

# ADVANCED SOFTWARE ENGINEERING COURSE – FALL 2015

## POST GRADUATE STUDIES AUTOMATION SYSTEM DOCUMENTATION

January 23, 2016

Course Supervised by:

Prof. Dr. Mohamed S. Abougabal

Presented by:

Ahmed Badie

Arsany Guirgus

Islam Moursy

Mohamed Madian

Samar Abo Zeed



## TABLE OF CONTENTS

1	Statement of work .....	10
1.1	Introduction/Background.....	10
1.2	Scope of Work .....	10
1.3	Period of Performance .....	10
1.4	Place of Performance .....	10
1.5	Work Requirements .....	10
2	Minutes of Meetings.....	12
2.1	Meeting Objectives .....	12
2.2	Action Item Review .....	12
2.3	Schedule Review .....	12
2.4	Planned Work for the Next Two Weeks.....	12
2.5	New Action Items .....	13
3	Project Management Plan .....	14
3.1	Preliminary project plan.....	14
4	Risk Management Plan .....	17
4.1	Introduction.....	17
4.2	Top Three Risks .....	18
4.3	Risk Management Approach .....	18
4.4	Risk Identification.....	18
4.5	Risk Qualification and Prioritization.....	19
4.6	Risk Monitoring .....	19
4.7	Risk Mitigation and Avoidance.....	19
4.8	Risk Register .....	20
5	Detailed Design .....	21
5.1	Introduction.....	21
5.1.1	Purpose .....	21
5.1.2	Scope.....	21
5.1.3	Overview .....	21




---

5.2	System Overview .....	21
5.3	System Architecture .....	22
5.3.1	Architectural Design.....	22
5.3.2	Decomposition Description .....	23
5.3.3	Design Rationale .....	24
5.4	Data Dictionary.....	40
5.5	Human Interface Design.....	43
5.5.1	Overview of User Interface .....	43
5.5.2	Screen Images .....	43
6	Requirements Elicitation.....	55
7	Requirements Traceability Matrix .....	57
7.1	Requirements Traceability Matrix (RTM).....	57
8	Process Activity Task Matrix .....	60
8.1	Tasks .....	60
8.2	Activities .....	61
8.3	Processes.....	61
8.4	PATM in tabular format .....	61
9	Quality Assurance Plan .....	64
9.1	Introduction.....	64
9.1.1	Purpose and Scope.....	64
9.1.2	References .....	64
9.1.3	Assumptions and Limitations.....	64
9.2	QA Auditing Targets and Standards .....	64
9.2.1	Processes to be audited .....	64
9.2.2	Work Products to be audited.....	65
9.2.3	Project Checkpoints (exit criteria) .....	65
9.2.4	Audit Standards.....	66
9.3	Management and Organizations.....	66
9.3.1	SQAP Roles and Responsibilities.....	66
9.3.2	Required Skills.....	66
9.3.3	Escalation Mechanism .....	67

---



9.4	Reporting.....	67
9.4.1	Verification and Validation of Requirements .....	67
9.4.2	Verification.....	68
9.4.3	Validation .....	68
9.5	Approvals.....	68
10	QA Work Products Audit Checklists .....	69
11	Test Plan.....	74
11.1	Introduction .....	74
11.2	Test objectives .....	74
11.2.1	Extent of tests .....	74
11.3	Test levels .....	74
11.3.1	Black box tests .....	74
11.3.2	Component test .....	75
11.3.3	Integration Test.....	75
11.3.4	System Test .....	75
11.3.5	Acceptance Test .....	75
11.4	Features to be Tested/Features Not to be Tested .....	75
11.4.1	Features to be tested.....	75
11.4.2	Items that will not be tested.....	76
11.5	Approach .....	76
11.5.1	General Test Strategy.....	76
11.5.2	Integration Test Strategy .....	76
11.6	Pass/Fail Criteria .....	76
11.6.1	Component Pass/Fail criteria.....	77
11.6.2	Integration Pass/Fail criteria.....	77
11.6.3	System Pass/Fail criteria .....	77
11.7	Suspension/Resuming Criteria .....	77
11.7.1	Automated Unit Test Suite .....	78
11.7.2	Build Acceptance Test (BAT) .....	78
11.7.3	Regression Testing .....	78
11.7.4	System Design Changes .....	78



11.8	Test Deliverables .....	79
11.9	Testing Tasks.....	79
11.10	Test Environment .....	79
11.11	Staffing and Training Needs.....	79
11.12	Risks and Contingences .....	79
11.13	Approvals .....	80
12	Test Procedures .....	81
12.1	Adding a student to some program.....	81
12.1.1	Purpose .....	81
12.1.2	Test Procedure Identifier .....	81
12.1.3	Input Specifications.....	81
12.1.4	Output Specifications.....	81
12.1.5	Procedure Steps .....	81
12.2	Testing the eligibility of some student to finish his study.....	81
12.2.1	Purpose .....	81
12.2.2	Test Procedure Identifier .....	81
12.2.3	Input Specifications.....	81
12.2.4	Output Specifications.....	81
12.2.5	Procedure Steps .....	82
12.3	Updating student basic information.....	82
12.3.1	Purpose .....	82
12.3.2	Test Procedure Identifier .....	82
12.3.3	Input Specifications.....	82
12.3.4	Output Specifications.....	82
12.3.5	Procedure Steps .....	82
12.4	Updating student past grades information .....	82
12.4.1	Purpose .....	82
12.4.2	Test Procedure Identifier .....	83
12.4.3	Input Specifications.....	83
12.4.4	Output Specifications.....	83
12.4.5	Procedure Steps .....	83



12.5	Updating student program information.....	83
12.5.1	Purpose .....	83
12.5.2	Test Procedure Identifier .....	83
12.5.3	Input Specifications.....	83
12.5.4	Output Specifications.....	84
12.5.5	Procedure Steps .....	84
12.6	Register a student to a program.....	84
12.6.1	Purpose .....	84
12.6.2	Test Procedure Identifier .....	84
12.6.3	Input Specifications.....	84
12.6.4	Output Specifications.....	84
12.6.5	Procedure Steps .....	84
12.7	Get number of years since student registration in some program.....	85
12.7.1	Purpose .....	85
12.7.2	Test Procedure Identifier .....	85
12.7.3	Input Specifications.....	85
12.7.4	Output Specifications.....	85
12.7.5	Procedure Steps .....	85
12.8	Deleting a student from the whole system .....	85
12.8.1	Purpose .....	85
12.8.2	Test Procedure Identifier .....	85
12.8.3	Input Specifications.....	86
12.8.4	Output Specifications.....	86
12.8.5	Procedure Steps .....	86
12.9	Maintaining Courses Category .....	86
12.9.1	Purpose .....	86
12.9.2	Test Procedure Identifier .....	86
12.9.3	Input Specifications.....	86
12.9.4	Output Specifications.....	86
12.9.5	Procedure Steps .....	86
12.10	Maintaining Courses Grades Categories .....	87




---

12.10.1	Purpose.....	87
12.10.2	Test Procedure Identifier .....	87
12.10.3	Input Specifications .....	87
12.10.4	Output Specifications .....	87
12.10.5	Procedure Steps .....	87
12.11	Maintaining Professors Titles Categories .....	87
12.11.1	Purpose.....	87
12.11.2	Test Procedure Identifier .....	88
12.11.3	Input Specifications .....	88
12.11.4	Output Specifications .....	88
12.11.5	Procedure Steps .....	88
12.12	Adding/Updating a course.....	88
12.12.1	Purpose.....	88
12.12.2	Test Procedure Identifier .....	88
12.12.3	Input Specifications .....	88
12.12.4	Output Specifications .....	88
12.12.5	Procedure Steps .....	89
12.13	Adding a course with errors .....	89
12.13.1	Purpose.....	89
12.13.2	Test Procedure Identifier .....	89
12.13.3	Input Specifications .....	89
12.13.4	Output Specifications .....	89
12.13.5	Procedure Steps .....	89
12.14	Adding a complementary course.....	90
12.14.1	Purpose.....	90
12.14.2	Procedure Identifier .....	90
12.14.3	Input Specifications .....	90
12.14.4	Output Specifications .....	90
12.14.5	Procedure Steps .....	90
12.15	Deleting a course .....	91
12.15.1	Purpose.....	91

---



12.15.2	Test Procedure Identifier .....	91
12.15.3	Input Specifications .....	91
12.15.4	Output Specifications .....	91
12.15.5	Procedure Steps .....	91
12.16	Adding course to some program in some semester .....	91
12.16.1	Purpose.....	91
12.16.2	Test Procedure Identifier .....	91
12.16.3	Input Specifications .....	91
12.16.4	Output Specifications .....	91
12.16.5	Procedure Steps .....	91
12.17	Getting courses of some program in some semester .....	92
12.17.1	Purpose.....	92
12.17.2	Test Procedure Identifier .....	92
12.17.3	Input Specifications .....	92
12.17.4	Output Specifications .....	92
12.17.5	Procedure Steps .....	92
12.18	Delete some course in some program and some semester.....	93
12.18.1	Purpose.....	93
12.18.2	Test Procedure Identifier .....	93
12.18.3	Input Specifications .....	93
12.18.4	Output Specifications .....	93
12.18.5	Procedure Steps .....	93
12.19	Update some course in some program and some semester .....	93
12.19.1	Purpose.....	93
12.19.2	Test Procedure Identifier .....	93
12.19.3	Input Specifications .....	94
12.19.4	Output Specifications .....	94
12.19.5	Procedure Steps .....	94
12.20	Adding course to some program in some semester with some errors.....	94
12.20.1	Purpose.....	94
12.20.2	Test Procedure Identifier .....	94





12.20.3	Input Specifications .....	94
12.20.4	Output Specifications .....	94
12.20.5	Procedure Steps .....	94
13	Configuration Management Plan.....	96
13.1	Introduction .....	96
13.2	Roles and Responsibilities .....	96
13.2.1	Configuration Control Board (CCB) .....	96
13.2.2	Project Sponsor .....	96
13.2.3	Project Manager .....	96
13.2.4	Configuration Manager .....	96
13.2.5	Lead Engineers .....	97
13.2.6	Engineers.....	97
13.3	Configuration Control .....	97
13.4	Configuration Management Database (CMDB).....	98
13.5	Configuration Status Accounting.....	99
13.6	Configuration Audits.....	99
14	Configuration Management Environment.....	101
14.1	Configuration management environment: .....	103
14.1.1	Repo structure: .....	103
15	Templates.....	105
15.1	Quality Assurance Audit Reports.....	105
15.2	Change Request Template.....	107
15.3	Change Log Template .....	109
15.4	Risk Register.....	110
16	References .....	111



## LIST OF TABLES

Table 1 Preliminary project plan .....	14
Table 2 Data Dictionary.....	40
Table 3 Requirements Traceability Matrix.....	57
Table 4 Process Activity Task Matrix.....	61
Table 5 SQAP Roles and Responsibilities .....	66
Table 6 Risk categories.....	67
Table 7 QA Work Products Audit Checklists .....	69
Table 8 Configuration Management Roles and Responsibilities .....	101

## LIST OF FIGURES

Figure 1 Program Modules Decomposition .....	23
Figure 2 Overall ERD .....	24
Figure 3 Program Course Layer ERD .....	25
Figure 4 Status Layer ERD .....	26
Figure 5 User and Privileges Layers ERD .....	27
Figure 6 Scientific degree Layer ERD .....	28
Figure 7 Student Program Layer ERD.....	29
Figure 8 Configuration Management Repository .....	104



## 1 Statement of work

### 1.1 Introduction/Background

Computer System Engineering department at Faculty of Engineering has recently approved the graduate studies project. It will be a website that helps students to check courses, register on a course, checking grade of his courses and withdraw any course. Also It helps professor to add his courses.

