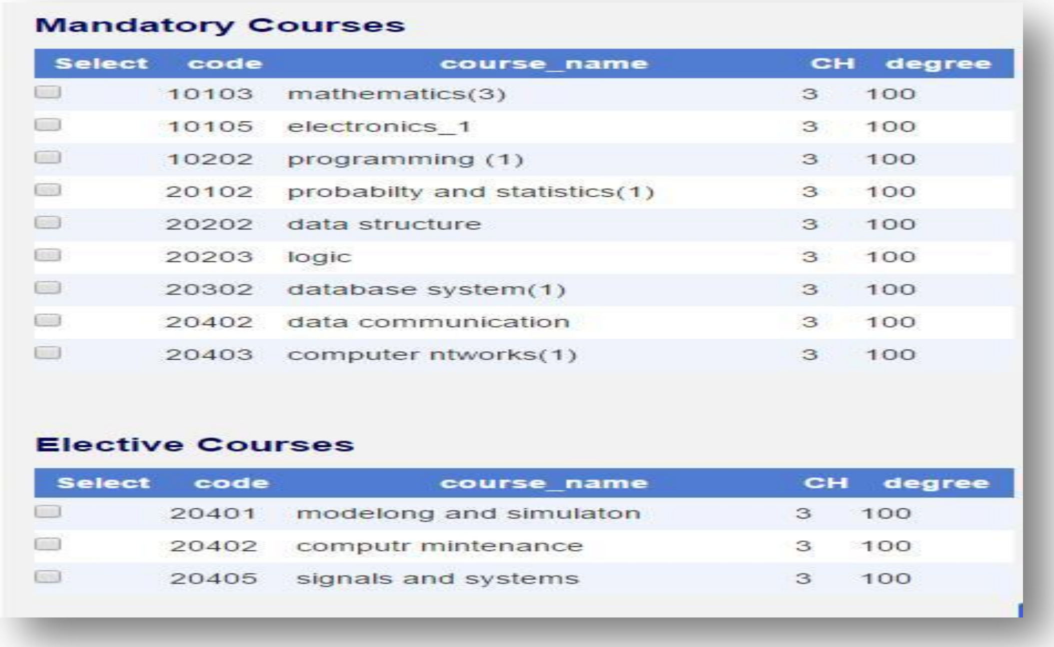
- If his GPA is less than (1.4): student is considered fail and he must re-enter this semester.

- If his GPA is more than (1.4): system will check the degrees of all courses he was tested in.

  ✓ If there are courses whose degrees are less than (1.4):

  System will pass these courses to the current semester,

  ✓ But if all degrees are more than (1.4) system will show a list of the only courses which he did not study before.



Mandatory Courses

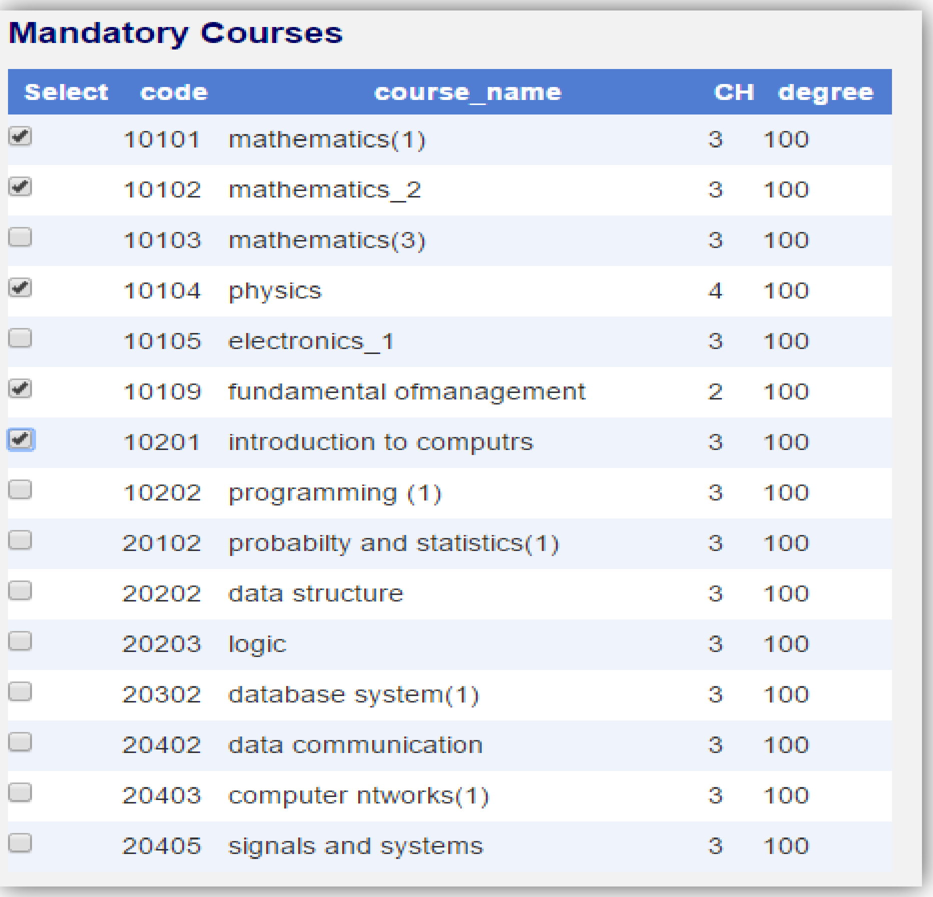
Select	code	course_name	CH	degree
□	10103	mathematics(3)	3	100
□	10105	electronics_1	3	100
□	10202	programming (1)	3	100
□	20102	probabilty and statistics(1)	3	100
□	20202	data structure	3	100
□	20203	logic	3	100
□	20302	database system(1)	3	100
□	20402	data communication	3	100
□	20403	computer ntworks(1)	3	100