### 1.2 Scope of Work

The scope of work for the graduate studies project includes all processes starting from registration on Master/PhD ending to getting the degree. Specific deliverables and milestones will be listed in the Work Requirements and Schedules and Milestones sections of this SOW.

### 1.3 Period of Performance

The period of performance for the graduate studies project is 1 month beginning on 1 Dec 2015 through 1 Jan 2015. All work must be scheduled to complete within this timeframe. Any modifications or extensions will be requested. they will be reviewed and discussed.

### 1.4 Place of Performance

It will be required to meet with Secretary at Computer System Engineering department to get all requirement and he is responsible to give us all detailed requirements and all reports that needed to be implemented. We will deploy the website at the department's server to be accessible by all students and professors.

### 1.5 Work Requirements

As part of the graduate studies project the development team will be responsible for performing tasks throughout various stages of this project. The following is a list of these tasks which will result in the successful completion of this project:

- Registration process for Student
- Professor registration
- Adding/Updating/Removing course
- Course registration for student
- Add or remove a course for a student
- Calculating GPA for a student
- Required Reports
  - o Print reports for students (courses taken, grades, GPA....)
  - o Print reports for courses (number of students attended, credit hours, professors instructed...)
  - o Print reports for professors (what they did in a semester, hours they gave....)
  - o Find if the master program is finished by the student or not (TOEFEL, making committee, finishing seminar...)



- 
- Find number of years of registration of some student
  - Find students who dropped a term or more without taking any course or doing anything
  - Find students who have passed two years and didn't get supervision paper.
  - Having 5 scientific degrees
  - Having 3 semesters for education
  - Number of allowed hours to register per semester.



## 2 Minutes of Meetings

DATE:	NOV 21, 2015
TIME:	12:00 PM TO 1:00 PM
LOCATION:	FACULTY OF ENGINEERING
CHAIR:	ISLAM

### 2.1 Meeting Objectives

We are checking tasks division document and select tasks that will be done during the next week.

### 2.2 Action Item Review

Islam

1. Task division - DONE

### 2.3 Schedule Review

Work Completed

- Tasks division

### 2.4 Planned Work for the Next Two Weeks

- Statement of work
- Partial Project Management Plan
- Requirements Elicitation Plan
- Requirements Management DB
- Configuration Management Environment
- Project Management Plan
- Risk Management Plan
- Minutes of Meetings
- Product Development Plan
- Requirements Traceability Matrix
- Peer Review Plan
- Peer Review Checklists
- Defects Summary Reports
- Quality Assurance Plan
- QA Work Products Audit Checklists
- Configuration Management Plan
- Functional Configuration Audit Reports
- Physical Configuration Audit Reports



---

## 2.5 New Action Items

- Statement of work - Ahmed Badie, Tuesday 24 Nov 2015
- Partial Project Management Plan - Ahmed Badie, Tuesday 24 Nov 2015
- Requirements Elicitation Plan N - Mohamed Emam, Tuesday 24 Nov 2015
- Requirements Management DB - Mohamed Emam, Tuesday 24 Nov 2015
- Configuration Management Environment - Mohamed Madian, Tuesday 24 Nov 2015
- Project Management Plan - Islam, Tuesday 24 Nov 2015
- Risk Management Plan N - Samar, Tuesday 24 Nov 2015
- Minutes of Meetings - Ahmed Badie, Tuesday 24 Nov 2015
- Product Development Plan - Mohamed Emam, Mohamed Madian, Tuesday 24 Nov 2015
- Requirements Traceability Matrix - Arsany, Tuesday 24 Nov 2015
- Peer Review Plan - Mohamed Emam, Tuesday 24 Nov 2015
- Peer Review Checklists - Mohamed Emam, Tuesday 24 Nov 2015
- Defects Summary Reports - Mohamed Madian, Tuesday 24 Nov 2015
- Quality Assurance Plan N - Islam, Tuesday 24 Nov 2015
- QA Work Products Audit Checklists - Islam, Tuesday 24 Nov 2015
- Configuration Management Plan N - Samar, Tuesday 24 Nov 2015
- Functional Configuration Audit Reports - Mohamed Madian, Tuesday 24 Nov 2015
- Physical Configuration Audit Reports - Mohamed Madian, Tuesday 24 Nov 2015



## 3 Project Management Plan

### 3.1 Preliminary project plan

Table 1 Preliminary project plan

ID	Name	Duration	Start	Finish	Predecessors
0	Software Development	27.88 days	November 30, 2015 8:00 AM	January 6, 2016 4:00 PM	
1	Scope	1 day	November 30, 2015 8:00 AM	November 30, 2015 5:00 PM	
2	Determine project scope	4 hrs	November 30, 2015 8:00 AM	November 30, 2015 12:00 PM	
3	Define preliminary resources	2 hrs	November 30, 2015 1:00 PM	November 30, 2015 3:00 PM	2
4	Secure core resources	2 hrs	November 30, 2015 3:00 PM	November 30, 2015 5:00 PM	3
5	Scope complete	0 days	November 30, 2015 5:00 PM	November 30, 2015 5:00 PM	4
6	Analysis/Software Requirements	5 days	December 1, 2015 8:00 AM	December 7, 2015 5:00 PM	
7	Conduct needs analysis	1 day	December 1, 2015 8:00 AM	December 1, 2015 5:00 PM	5
8	Draft preliminary software specifications	1 day	December 2, 2015 8:00 AM	December 2, 2015 5:00 PM	7
9	Review software specifications/budget with team	3 hrs	December 3, 2015 8:00 AM	December 3, 2015 11:00 AM	8
10	Incorporate feedback on software specifications	1 hr	December 3, 2015 11:00 AM	December 3, 2015 12:00 PM	9
11	Develop delivery timeline	2 hrs	December 3, 2015 1:00 PM	December 3, 2015 3:00 PM	10
12	Secure required resources	2 hrs	December 3, 2015 3:00 PM	December 3, 2015 5:00 PM	11
13	Analysis complete	2 days	December 4, 2015 8:00 AM	December 7, 2015 5:00 PM	12
14	Design	8 days	December 8, 2015 8:00 AM	December 17, 2015 5:00 PM	
15	Review preliminary software specifications	2 days	December 8, 2015 8:00 AM	December 9, 2015 5:00 PM	13
16	Develop functional specifications	2 days	December 10, 2015 8:00 AM	December 11, 2015 5:00 PM	15



ID	Name	Duration	Start	Finish	Predecessors
17	Develop prototype based on functional specifications	2 days	December 14, 2015 8:00 AM	December 15, 2015 5:00 PM	16
18	Review functional specifications	1 day	December 16, 2015 8:00 AM	December 16, 2015 5:00 PM	17
19	Incorporate feedback into functional specifications	1 day	December 17, 2015 8:00 AM	December 17, 2015 5:00 PM	18
20	Design complete	0 days	December 17, 2015 5:00 PM	December 17, 2015 5:00 PM	19
21	Development	9.88 days	December 18, 2015 8:00 AM	December 31, 2015 4:00 PM	
22	Review functional specifications	3 hrs	December 18, 2015 8:00 AM	December 18, 2015 11:00 AM	20
23	Identify modular design parameters	3 hrs	December 18, 2015 11:00 AM	December 18, 2015 3:00 PM	22
24	Assign development staff	1 hr	December 18, 2015 3:00 PM	December 18, 2015 4:00 PM	23
25	Develop code	7 days	December 18, 2015 4:00 PM	December 29, 2015 4:00 PM	24
26	Developer testing (primary debugging)	2 days	December 29, 2015 4:00 PM	December 31, 2015 4:00 PM	25
27	Development complete	0 days	December 31, 2015 4:00 PM	December 31, 2015 4:00 PM	26
28	Testing	12.25 days	December 18, 2015 8:00 AM	January 5, 2016 10:00 AM	
29	Develop unit test plans using product specifications	4 hrs	December 18, 2015 8:00 AM	December 18, 2015 12:00 PM	20
30	Develop integration test plans using product specifications	4 hrs	December 18, 2015 8:00 AM	December 18, 2015 12:00 PM	20
31	Unit Testing	2.38 days	December 31, 2015 4:00 PM	January 5, 2016 10:00 AM	
32	Review modular code	4 hrs	December 31, 2015 4:00 PM	January 4, 2016 10:00 AM	27
33	Test component modules to product specifications	1 hr	January 4, 2016 10:00 AM	January 4, 2016 11:00 AM	32
34	Identify anomalies to product specifications	1 hr	January 4, 2016 11:00 AM	January 4, 2016 12:00 PM	33
35	Modify code	4 hrs	January 4, 2016 1:00 PM	January 4, 2016 5:00 PM	34
36	Re-test modified code	2 hrs	January 5, 2016 8:00 AM	January 5, 2016 10:00 AM	35
37	Unit testing complete	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	36





ID	Name	Duration	Start	Finish	Predecessors
38	Integration Testing	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	
39	Test module integration	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	37
40	Identify anomalies to specifications	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	39
41	Modify code	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	40
42	Re-test modified code	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	41
43	Integration testing complete	0 days	January 5, 2016 10:00 AM	January 5, 2016 10:00 AM	42
44	Deployment	1.63 days	January 5, 2016 10:00 AM	January 6, 2016 4:00 PM	
45	Determine final deployment strategy	1 hr	January 5, 2016 10:00 AM	January 5, 2016 11:00 AM	43
46	Develop deployment methodology	1 hr	January 5, 2016 11:00 AM	January 5, 2016 12:00 PM	45
47	Deploy software	1 day	January 5, 2016 1:00 PM	January 6, 2016 12:00 PM	46
48	Deployment complete	0 days	January 6, 2016 12:00 PM	January 6, 2016 12:00 PM	47
49	Training	0.38 days	January 6, 2016 1:00 PM	January 6, 2016 4:00 PM	
50	training for end users	3 hrs	January 6, 2016 1:00 PM	January 6, 2016 4:00 PM	48
51	Software development complete	0 days	January 6, 2016 4:00 PM	January 6, 2016 4:00 PM	50



## 4 Risk Management Plan

### 4.1 Introduction

As organizations begin new projects they begin operating in an area of uncertainty that comes along with developing new and unique products or services. By doing so, these organizations take chances which results in risk playing a significant part in any project. The purpose of the risk management plan is to establish the framework in which the project team will identify risks and develop strategies to mitigate or avoid those risks. However, before risks can be identified and managed, there are preliminary project elements which must be completed. These elements are outlined in the risk management approach.

This project is considered a medium risk project as it has an overall risk score of 24 on a scale from 0 to 100. The project risk score is the average of the risk scores of the most significant risks to this project. A risk score below 16 is low risk project, a score between 16 and 45 is a medium risk project and a score above 45 is a high risk project.

Before risk management begins it is imperative that a foundation is established for providing structured project information, thus, the following project elements were completed and defined prior to developing this Risk Management Plan:

- Define work scope, schedule, resources, and cost elements
  - Develop project WBS/WBS dictionary
  - Develop master schedule and detailed schedules
  - Estimate project cost and finalize budget
  - Identify required and available resources
  - Establish performance measurement metrics
- Define minimum and maximum baseline thresholds
  - Schedule
  - Resources
  - Cost
- Baseline reporting requirements
  - Format
  - Frequency of distribution
  - Distribution list
- Define Risk Management Roles and Responsibilities
  - Project Manager chairs the risk assessment meetings
  - Project team participates in risk assessment meetings and members serve as meeting recorder and timekeeper
  - Key stakeholders participate in risk assessment meetings
  - Project Sponsor may participate in risk assessment meetings



## 4.2 Top Three Risks

The top three high probability and high impact risks to this project are:

### **Delay in Schedule**

Due to in-experience in software engineering methodology application.

All members work a full time job besides working on the project.

### **New Development environment**

Not all members have the required training for the used tools.