Elective Courses

Select	code	course_name	CH	degree
□	20401	modelong and simulaton	3	100
□	20402	computr mintenance	3	100
□	20405	signals and systems	3	100

Figure(8.8)

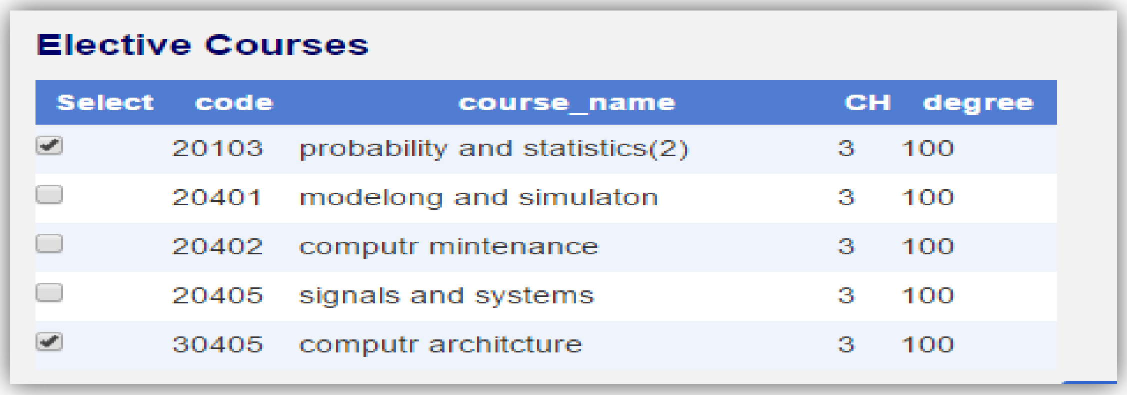
Input: Selecting of courses which he wants.



Mandatory Courses

Select	code	course_name	CH	degree
☑	10101	mathematics(1)	3	100
☑	10102	mathematics_2	3	100
□	10103	mathematics(3)	3	100
☑	10104	physics	4	100
□	10105	electronics_1	3	100
☑	10109	fundamental ofmanagement	2	100
☑	10201	introduction to computrs	3	100
□	10202	programming (1)	3	100
□	20102	probabilty and statistics(1)	3	100
□	20202	data structure	3	100
□	20203	logic	3	100
□	20302	database system(1)	3	100
□	20402	data communication	3	100
□	20403	computer ntworks(1)	3	100
□	20405	signals and systems	3	100

Figure(8.9)