## 4.3 Risk Management Approach

The approach we have taken to manage risks for this project included a methodical process by which the project team identified, scored, and ranked the various risks. The most likely and highest impact risks were added to the project schedule to ensure that the assigned risk managers take the necessary steps to implement the mitigation response at the appropriate time during the schedule. Risk managers will provide status updates on their assigned risks in the bi-weekly project team meetings, but only when the meetings include their risk's planned timeframe. Upon the completion of the project, during the closing process, the project manager will analyze each risk as well as the risk management process. Based on this analysis, the project manager will identify any improvements that can be made to the risk management process for future projects. These improvements will be captured as part of the lessons learned knowledge base.

## 4.4 Risk Identification

For this project, risk identification was conducted in the initial project risk assessment meeting. The method used by the project team to identify risks was the Crawford Slip method. The project manager chaired the risk assessment meeting and distributed notepads to each member of the team and allowed 10 minutes for all team members to record as many risks as possible.

### **Expert Interview**

Two Expert Interviews were held for this project. The interviews revealed several risks which were then mitigated by making changes to the project plan. The remaining risks are included in the Risk Register.

### **Risk Assessment Meeting**

A risk assessment meeting was held with key team members and stakeholders. The risks identified during this meeting were added to the project plan and Risk Register.



---

## Historical Review of Similar Projects

The project team reviewed the history of similar projects in order to determine the most common risks and the strategies used to mitigate those risks.

### 4.5 Risk Qualification and Prioritization

In order to determine the severity of the risks identified by the team, a probability and impact factor was assigned to each risk. This process allowed the project manager to prioritize risks based upon the effect they may have on the project. The project manager utilized a probability-impact matrix to facilitate the team in moving each risk to the appropriate place on the chart.

Once the risks were assigned a probability and impact and placed in the appropriate position on the chart, the recorder captured the finished product and the project manager moved the process on to the next step: risk mitigation/avoidance planning.

### 4.6 Risk Monitoring

The most likely and greatest impact risks have been added to the project plan to ensure that they are monitored during the time the project is exposed to each risk. At the appropriate time in the project schedule a Risk Manager is assigned to each risk. During the bi-weekly project team meeting the Risk Manager for each risk will discuss the status of that risk; however, only risks which fall in the current time period will be discussed. Risk monitoring will be a continuous process throughout the life of this project. As risks approach on the project schedule the project manager will ensure that the appropriate risk manager provides the necessary status updates which include the risk status, identification of trigger conditions, and the documentation of the results of the risk response.

### 4.7 Risk Mitigation and Avoidance

The project manager has led the project team in developing responses to each identified risk. As more risks are identified, they will be qualified and the team will develop avoidance and mitigation strategies. These risks will also be added to the Risk Register and the project plan to ensure they are monitored at the appropriate times and are responded to accordingly.

The risks for this project will be managed and controlled within the constraints of time, scope, and cost. All identified risks will be evaluated in order to determine how they affect this triple constraint. The project manager, with the assistance of the project team, will determine the best way to respond to each risk to ensure compliance with these constraints.

In extreme cases it may be necessary to allow flexibility to one of the project's constraints. Only one of the constraints for this project allows for flexibility as a last resort. If necessary, funding may be added to the project to allow for more resources in order to meet the time (schedule) and scope constraints. Time and scope are firm constraints and allow for no



flexibility. Again, the cost constraint is flexible only in extreme cases where no other risk avoidance or mitigation strategy will work.

#### 4.8 Risk Register

The Risk Register for this project is a log of all identified risks, their probability and impact to the project, the category they belong to, mitigation strategy, and when the risk will occur. The register was created through the initial project risk management meeting led by the project manager. During this meeting, the project team identified and categorized each risk. Additionally, the team assigned each risk a score based on the probability of it occurring and the impact it could potentially have. The Risk Register also contains the mitigation strategy for each risk as well as when the risk is likely to occur.

Based on the identified risks and timeframes in the risk register, each risk has been added to the project plan. At the appropriate time in the plan—prior to when the risk is most likely to occur—the project manager will assign a risk manager to ensure adherence to the agreed upon mitigation strategy. Each risk manager will provide the status of their assigned risk at the bi-weekly project team meeting for their risk's planned timeframe.

The Risk Register template can be found in Templates sections of this document



## 5 Detailed Design

### 5.1 Introduction

#### 5.1.1 Purpose

The purpose of this document is to Describes the architecture and system design of SoftwareEngineeringProject and to list all features provided by the application and also the Application Interface and database design.

#### 5.1.2 Scope

The need from the application is to provide simple practical solution for registering in postgraduate programs in the faculty of engineering and to keep track of student throw all stages of the program.

#### 5.1.3 Overview

This document will describe the idea and intention to develop this application and will talk in detailed with the design and architecture of the application and also will describe in details the database design and the front end of the application to provide the needed features simply.

### 5.2 System Overview

The application divided into 6 main Data Layer (Program course, Scientific Degree, Status, User, Privilege, Student Program) where the business and logics are scattered across them in order to encapsulate business and to ensure the modularity aspects. Such that:

- Scientific Degree
  - It is used document and manages student scientific degree.
- Program Course:
  - It Is used to create semester courses for both master and doctoral programs.
- User
  - It Is used to manage and CRUD operations for users type like (students, professors).
- Status
  - it is used to track the progress of student across the program registered in.
- Privilege
  - It Is used to ensure the protection application by adding privileges across the whole application.
- Student Program
  - Is used to track the registration of student in the program and maintain the professor student relationship.



Then comes to the UI sections where divided into 5 sections (Metadata, programs, students, Users and Reports) each UI section will be responsible for maintaining the databases entity and display the information needed.

## 5.3 System Architecture

### 5.3.1 Architectural Design

The System divided into 5 main modules, each module has its encapsulated logic and maintain its tables, the system divided into modules in order to be easy developed and also maintained in further phases if the project needed to be enhanced, each module have its own screen and tables, where data are only maintained and managed only by it screens, our modules are (Metadata Module, Programs Module, Students Module, Users Module, Reports Module).

The following section we will give brief about each module and affected important tables:

**Metadata Module:** the functionality from this module is to maintain the metadata of the system like (courses, etc.) the data that important to start up the application and do the daily tasks.

**Program Module:** the main two functions this module provides are define a program, and assign courses to that program, and register new program and to maintain it, also to setup its classes and maintain them, also to add program classes to the needed semester and course classes to the courses, and maintain all that data with the semester needed.

**Students Module:** the main function that module provides are adding scientific degree to the students and to add courses for them, and also set the grades for this students and their courses.

**Reports Module:** in this module the function required to generate the needed reports from the system.



### 5.3.2 Decomposition Description

- Scientific Degree
  - It is used document and manages student scientific degree.
- Program Course
  - It is used to create semester courses for both master and doctoral programs.
- User
  - It is used to manage and CRUD operations for users type like (students, professors).
- Status
  - it is used to track the progress of student across the program registered in.
- Privilege
  - It is used to ensure the protection application by adding privileges across the whole application.
- Student Program
  - Is used to track the registration of student in the program and maintain the professor student relationship.

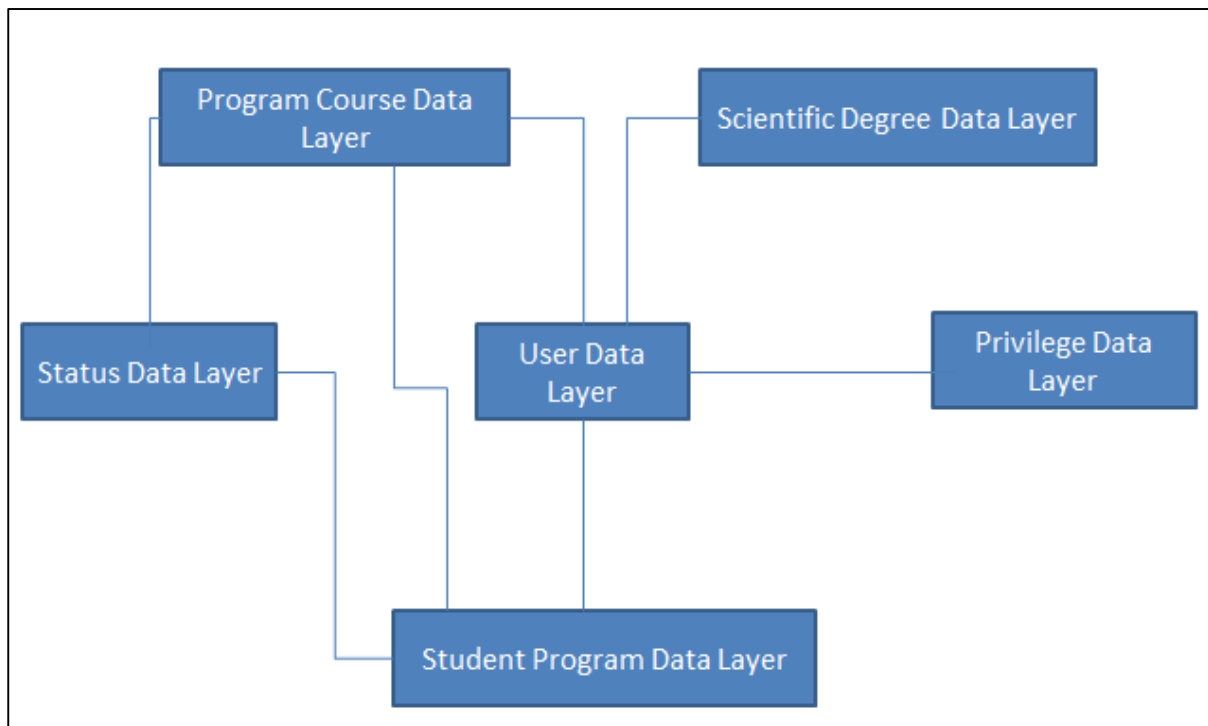


Figure 1 Program Modules Decomposition



### 5.3.3 Design Rationale

In the ERD diagram we illustrate how the relation between data entities is managed and how the data assembled together. The main entities of the application as it will be described its functionality.