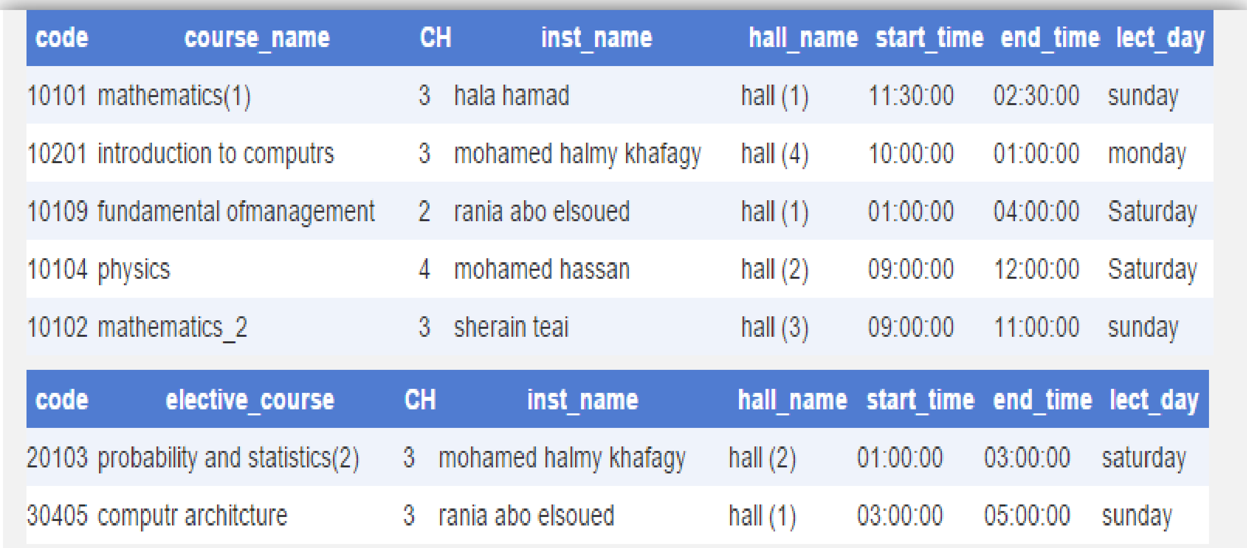
Elective Courses

Select	code	course_name	CH	degree
☑	20103	probability and statistics(2)	3	100
□	20401	modelong and simulaton	3	100
□	20402	computr mintenance	3	100
□	20405	signals and systems	3	100
☑	30405	computr architcture	3	100

Figure(8.10)

Output: Two probabilities:

- If the student selects the courses in a correctly way:
  - ✓ Time Table of selected courses will appear :



code	course_name	CH	inst_name	hall_name	start_time	end_time	lect_day
10101	mathematics(1)	3	hala hamad	hall (1)	11:30:00	02:30:00	sunday
10201	introduction to computrs	3	mohamed halmy khafagy	hall (4)	10:00:00	01:00:00	monday
10109	fundamental ofmanagement	2	rania abo elsoued	hall (1)	01:00:00	04:00:00	Saturday
10104	physics	4	mohamed hassan	hall (2)	09:00:00	12:00:00	Saturday
10102	mathematics_2	3	sherain teai	hall (3)	09:00:00	11:00:00	sunday

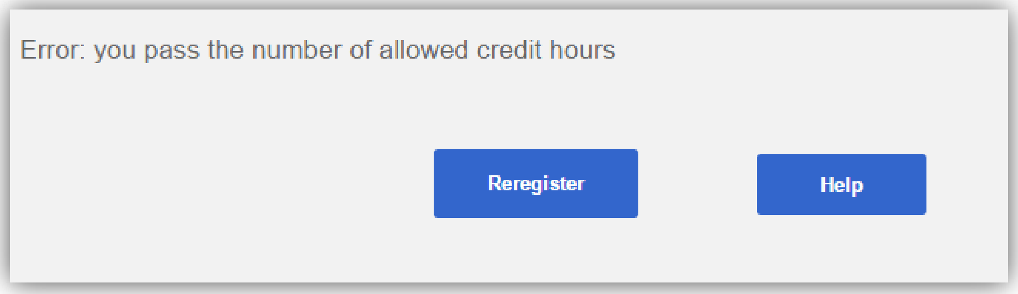
code	elective_course	CH	inst_name	hall_name	start_time	end_time	lect_day
20103	probability and statistics(2)	3	mohamed halmy khafagy	hall (2)	01:00:00	03:00:00	saturday
30405	computr architcture	3	rania abo elsoued	hall (1)	03:00:00	05:00:00	sunday

Figure(8.11)

- If the student selects the courses in a wrong way:

The wrong way in our system is divided into two cases:

1. **Non allowed number of credit hours Error message:**
   - If he selects a number of Courses which their credit hours are larger than his allowed number, Thus an Error message will appear.