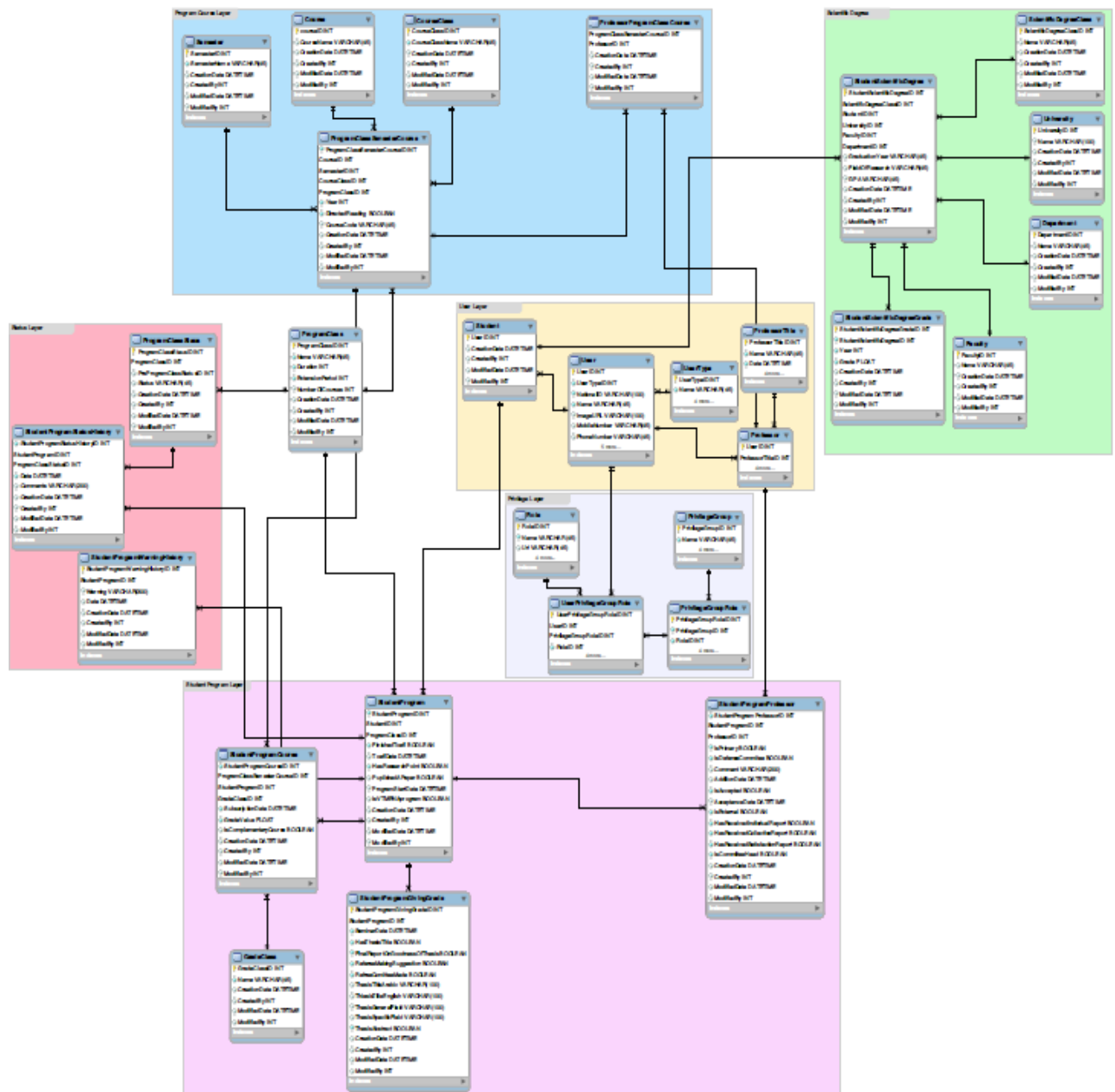


Figure 2 Overall ERD

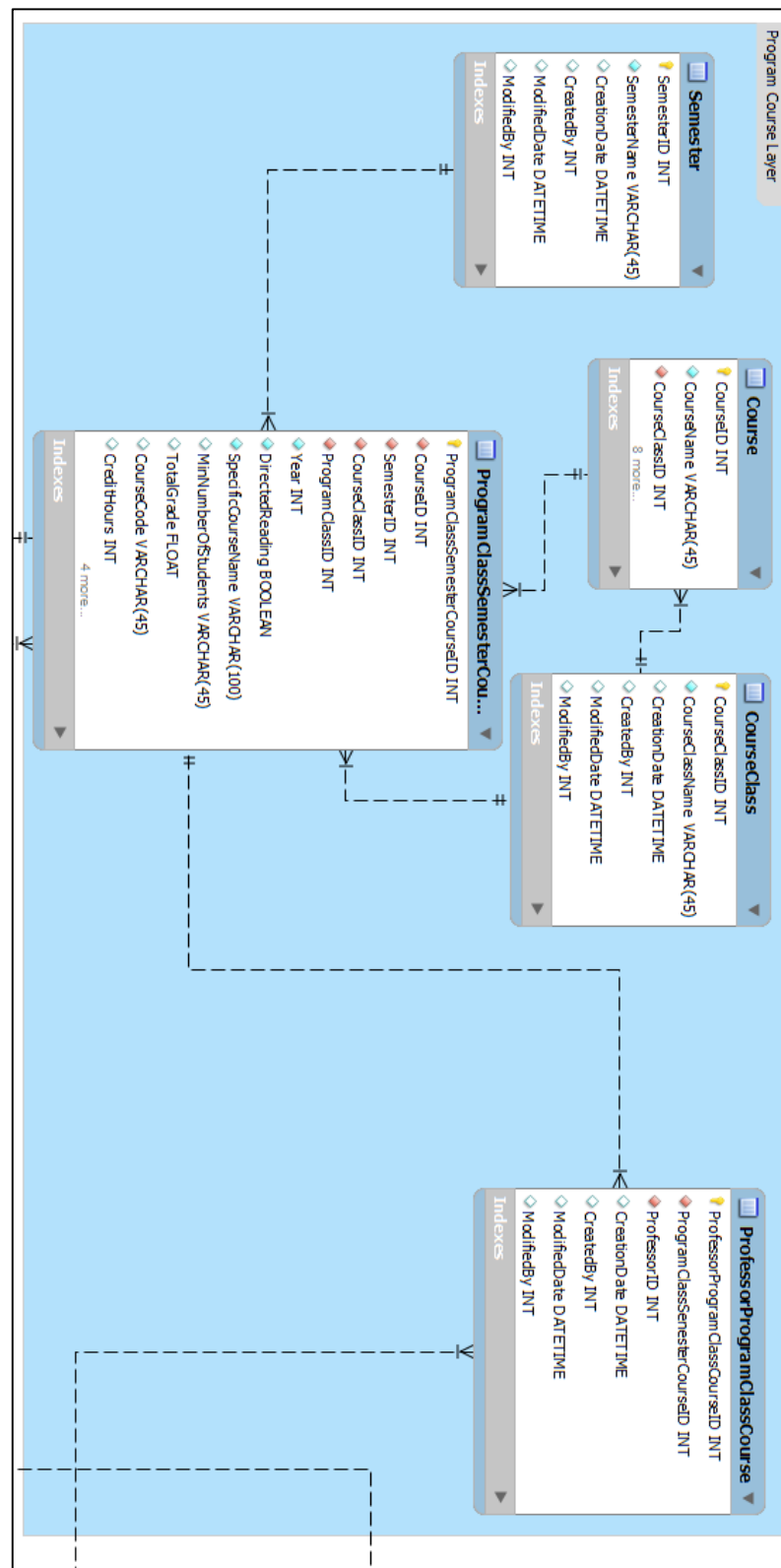


Figure 3 Program Course Layer ERD

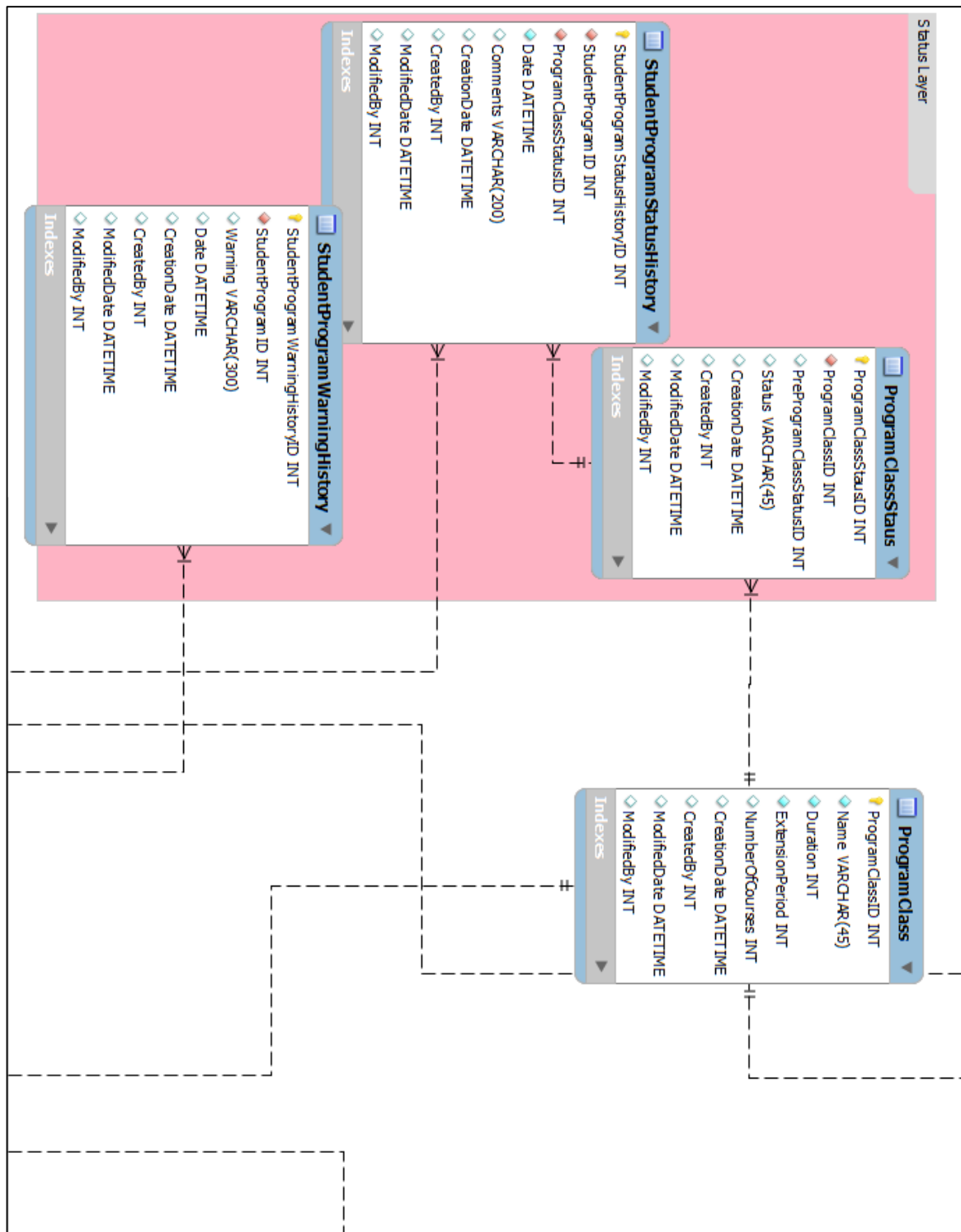


Figure 4 Status Layer ERD

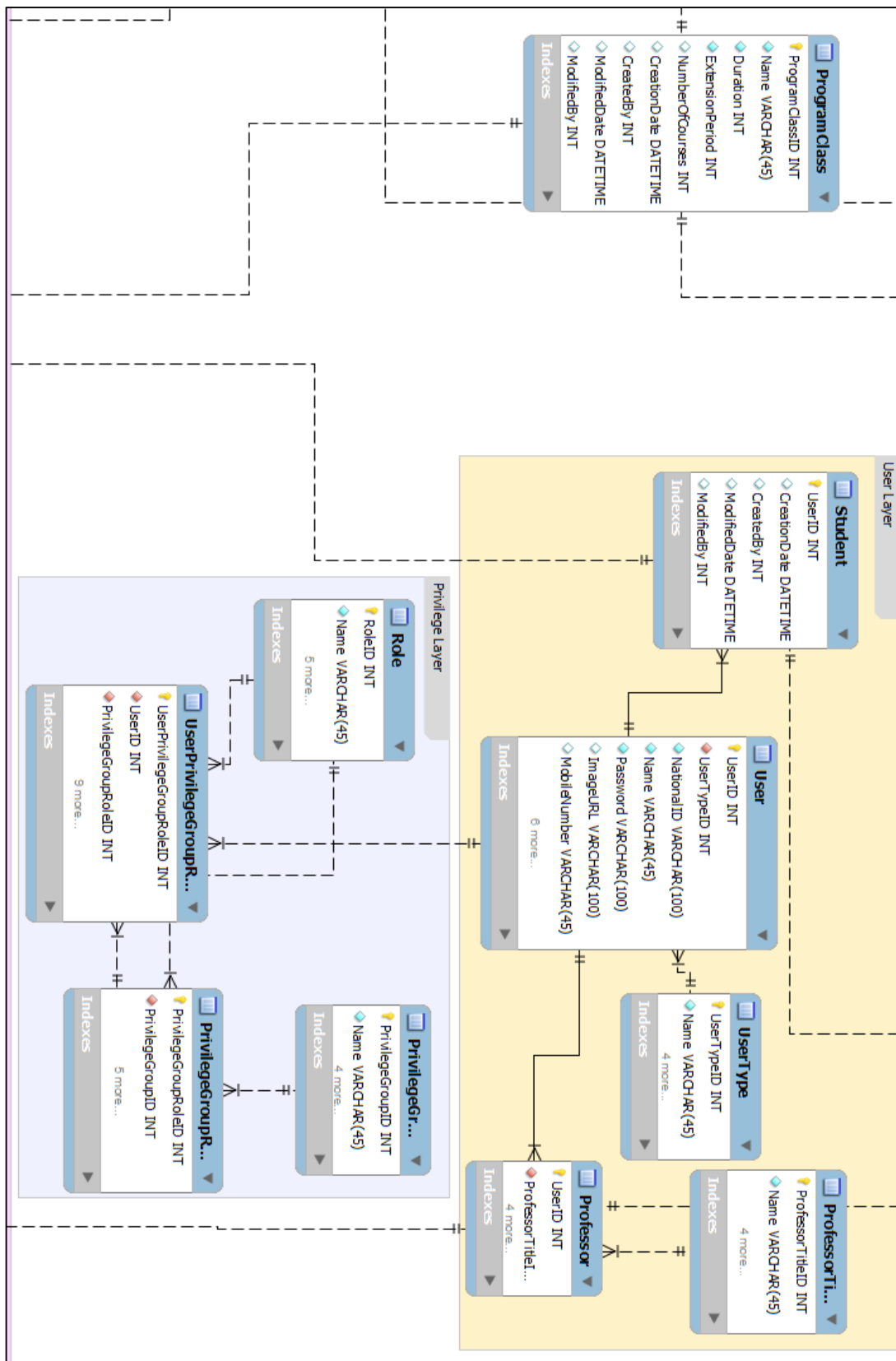


Figure 5 User and Privileges Layers ERD

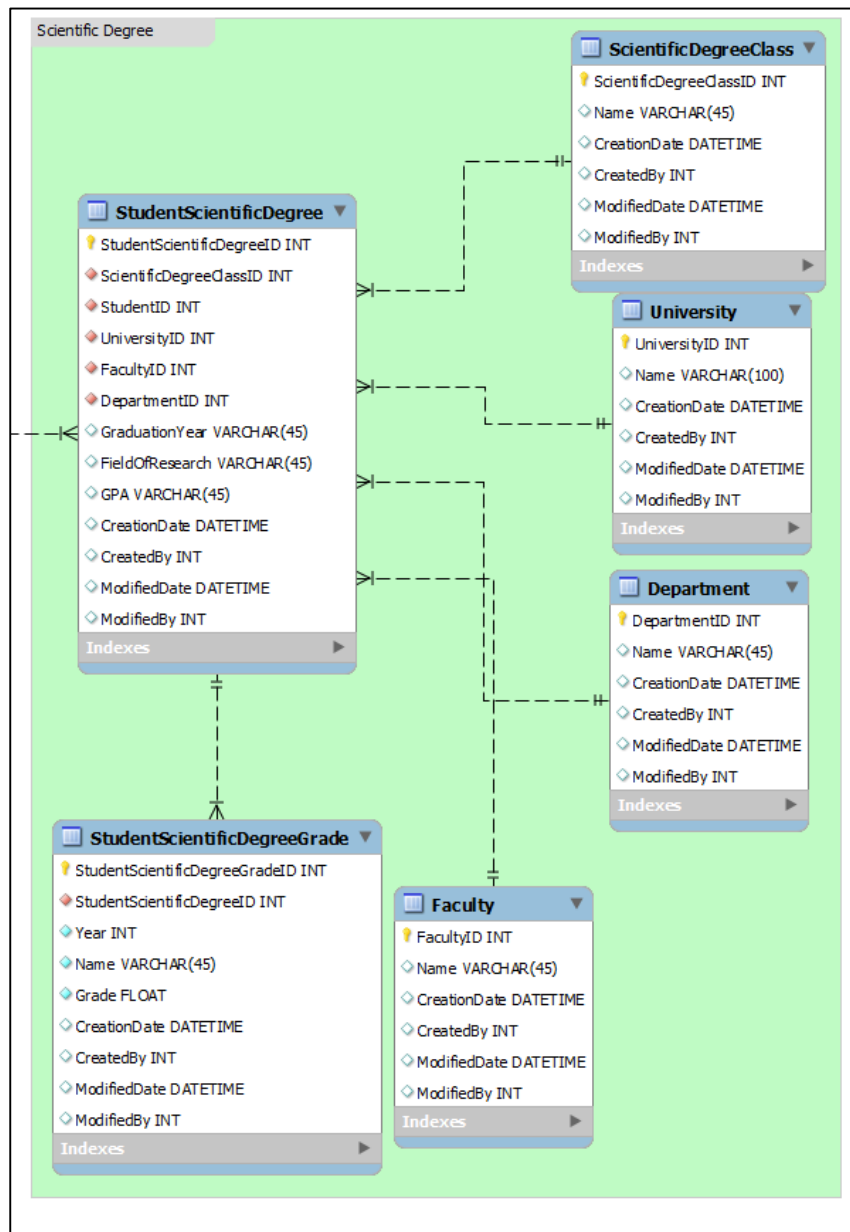


Figure 6 Scientific degree Layer ERD

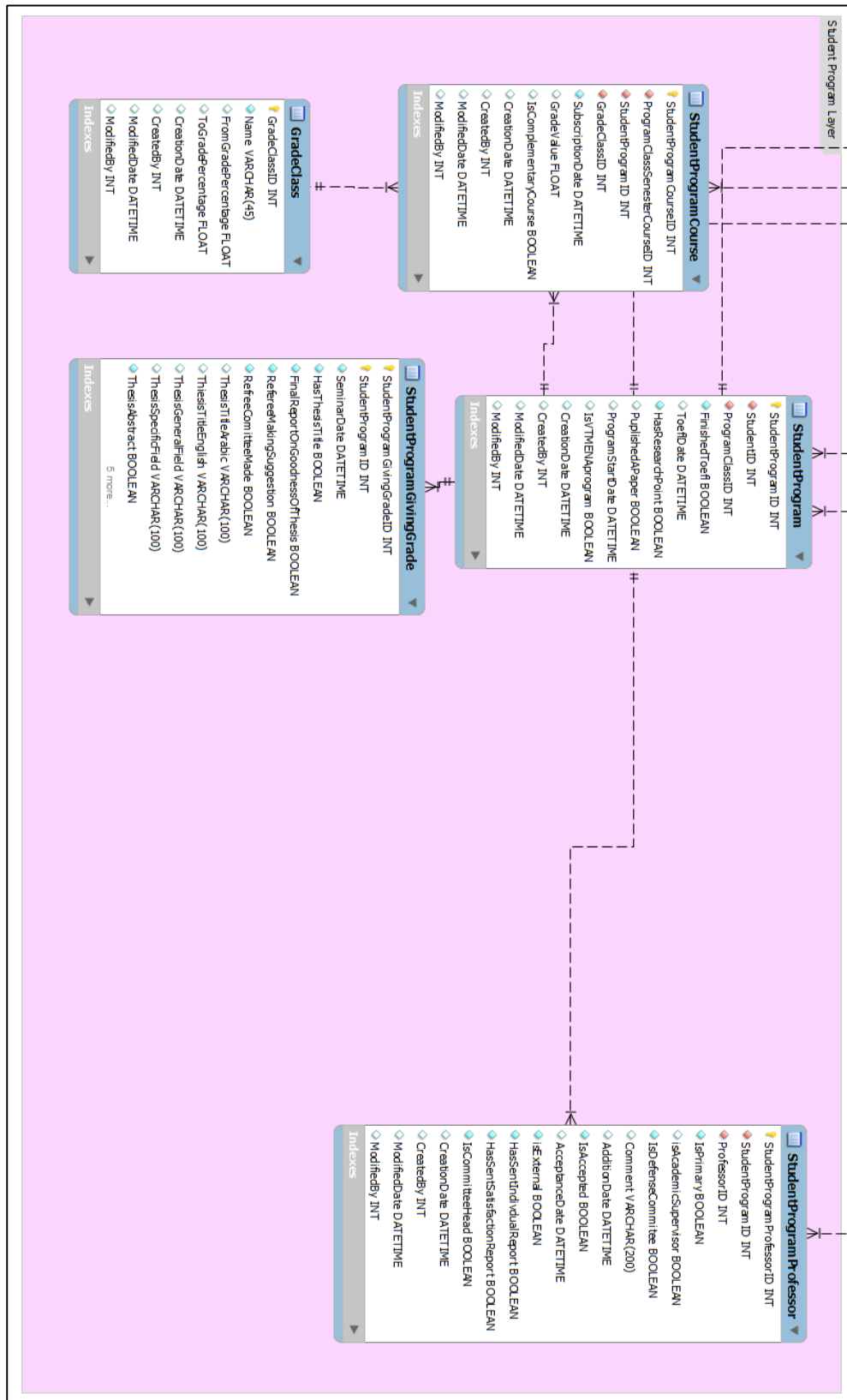


Figure 7 Student Program Layer ERD



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Course	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* courseID	Int		Positive Value, Not Null	
CourseName	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> CourseClass	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* CourseClassID	Int		Positive Value, Not Null	
CourseClassName	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Semester	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* SemesterID	Int		Positive Value, Not Null	
SemesterName	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



Project: PGSA			Page 1/1
Entity: ProgramClass	Date: 1/17/2016		
Attribute	Type	Size	Validations
* ProgramClassID	Int		Positive Value, Not Null
Name	Varchar	45	Null
Duration	Int		Null
ExtensionPeriod	Int		Null
NumberOfCourses	Int		Null
CreationDate	DateTime		Null
CreatedBy	Int		Null
ModifiedDate	DateTime		Null
ModifiedBy	Int		Null

Project: PGSA				Page 1/1
Entity: ProgramClassSenesterCourse	Date: 1/17/2016			
Attribute	Type	Size	Validations	
* ProgramClassSenesterCourseID	Int		Positive Value, Not Null	
CourseID	Int		Null	
SemesterID	Int		Null	
CourseClassID	Int		Null	
ProgramClassID	Int		Null	
Year	Int		Null	
DirectedReading	TINYInt		Null	
CourseCode	Varchar	45	Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

Project: PGSA			Page 1/1
Entity: ProfessorTitle	Date: 1/17/2016		
Attribute	Type	Size	Validations
* ProfessorTitleID	Int		Positive Value, Not Null
Name	Varchar	45	Null
Date	DateTime		Null
CreatedBy	Int		Null
ModifiedDate	DateTime		Null
ModifiedBy	Int		Null





<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> UserType	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* UserID	Int		Positive Value, Not Null	
Name	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> User	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* UserID	Int		Positive Value, Not Null	
UserTypeID	Int	45	Null	
NationalID	Int		Null	
Name	Varchar	45	Null	
ImageURL	Varchar	45	Null	
MobileNumber	Varchar	45	Null	
PhoneNumber	Varchar	45	Null	
Email	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Professor	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* UserID	Int		Positive Value, Not Null	
ProfessorTitleID	Int		Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> ProfessorProgramClassCourse	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* ProgramClassSenesterCourseID	Int		Positive Value, Not Null	
ProfessorID	Int		Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Student	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* UserID	Int		Positive Value, Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgram	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* StudentProgramID	Int		Positive Value, Not Null	
StudentID	Int		Null	
ProgramClassID	Int		Null	
FinishedToefl	TintInt	1	Null	
ToeflDate	DateTime		Null	
HasResearchPoint	TintInt	1	Not Null	
PuplishedAPaper	TintInt	1	Null	
ProgramStartDate	TintInt	1	Null	
IsVTMENAprogram	TintInt	1	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> GradeClass	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* GradeClassID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgramCourse	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* StudentProgramCourseID	Int		Positive Value, Not Null	
ProgramClassSenesterCourseID	Int		Not Null	
StudentProgramID	Int		Not Null	
GradeClassID	Int		Not Null	
SubscriptionDate	DateTime		Not Null	
GradeValue	Float		Not Null	
IsComplementaryCourse	TinyInt	1	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgramProfessor	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* StudentProgramProfessorID	Int		Positive Value, Not Null	
StudentProgramID	Int		Not Null	
ProfessorID	Int		Not Null	
IsPrimary	TinyInt	1	Not Null	
IsDefenseCommitee	TinyInt		Not Null	
Comment	Float		Not Null	
AdditionDate	TinyInt	1	Null	
IsAccepted	DateTime		Null	
AcceptanceDate	Int		Null	
isExternal	DateTime		Null	