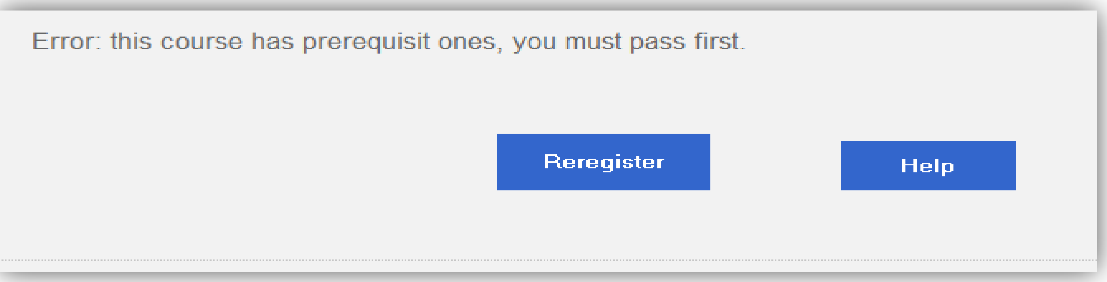


Error: you pass the number of allowed credit hours

Figure(8.12)

2. A prerequisite Error message:

- If the student selects course which has a prerequisite one, this message will appear that he cannot study this course until he passes the prerequisite one.

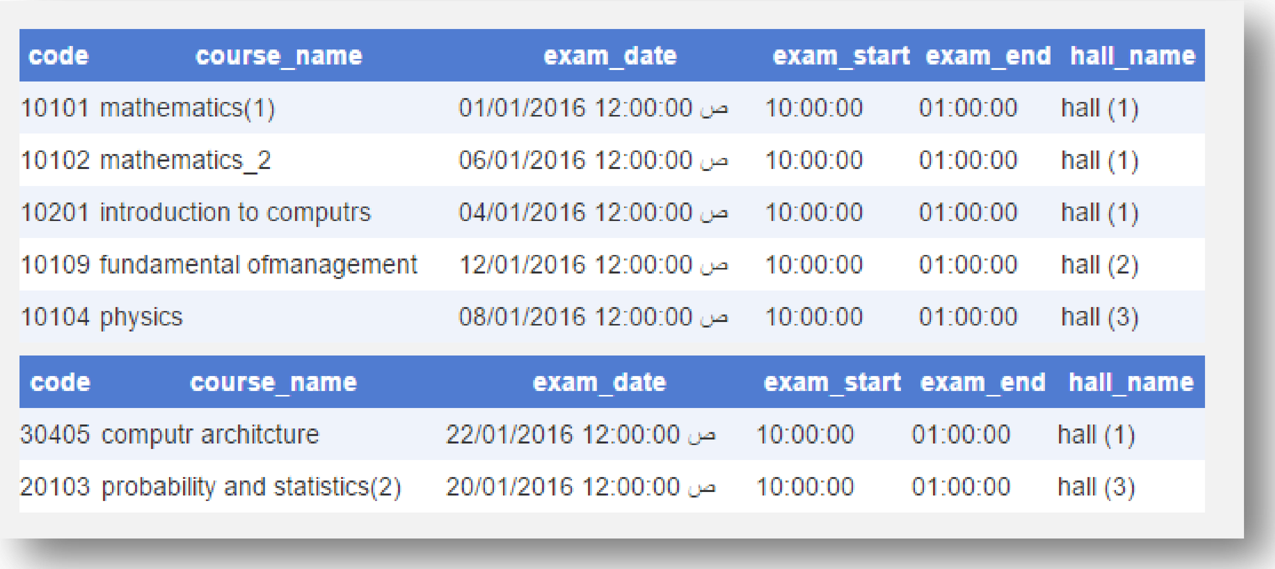


Error: this course has prerequisit ones, you must pass first.

Figure (8.13)

8.4 Test case 3: View Exam table:

- ***Test Precondition:*** the student has accessed the internet.
- ***Input:*** Entering password and user name in a correctly way in (Exams scheduling) page.
- **Output:** This table will appear, according to his selected courses.