HasReceivedIndividualReport	Int		Null
HasReceivedCollectiveReport	TinyInt	1	Not Null
HasReceivedSatisfactionReport	TinyInt	1	Not Null
IsCommitteeHead	TinyInt	1	Not Null
CreationDate	TinyInt	1	Not Null
CreatedBy	DateTime		Null
ModifiedDate	Int		Null
ModifiedBy	DateTime		Null

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> ProgramClassStaus	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* ProgramClassStausID	Int		Positive Value, Not Null	
ProgramClassID	Int		Not Null	
PreProgramClassStatusID	Int		Null	
status	Varchar	45	Null	
CreationDate	DateTime		Not Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgramStatusHistory	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* StudentProgramStatusHistoryID	Int		Positive Value, Not Null	
StudentProgramID	Int		Not Null	
ProgramClassStatusID	Int		Not Null	
Date	DateTime		Not Null	
Comments	Varchar	100	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgramWarningHistory	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* StudentProgramWarningHistoryID	Int		Positive Value, Not Null	
StudentProgramID	Int		Not Null	
Warning	Int		Null	
Date	DateTime		Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> ScientificDegreeClass	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* ScientificDegreeClassID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> University	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* UniversityID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Faculty	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* FacultyID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Department	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* DepartmentID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentScientificDegree	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
StudentScientificDegreeID	Int		Not Null	
ScientificDegreeClassID	Int		Not Null	
StudentID	Int		Not Null	
UniversityID	Int		Not Null	
FacultyID	Int		Not Null	
DepartmentID	Int		Not Null	
GraduationYear	Varchar	45	Null	
FieldOfResearch	Varchar	45	Null	
GPA	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> Role	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* RoleID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
Url	Varchar	45	Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> PrivilegeGroup	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* PrivilegeGroupID	Int		Positive Value, Not Null	
Name	Varchar	45	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> PrivilegeGroupRole	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
* PrivilegeGroupRoleID	Int		Positive Value, Not Null	
PrivilegeGroupID	Int		Not Null	
RoleID	Int		Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	



<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> UserPrivilegeGroupRole	<b>Date:</b> 1/17/2016			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
UserPrivilegeGroupRoleID	Int		Not Null	
UserID	Int		Not Null	
PrivilegeGroupRoleID	Int		Not Null	
RoleID	Int		Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentScientificDegreeGrade	<b>Date:</b> 1/17/2016			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
StudentScientificDegreeGradeID	Int		Not Null	
StudentScientificDegreeID	Int		Not Null	
PrivilegeGroupRoleID	Int		Not Null	
Year	Int		Not Null	
Grade	Float		Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	





<b>Project:</b> PGSA				<b>Page 1/1</b>
<b>Entity:</b> StudentProgramGivingGrade	<b>Date: 1/17/2016</b>			
<b>Attribute</b>	<b>Type</b>	<b>Size</b>	<b>Validations</b>	
StudentProgramGivingGradeID	Int		Not Null	
StudentProgramID	Int		Not Null	
SeminarDate	DateTime		Not Null	
HasThesisTitle	TinyInt	1	Not Null	
FinalReportOnGoodnessOfThesis	TinyInt	1	Not Null	
RefereeMakingSuggestion	TinyInt	1	Not Null	
RefreeComitteeMade	TinyInt	1	Null	
ThesisTitleArabic	Varchar	100	Null	
ThesisTitleEnglish	Varchar	100	Null	
ThesisGeneralField	Varchar	100	Null	
ThesisSpecificField	Varchar	100	Null	
ThesisAbstract	TinyInt	1	Not Null	
CreationDate	DateTime		Null	
CreatedBy	Int		Null	
ModifiedDate	DateTime		Null	
ModifiedBy	Int		Null	

## 5.4 Data Dictionary

In this section we are listing all the available data entities along with their functions.

Table 2 Data Dictionary

Table Name	Description
Course	The function from this table is to store all courses found in the programs along with their descriptions
CourseClass	The function from this table is to store all course classes found in the programs along with their descriptions
Department	The function from this table is to store Department found in the all faculties along with their descriptions
Faculty	The function from this table is to store all Faculties along with their descriptions
GradeClass	The function from this table is to store all Grades for all classes for all students registered across all programs
PrivilegeGroup	The function from this table is to store all available privilege groups along with their descriptions
PrivilegeGroupRole	The function from this table is to store all roles for the stored privilege groups.



Table Name	Description
Professor	The function from this table is to store all professors with their needed descriptions
ProfessorProgramClassCourse	The function from this table is to store links between professor and program classes
ProfessorTitle	The function from this table is to store all professor titles
ProgramClass	The function from this table is to store all program classes with their descriptions
ProgramClassSemesterCourse	The function from this table is to store link between program classes and semesters
ProgramClassStatus	The function from this table is to store all program class status with their descriptions.
Role	The function from this table is to store all Roles with their descriptions.
ScientificDegreeClass	The function from this table is to store scientific degree class with their descriptions.
Semester	The function from this table is to store all semester with their descriptions.
Student	The function from this table is to store all students along with their descriptions
StudentProgram	The function from this table is to store link between student and programs.
StudentProgramCourse	The function from this table is to store links between student, program and courses.
StudentProgramGivingGrade	The function from this table is to store links between student, program and grade.
StudentProgramProfessor	The function from this table is to store links between student, program and professor.
StudentProgramStatusHistory	The function from this table is to store links between student, program and status history.
StudentProgramWarningHistory	The function from this table is to store link between student, program and warning history.
StudentScientificDegree	The function from this table is to store all student scientific degree grades along with their descriptions.
StudentScientificDegreeGrade	The function from this table is to store all student scientific degree along with their descriptions.
University	The function from this table is to store all registered Universities in the system.
User	The function from this table is to store all Users in the system along with their descriptions
UserPrivilegeGroupRole	The function from this table is to store link between user, privilege group and roles.



---

Table Name	Description
UserType	The function from this table is to store all user types with their descriptions.



## 5.5 Human Interface Design

### 5.5.1 Overview of User Interface

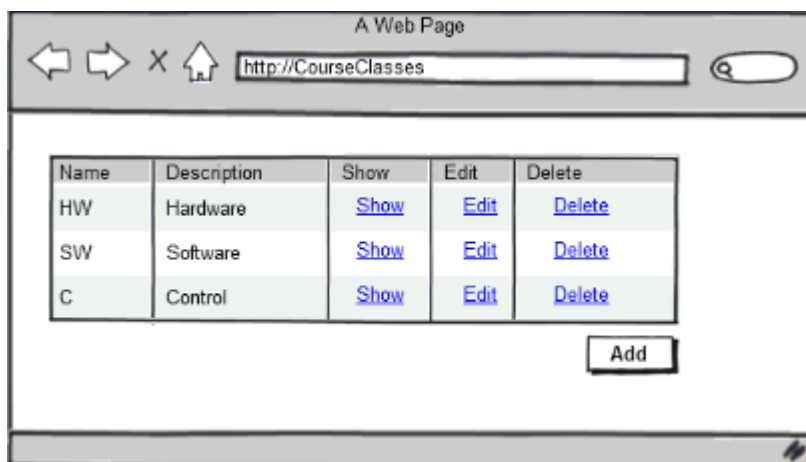
In this section we will illustrate the screens that the developed application provided for the user in order to automate the whole process and give the user of the application the whole information needed to manage and track the students across whole programs and also to adjust the meta data for the application and to provide for the user more convenient way to manage the daily tasks from the CRUD operations, or even the needed reports.

### 5.5.2 Screen Images

#### 5.5.2.1 Meta Data screens

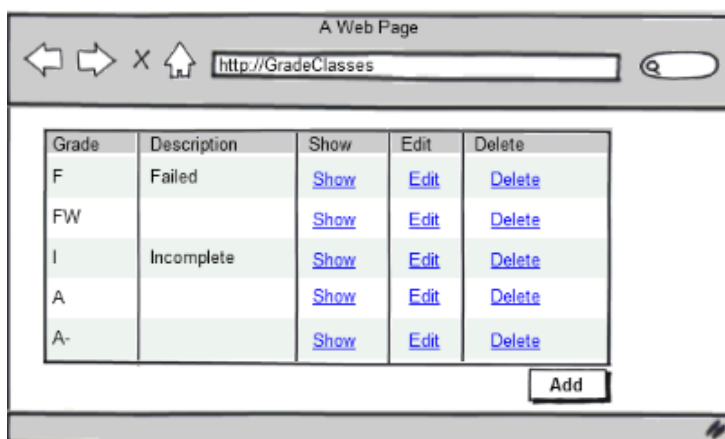
##### 5.5.2.1.1 Course Manage Screen

In Course Manage Screen User Can List Courses types (HW, SW, C, APP ) , and also Can Add, Update or Delete the selected course type.



##### 5.5.2.1.2 Grade Manage Screen

In Grade Manage Screen User Can List Grades available (F, FW, I, A, A- ,...etc.) , and also Can Add, Update or Delete the selected grade.





#### 5.5.2.1.3 Course Manage Screen

In Courses Manage Screen User Can List Courses (Data Mining, Software Engineering , ... etc. ) , and also Can Add, Update or Delete the selected course.

Name	Course Class	Course Code	CreditHours	TotalGrade	Min students number	Show	Edit	Delete
Data Mining	SW	123456	3	150	5	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

#### 5.5.2.1.4 Professor Titles Manage Screen

In Professor Titles Manage Screen User Can List all available titles (TA, DR ) , and also Can Add, Update or Delete the selected Title.

Name	Show	Edit	Delete
TA	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Dr.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

#### 5.5.2.1.5 Scientific Degree Manage Screen

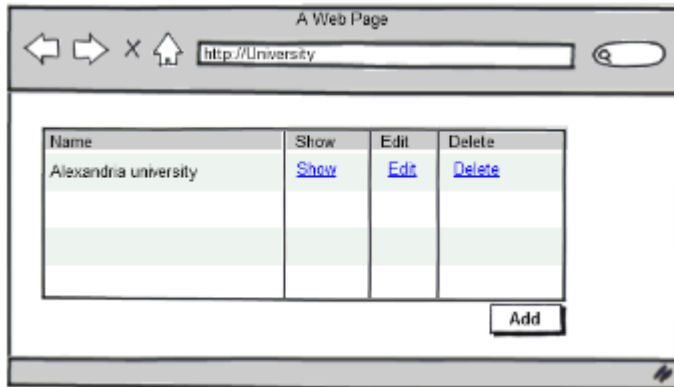
in Scientific Degree Manage Screen User Can List Scientific Degree available (M.Sc, PHD, ,...etc.) , and also Can Add, Update or Delete the selected one.

Name	Show	Edit	Delete
B.Sc.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
M.Sc.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Diploma of graduate studies	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
PHD	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>



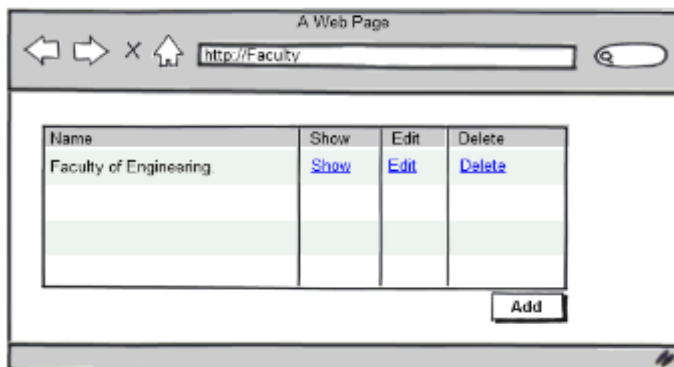
#### 5.5.2.1.6 University Manage Screen

in University Manage Screen User Can List Universities available (Alexandria University ,...etc.) , and also Can Add, Update or Delete the selected one.



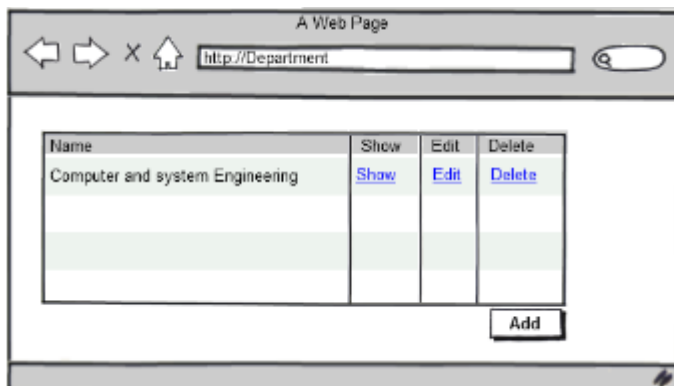
#### 5.5.2.1.7 Faculty Manage Screen

in Faculty Manage Screen User Can List Faculties available (Engineering Faculty ,...etc.) , and also Can Add, Update or Delete the selected one.



#### 5.5.2.1.8 Department Manage Screen

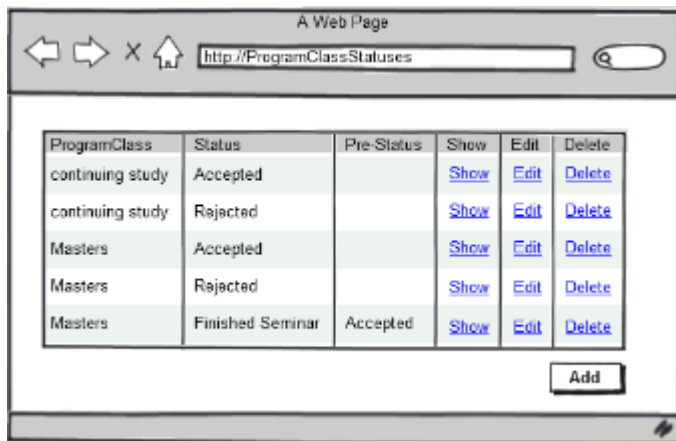
in Department Manage Screen User Can List all Departments available (Computer Science ,...etc.) , and also Can Add, Update or Delete the selected one.





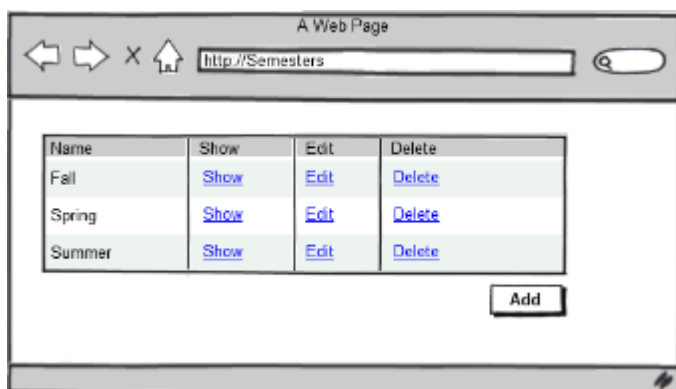
#### 5.5.2.1.9 Program Manage Screen

in Program Manage Screen User Can List all Programs available (Masters,...etc.) , and also Can Add, Update or Delete the selected one.



#### 5.5.2.1.10 Semesters Manage Screen

in Semesters Manage Screen User Can List all Semesters available (Fall, Spring,...etc.) , and also Can Add, Update or Delete the selected one.

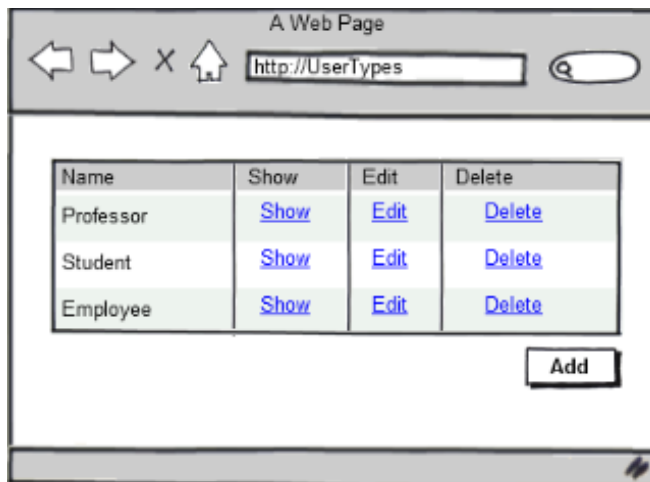




### 5.5.2.2 Privileges screens

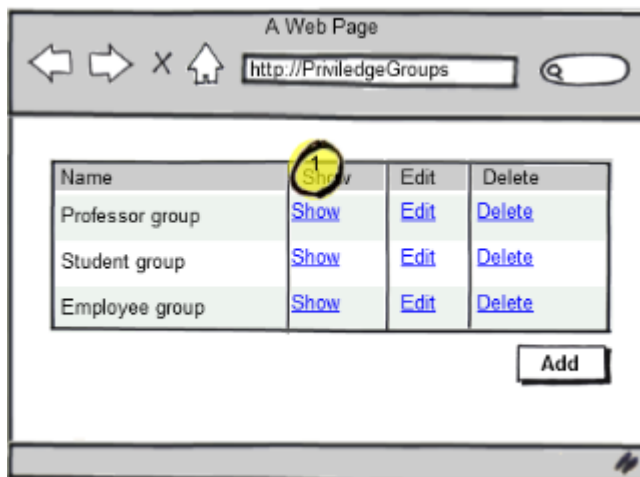
#### 5.5.2.2.1 User Type Manage Screen

in User Type Manage Screen User Can List all User Type available (Professor, Student, Employee ,...etc.) , and also Can Add, Update or Delete the selected one.



#### 5.5.2.2.2 Privilege Groups Manage Screen

in Privilege Groups Manage Screen User Can List all Privilege Groups available (Professor Group, Student Group, Employee Group ,...etc.) , and also Can Add, Update or Delete the selected one.







#### 5.5.2.2.3 Users Manage Screen

in Users Manage Screen User Can List all Users available , and also Can Add, Update or Delete the selected one.

User type	National ID	Name	Mobile Number	Phone Number	Email	Show	Edit	Delete
Employee	12345678912345	Mohamed El Sawy	01002222222	035555555	Ahmed@gmail.com	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**

#### 5.5.2.2.4 Privilege Groups Manage Screen

Page Name	Can Add	Can Edit	Can Delete	Can Show	Show	Edit	Delete
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**

AutoComplete national ID **Add**

Page Name	Can Add	Can Edit	Can Delete	Can Show	Show	Edit	Delete
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
					<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**



### 5.5.2.3 Program screens

#### 5.5.2.3.1 Program Type Manage Screen

in Program Type Manage Screen User Can List Program Types available (Masters, PHD,...etc.) , and also Can Add, Update or Delete the selected one.

Name	Duration	ExtensionPeriod	NumberOfCourses	Show	Edit	Delete
continuing study				<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Masters				<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
PHD				<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**

- Database Tables: ProgramClass

#### 5.5.2.3.2 Programs Manage Screen

in Programs Manage Screen User Can List all Programs available and finished one(Master Spring 2014,...etc.) , and also Can Add, Update or Delete the selected one.

Search

Semester Spring Year 2015 **Search**

- Database Tables: ProgramClassSemesterCourse  
- Select semester automatically to be the current semester  
- year value should be the current year

Program	Semester	Year	Course Class	Course Code	Course	Specific Course Name	Show	Edit	Delete
Masters	Spring	2015	SW	123456	Directed readings computer 1	Data Mining	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**



#### 5.5.2.3.3 Program Edit/Add Screen

in Program Edit/Add Screen User Can Change the details of selected program or just add new one.

#### 5.5.2.4 Program screens

##### 5.5.2.4.1 Students Manage Screen

in Students Manage Screen User Can List all Students available, and also Can Add, Update or Delete the selected one.

National ID	Name	Mobile Number	Phone Number	Email	Show	Edit	Delete
12345678912345	Ahmed Abd-Allah	01002222222	035555555	Ahmed@gmail.com	Show	Edit	Delete



#### 5.5.2.4.2 Students Edit/Add Screen

in Students Edit/Add Screen User view the selected user with all fields to edit, and also list the registered program and the scientific degree for this user.

Personal Info

National ID: 12345678912345

Name: Ahmed Abd-Allah

Mobile Number: 01002222222

Email: Ahmed@gmail.com

Phone Number: 035555555

- Database Tables: Student, User

Scientific Degrees

Scientific Degree Class	University	Faculty	Department	Graduation Year	Field Of Research	GPA	Show	Edit	Delete
							Show	Edit	Delete

- Database Tables: StudentScientificDegree

Add

Programs

Program	Start Date	Finished Toefl	Toefl Date	Has Research Point	Published A Paper	Is VT MENA program	Show	Edit	Delete
Masters	StartDate	<input checked="" type="checkbox"/>	Toefl Date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Show	Edit	Delete

- Database Tables: StudentProgram

Add

#### 5.5.2.4.3 Scientific Degree Edit/Add Screen

in Scientific Degree Edit/Add Screen User view the selected Scientific Degree Edit/Add with all fields to edit, and also list the grades under that Scientific Degree .



A Web Page

http://Student/Scientific Degrees/Add-Show-Edit 4

Scientific Degree Class **B.Sc.**

University **ComboBox**

Faculty **ComboBox**

Department **ComboBox**

GraduationYear **Some text**

Field Of Research **Some text**

GPA **Some text**

Grades

Name	Year	Grade	Show	Edit	Delete
Prep	2010	50	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

**Add**

- Database Tables:  
StudentScientificDegree

- Database Tables:  
StudentScientificDegreeGrade  
- According to the type of Scientific Degree class, the section of Scientific Degree Grade will be visible only for B.Sc.

#### 5.5.2.4.4 Student Programs Courses Mange Screen

in Student Programs Courses Manage Screen User Can Update the selected one, and also view the professors who teach that course.

A Web Page

http://Student/Programs/Courses/Show or Edit or Add 6

Program **Masters**

Semester **Spring** Year **2015**

Specific Course **Data Mining** *Student took this course before*

Course Class **SW**

Course **Directed reading in computer 1**

Course Code **123456**

Number of Credit Hours **3**

Total Grade **150**

☒ Directed Reading

Student Grade **111**

Student Grade Class **A**

**Save**

Professors

Ahmed Ali  
Sara Hamdy

- Database Tables:  
ProfessorProgramClassCourse

- Database Tables:  
StudentProgramCourse, ProgramClassSemesterCourse



#### 5.5.2.5 Student Supervisors Manage Screen

in Student Supervisors Manage Screen User Can Update the selected one.