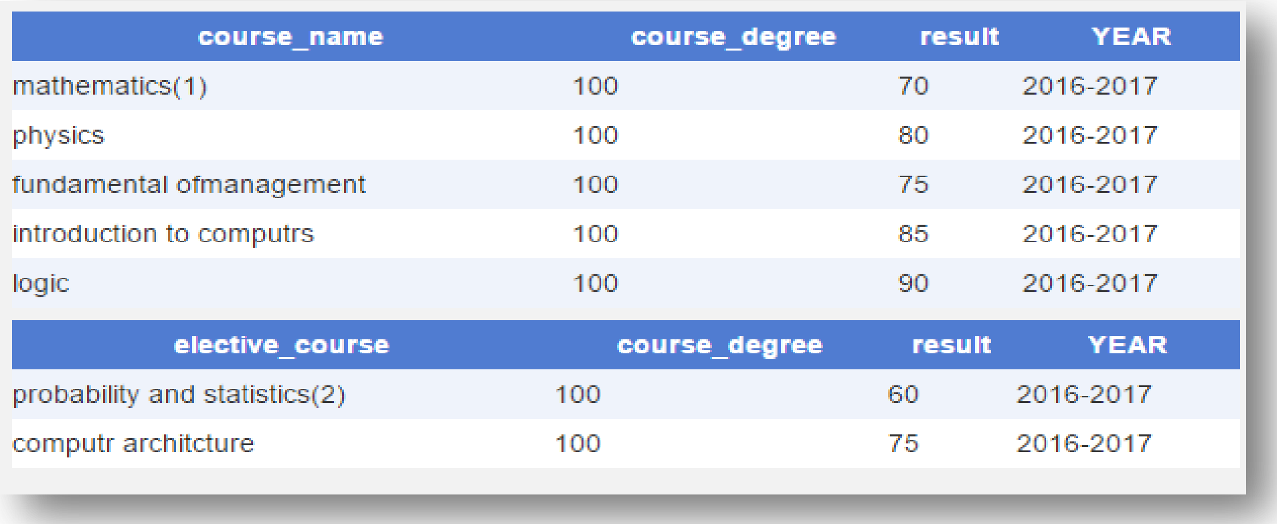
code	course_name	exam_date	exam_start	exam_end	hall_name
10101	mathematics(1)	01/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (1)
10102	mathematics_2	06/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (1)
10201	introduction to computrs	04/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (1)
10109	fundamental ofmanagement	12/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (2)
10104	physics	08/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (3)

code	course_name	exam_date	exam_start	exam_end	hall_name
30405	computr architcture	22/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (1)
20103	probability and statistics(2)	20/01/2016 12:00:00 ص	10:00:00	01:00:00	hall (3)

Figure (8.14)

8.5 Test case 4: View Result:

- ***Test Precondition:*** the student has accessed the internet.
- ***Input:*** Entering password and user name in a correctly way in (Results of Exams) page.
- ***Output:*** This table will appear, this table will be filled from the database according to his courses he choice on the current semester, and he attend their exams.



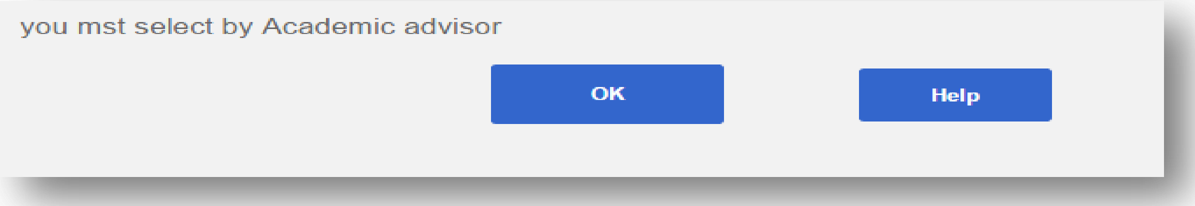
course_name	course_degree	result	YEAR
mathematics(1)	100	70	2016-2017
physics	100	80	2016-2017
fundamental ofmanagement	100	75	2016-2017
introduction to computrs	100	85	2016-2017
logic	100	90	2016-2017

elective_course	course_degree	result	YEAR
probability and statistics(2)	100	60	2016-2017
computr architcture	100	75	2016-2017

Figure (8.15)

Cases of output of test case 4 ‘view result’:

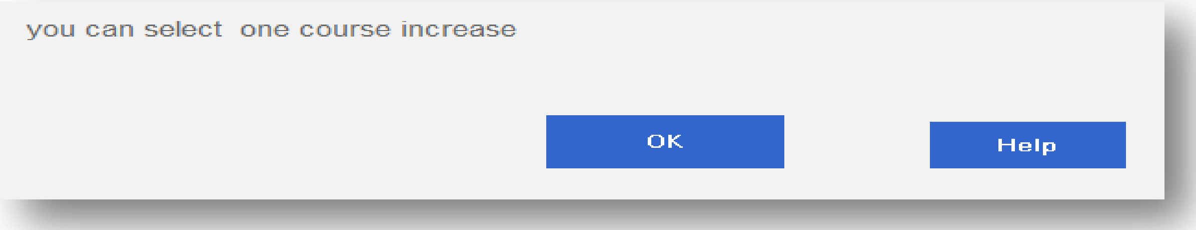
1-If the student is so weak (means that student has so weak GPA), student must register under the care of his academic guide, this will appear:



you mst select by Academic advisor

Figure (8.16)

2-If student's GPA for courses is more than 3.4 (Excellent), system will give student three credit hours bonus (over) on elective courses, this will appear:

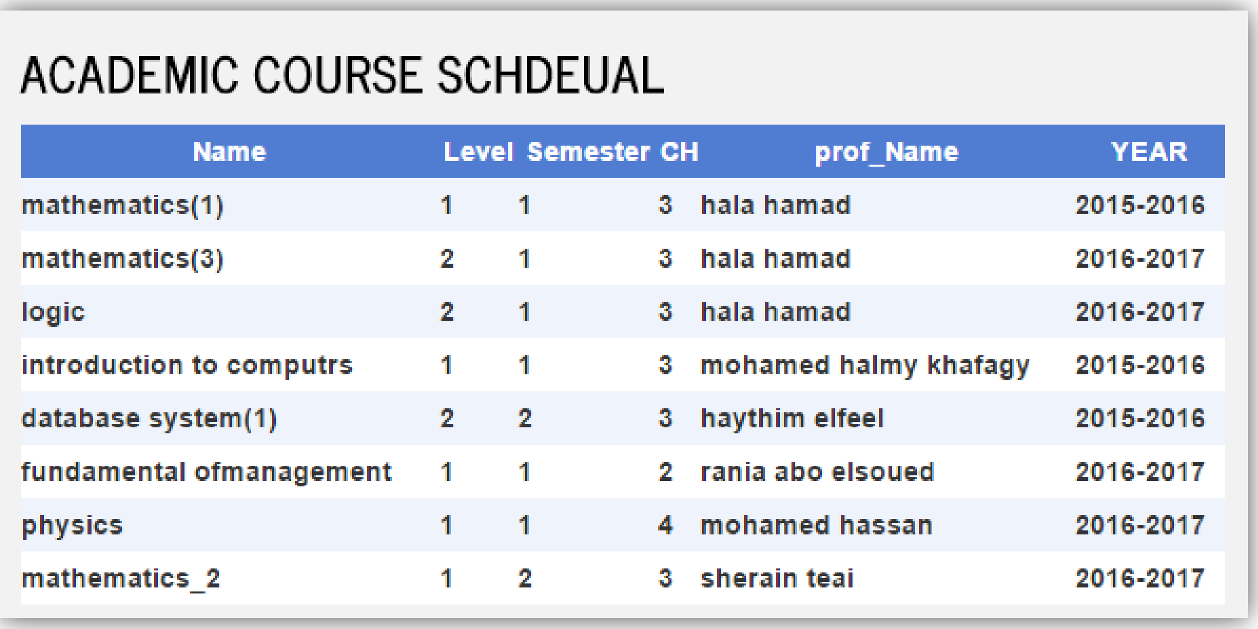


you can select one course increase

Figure (8.17)

8.6 Test case 5: Academic course schedule:

- *Test Precondition:* the student and staff have accessed the internet.
- *Input:* Entering password and user name in a correctly way in (courses schedule) page.
- *Output:* This table will appear.



ACADEMIC COURSE SCHDEUAL

Name	Level	Semester	CH	prof_Name	YEAR
mathematics(1)	1	1	3	hala hamad	2015-2016
mathematics(3)	2	1	3	hala hamad	2016-2017
logic	2	1	3	hala hamad	2016-2017
introduction to computrs	1	1	3	mohamed halmy khafagy	2015-2016
database system(1)	2	2	3	haythim elfeel	2015-2016
fundamental ofmanagement	1	1	2	rania abo elsoued	2016-2017
physics	1	1	4	mohamed hassan	2016-2017
mathematics_2	1	2	3	sherain teai	2016-2017

Figure (8.18)

Chapter 9

“Summary and future work”

9.1 Summary

9.2 Future work

9.1 Summary

Phase1 about FCI EL-fayoum University:

- Obliging the student to eliminate the Tuition Duration in a specific period was making many problems. These problems are sometimes causing stress for some students, and causing space of time for other students according to their capabilities in terms of time and abilities of the study

- Through the website the student knows what it duties and its rights, also it Provides many functions that are necessary for students and staff. It provides news up to date such as activities, studying and exams tables, champions, prizes, events, results.

- Freedom for the student to choice a set of courses with specific credit hours that he could study, by choosing compulsory and elective courses.

- Elective courses which do not have prerequisite courses will be allowed to the student otherwise, if one has a prerequisite course, thus this elective one will not be allowed to him until he pass this prerequisite.

- Activation of a specific course, it must have at least twenty students register in this one.

- **Bachelor degrees in Computers and Information in the following specialization:**

A. Department of Computer Science.
B. Department of Information Systems.
C. Information technology

- **Semester consists of seventeen weeks distributed as follows:**

A. Registration period is one week.
B. The study period is fourteen weeks.
C. The exams period is two weeks.

9.2 Future work

We intend to improve our (phase 1) web site to include additional functions for the staff and students such as:

- Students can pay the fees online.
- Set up Wi-Fi café (to access internet inside the university and every student has an account on the website, thus they could access the Wi-Fi.)
- Instructors have the ability to upload the lectures on the website.
- Students can download lectures from the website.
- Employees have the ability of entering the web site (print a paper include all student's degree in specific course in a computers science department).
- Designing (phase 2) Online Credit hours registration system for EL-Fayoum University (for all Faculties) with all previous improvements.

Appendix A

1. List of figures.

2. List of tables.

3. List of equations.

1. LIST OF FIGURES

Figure (1.1) System overview……………………………………………12

Figure (2.1) GPA example……………………………………………….27

Figure (3.1) SWOT analysis………………………………………………...33

Figure (3.2) requirement specification…….…………………………….38

Figure (4.1) SDLC…………………………………………………………...44

Figure (4.2) system's active actors…………………………...………55

Figure (4.3) Administrator's use case diagram……………………..……….59

Figure (4.4) Select, course use case diagram………………..……………....64

Figure (4.5) select department use case diagram……………………………65

Figure (4.6) registration use case diagram ………………..…………………68

Figure (4.7) view result use case diagram ……………..……………………72

Figure (4.8) login sequence diagram ……………………………………...75

Figure (4.9) select course sequence diagram………………………………...75

Figure (4.10) college level plan sequence diagram …………………………76

Figure (4.11) registration sequence diagram ………………………………..77

Figure (4.12) view result sequence diagram…………………………………78

Figure (4.13) insert sequence diagram ………………………………………79

Figure (4.14) delete sequence diagram ……………………….……………..80

Figure (4.15) update sequence diagram …..…………………………………81

Figure (4.16) login flow chart diagram ……………………………………..82

Figure (4.17) select course time table flow chart diagram…………………..83

Figure (4.18) view college plan level flow chart diagram ……………….….84

Figure (4.19) registration flowchart diagram ……………………….………85

Figure (4.20) view result flow chart diagram ……………………..…………86

Figure (4.21) insert flow chart diagram ………………………………………..87

Figure (4.22) delete flow chart diagram ………………..……………………88

Figure (4.23) update flow chart diagram ………………………………………89

Figure (5.1) ERD diagram …………..…………………………………………100

Figure(6.1) create data base table………………………………………………113

Figure(6.2)create person_type table……………………………………………115

Figure(6.3)create personal_code table………………………………………… 115

Figure(6.4)create login table……………………………………………………116

Figure(6.5)create transaction_login table………………………………………117

Figure(6.6)create student_personal table……………………………….……118

Figure(6.7)create programs table……………………………………………..119

Figure(6.8) create student_Academic table…………………………………..120

Figure(6.9) create course_type table……………………………………..…. 121

Figure(6.10) create halls table………………………………………………. 122

Figure(6.11) create courses table…………………………………………… 123

Figure(6.12) create pre_courses table……………………………………… 124

Figure(6.13) create elective_coursestable…………………………..……. 126

Figure(6.14) create Registeration table………………………………………127

Figure(6.15) create instructor_personaltable……………………………… 128

Figure(6.16) create scientific_degreetable……………………………………129

Figure(6.17) create instructor_Academictable…………………………………130

Figure(6.18) create instructor_course table…………………………….……131

Figure(6.19) create exam table……………………..132

Figure(6.20) create exam_locationtable………………………………………..133

Figure(6.21) create student_resultstable……………………………………….134

Figure(6.22) create lectures table………………………………………………135

Figure(6.23) create local_labtable…………………………………….............136

Figure(6.24) create sections table……………………………………………….137

Figure(6.25) create labs table…………………………………………….…….138

Figure(6.26) create section_instructortable…………………………..…….. 139

Figure(6.27) create lab_instructor table……………………………………… 140

Figure(6.28) creating ASP project web site……………………………………….. 141

Figure(6.29) creating master page……………………………………………………   142

Figure(6.30) : creating web form page inherit from master page……………….. 143

Figure (7.1) Epic security browser ……….……………………….………………… 156

Figure (7.2) 3TIER architecture ………………………………………….….…160

Figure (7.3) SQL server authentication…………………………………...…161

Figure (8.1) System interface………………………………………………… 165

Figure (8.2) Another system interface……………………………………… ...165

Figure (8.3)Login process figure ……………………………………………166

Figure (8.4) Output of truly login process……………………………………...166

Figure (8.5) Output of wrong login process……………………………………167

Figure (8.6)List of Mandatory courses for a new comer student………………168

Figure (8.7) List of Elective courses for a new comer student……………….168

Figure (8.8) List of Mandatory & Elective a current courses for student……169

Figure (8.9) Input for Registration case of Mandatory courses……………….170

Figure (8.10) Input for Registration case of Elective courses…………………170

Figure (8.11) Time Table for a student………………………………………...171

Figure (8.12) Non allowed number of credit hours Error message. …………...171

Figure (8.13) A prerequisite Error message……………………………………172

Figure (8.14) Exam table……………………………………………………….172

Figure (8.15) View result………………………………………………………173

Figure (8.16) Case one of output of test case 4 ‘view result’………………….173

Figure (8.17) Case two of output of test case 4 ‘view result’…………………..174

Figure (8.18) Academic course schedule……………………………………….174

2 .LIST OF TABLES

Table (2.1) course coding ……………………………………………………….18

Table (2.2) college requirements mandatory courses………………………….21

Table (2.3) college requirements optional courses…………………...…….22

Table (2.4) Department requirements mandatory courses……………..………23

Table (2.5) Department requirements optional courses………………………..25

Table (2.6) Numeric and Symbolic connotations of degree and estimates……26

Table (4.1) Student log in analysis table……………………………….. …….56

Table (4.2) Employees log in analysis table…………………………………..58

Table (4.3) Administrator's update analysis table ……………………………60

Table (4.4) Administrator's insert analysis table ………………….……………62

Table (4.5) Administrator's delete analysis table …………….…..………...63

Table (4.6) select course name analysis table …………………………………65

Table (4.7) select department analysis table …………………………………..66

Table (4.8) GPA calculation analysis table ……………………..……………..67

Table (4.9) registration analysis table ……………………………………………71

Table (4.10) view result analysis table ……………………………………………73

3. List of EQUATIONS

Equation (4.1) GPA calculation ……………………………………………………67

Appendix B

1. ABBREVIATIONS.
2. REFERANCES.

1 . ABBREVIATIONS

CS	Computer Science.
IS	Information System.
IT	Information technology.
DSDM	Dynamic Systems Development Model
FCI	Faculty of Computer and Information.
TA	Teaching Assistant.
US	United States.
CH	Credit Hours.
BS	Bachelor of Science.
GPA	Grade Point Average.
QAU	Quality Assurance Units.
DSASU	Development of Students' Assessment Systems Unit.
SWOT	Strengths, weaknesses, opportunities and threats.
SRS	Software Requirements Specification.

URD	User Requirement(s) Document .
SDLC	Systems Development Life Cycle.
RAD	Rapid Application Development.
JAD	Joint Application Development.
GUI	Graphical User Interface.
ERD	Entity Relationship Diagrams.
PK	Primary key.
FK	Foreign key.
SQL	Structured Query Language.
ASP	Application Service Provides.
IP	Internet Protocol.
DML	Data manipulation language.
HTML	Hypertext Markup Language.

2. *REFERANCES*

[1] Misr university website.

http://www.must.edu.eg.

[2] Faculty of computers and information website of Cairo university.

. http://www.fci.cu.edu.eg/ar/

[3] Nahda university website.

http://www.nub.edu.eg/

[4] College of information and technology of misr university book

[2008-2009].

[5] The Undergraduate schedule of Computers and Information Faculty of Fayoum University 2015.