AutoComplete Professor list ☐ Is Academic supervisor ☐ Is Primary

**Add**

Professor	Is Academic supervisor	Is Primary	Addition Date
Ahmed Ali	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2015
Sara Hamdy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015

**Save**

- Database Tables: All Professor  
left join StudentProgramProfessor  
where the "Select" column = true  
for those in  
StudentProgramProfessor.IsDefen  
seCommittee = false

#### 5.5.2.5.1 Student Program Committee Manage Screen

in Student Program Committee Manage Screen User Can Update the selected one.

AutoComplete Professor list ☐ Is External ☐ Has Sent Individual Report ☐ Has Sent Satisfaction Report

**Add**

Professor	Is External	Has Sent Individual Report	Has Sent Satisfaction Report	Addition Date
Ahmed Ali	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2015
Sara Hamdy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015
Soheir Fahmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015

**Save**

- Database Tables: All Professor  
left join StudentProgramProfessor  
where the "Select" column = true  
for those in  
StudentProgramProfessor.IsDefen  
seCommittee = true





#### 5.5.2.5.2 Students Programs Manage Screen

in Students Programs Manage screen user can view the courses, and add the too, also see the Status, warnings, Supervisors and the given grades for that student and also can update, add or delete them.

A Web Page

http://StudentPrograms/Show

### Courses

Semester	Year	Course Class	Course Code	Course	Specific Course Name	Course Grade	Student Grade	Student Grade Class	Show	Edit	Delete
Spring	2015	SW	123456	Directed readings computer 1	Data Mining	150	111	A	Show	Edit	Delete

- Database Tables: StudentProgramCourse, ProgramClassSemesterCourse

Add

### Status

Status	Date	Show	Edit	Delete
Accepted	Sep. 2010	Show	Edit	Delete
Extension Accepted	Sep. 2015	Show	Edit	Delete

- Database Tables: StudentProgramStatusHistory

Add

### Warnings

Date	Show	Edit	Delete
Sep. 2011	Show	Edit	Delete
Jan. 2011	Show	Edit	Delete

- Database Tables: StudentProgramWarningHistory

Add

### Supervisors

Professor	Is Academic supervisor	Is Primary	Addition Date
Ahmed Ali	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2010
Sara Handy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2010

- Database Tables: StudentProgramProfessor Where IsDefenseCommittee = false

Add/Edit

### Giving Grades

Thesis Title Arabic:  Some text

Thesis Title English:  Some text

Thesis General Field:  Some text

Thesis Specific Field:  Some text

Save

- Database Tables: StudentProgramGivingGrade

### Committee

Professor	Is External	Has Sent Individual Report	Has Sent Satisfaction Report	Addition Date
Ahmed Ali	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2015
Sara Handy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015
Soheir Fahmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015

- Database Tables: StudentProgramProfessor Where IsDefenseCommittee = true

Add/Edit



## 6 Requirements Elicitation

This section contains the data design as given to us by our customer Mr. Mohamed Elsayy.

Many tables exist but the following tables are those highlighted by him to be the important and required tables.

We can conclude from here the data that he needs to be in his software.

Note that:

- **Red** items are arrays or something that contains multiple values. They are described after each table.
- Item followed by “question mark” (?) is of type Boolean (implemented by Mr. Elsayy as checkbox).

These are the important tables that Mr. Elsayy told me about:

**Student** (name, supervisor, madeCommittee?, defenseDone?, madeSeminar?, removeStudent?, removeFromDepartment?, **typeOfStudy**, numberOfComplementaryCourses, numberOfDoneCourses, VTMENAprgram?, doneTOEFEL?, doneQualifyingExam?, dateOfDeptAcceptanceOnQualifyingExam, dateOfExtensionTo5thYear, numberOfWarnings, dateOfFirstWarning, dateOfLastWarning, dateOfRemovalAcceptance, **SupervisorsNames**, dateOfAcceptanceOnSupervision, dateOfAddingSupervisor, dateOfDeptMeeting, **thesisTitle**, convertingFromContinuingStudies?, dateOfConversion, coursesCalculation?, cumGrade, landlineNumber, mobileNumber, email, **coursesTaken**, **complementaryCourses** )

**typeOfStudy** contains: master, continuing study, phd, special programs, complementary courses phase. (Enum)

**SupervisorsNames** contains up to 3 supervisors names. (Array) **thesisTitle** contains the Arabic and English translations. (Array)

**coursesTaken** contains: (name, code, registration date, professor given, grade). This field should be array to contain as many courses as the student take. No upper limit. Each contains the described fields. (Array)

**complementaryCourses** contains: (name, grade) and should be multiple like the last point. (Array)

Some fields also exist in the forms that the student fill. We may add them or not. These are:

(nationality, dateOfBirth, placeOfBirth, address, highestScientificDegree, currentJob, placeOfJob, **allScientificDegree**, pointOfResearch, StudyPlanArabic, StudyPlanEnglish)

**allScientificDegree** may be one of 3 types:





1. B.Sc., the accompanying fields are: (type, university, faculty, department, grad.Year, **allGradesInStudyingYears**)
2. M.Sc., the accompanying fields are: (university, faculty, grad.Year, field of research).
3. Diploma of graduate studies, the accompanying fields are: (faculty, university, grad.Year, GPA).

**Course** (name, **professor**, code, creditHours ,type, **registeredStudents**)

**Professor:** may be array not just one name

**registeredStudents:** Array of students that contains (name, registrationSemesterAndYear, outOfDept?, continuingStudy?, calculated?, masterOrPhd? grade)

**ComplementaryCourse** (studentName, passedCourses?, **complementaryCourses**)

**complementaryCourses** contains Array of (courseName, grade)

**GivingGrade** (studentName, seminarDate, determineThesisTitle?,

finalReportOnGoodnessOfThesis?, refereeMakingSuggestion?,  
 refreeComitteeMade?, messageThatRefereeCommitteelsMade,  
 dateOfDeptAcceptanceOnMakingCommitte,  
 dateOfCollegeAcceptanceOnMakingCommitte,  
 dateOfUniversityAcceptanceOnMakingCommitte,defenseDate,  
 messageOfReservingLectureHall, messageToViceDeanOfDefenseDate, committeeHead,  
 internalExaminer, mainSuervisor, externalExaminer, thesisTitleArabic, thiesisTitleEnglish,  
 thesisGeneralField, thesisSpecificField, **indivdualReport?**, collectiveReport?,  
 thesisAbstract?, satisfactionReport?, dateOfDeptAcceptanceOnGivingGrade )  
**indivdualReport?** Array of Booleans determine if reports are given by the examiners or not.



## 7 Requirements Traceability Matrix

Definition: Traceability is the ability to chronologically interrelate the uniquely identifiable entities in a meaningful way. It refers to the ability to link requirements back to stakeholders' rationales and forward to corresponding design artifacts, code, and test cases. Traceability supports many software engineering activities, such as change impact analysis, compliance verification of code, regression test selection, and requirements validation. [1, p. 52]

### 7.1 Requirements Traceability Matrix (RTM)

The requirements traceability matrix for our project contains the following fields:

- A unique identification number containing the general category of the requirement and a number assigned in ascending order
- The requirement source and statement
- Modification field; when a requirement is changed, eliminated, or replaced, indicate disposition and authority for modification

Within above defined criteria, the RTM description is as follows:

Table 3 Requirements Traceability Matrix

ID	Req.	Source of Req.	Remarks
1	Student registration (with the accompanying fields required to exist).	Mr. Elsayy	Affect design
2	Course registration by some student.	Mr. Elsayy	Affect design
3	Professor registration	None	Required for proper design
4	Complementary courses table.	Mr. Elsayy	Affect design
5	Giving grade table	Mr. Elsayy	Affect design
6	Student course registration conflict (if some student register in the same code twice over years).	Mr. Elsayy	Procedure
7	Print reports for students (courses taken, codes, semesters of courses taken, grades, GPA,...)	Mr. Elsayy	Reporting
8	Print reports for courses (number of students attended, credit hours, professors instructed,...)	Mr. Elsayy	Reporting
9	Print reports for professors (what they did in a semester, hours they gave, name of students under his supervision)	Mr. Elsayy	Reporting



ID	Req.	Source of Req.	Remarks
10	Find if the master program is finished by the student or not (TOEFEL, making committee, finishing seminar,...)	Mr. Elsaywy	Procedure
11	Find number of years of registration of some student	Mr. Elsaywy	Procedure
12	Find students who dropped a term or more without taking any course or doing anything	Mr. Elsaywy	Procedure
13	Find students who have passed two years and didn't get supervision paper.	Mr. Elsaywy	Procedure
14	Having 5 scientific degrees	None	May affect design
15	Having 3 semesters for education	None	May affect design
16	Number of allowed hours to register per semester.	Reg. doc, p.4	
17	Lowest number of students to open course	Reg. doc, p. 4	
18	Rules to add, remove or withdraw from course	Reg. doc, p.4	
19	List of possible grades and what is included in calculating GPA	Reg. doc, p.5	May affect design
20	Registration in the same course twice rules	Reg. doc, p.5	
21	What is added to transcript and what is not	Reg. doc, p.5	Procedure
22	Registration in courses outside the department	Reg. doc, p.5	
23	Rules of student failure and removal from the program	Reg. doc, p.5	
24	Rules of removing course grade	Reg. doc, p.5	
25	Rules of calculating GPA	Reg. doc, p.5	Procedure
26	Rules of accepting and refusing course registration	Reg. doc, p.5	
27	Rules of converting credit hours	Reg. doc, p.10	Procedure
28	Number of hours to complete a program	Reg. doc, p.10,11,13	Procedure
29	Rules of complementary courses	Reg. doc, p.10,12	



ID	Req.	Source of Req.	Remarks
30	Rules of cancelling student registration	Reg. doc, p.11,13	
31	Obligatory and elective courses	Reg. doc, p.54	
32	Courses codes	Reg. doc, p.55,56	May affect design
33	Add or remove Supervisor (should state date and reason of this)	reports from Mr. Elsayy	
34	extend registration to fifth year	reports from Mr. Elsayy	
35	Print mid-year reports. This contains: date, typeOfDegree, student name, dept., date of registration, thesis title in arabic and in english, array of supervisors	reports from Mr. Elsayy	
36	Print goodness of thesis report. This should contain: student name, current job, registration date, thesis title in arabic and english, thesis specific field, thesis abstract.	reports from Mr. Elsayy	



## 8 Process Activity Task Matrix

This section describes the Process Activity Task Matrix (PATM) for the project of the Software Engineering course. This document is extracted from the Requirements Traceability Matrix (RTM)

### 8.1 Tasks

1. Add new student. (GSA\_SI\_M1)
2. Delete Student. (GSA\_SI\_M4)
3. Update student information. (GSA\_SI\_M2)
4. Register a student to a program. (GSA\_R\_M1)
5. Register a student to a course inside the department. (GSA\_R\_M2)
6. Register a student to a course outside the department. (GSA\_R\_M3)
7. Remove a student from course. (GSA\_R\_M4)
8. Withdraw a student from course. (GSA\_R\_M5)
9. Check number of hours which are taken by some student in some semester. It should be more than or equal to 3 and less than or equal to 12 hours. (GSA\_CR\_M1)
10. Add a professor. (SA\_PI\_M1)
11. Add/Update a course. (GSA\_CI\_M1)
12. Delete a course. (GSA\_CI\_M2)
13. Assign list of courses to some semester for some program. (GSA\_CSI\_M1)
14. Remove course from some semester for some program. (GSA\_CSI\_M2)
15. Print the list of courses offered for some program in some semester. (GSA\_CSI\_R1)
16. Update course general information (like course code). (GSA\_CI\_M3)
17. Update course information for some semester (like the professor giving it). (GSA\_CSI\_M3)
18. Add a complementary course. (GSA\_CI\_M4)
19. Print a student (annual) report including courses taken along their grades and current status. (P\_PR\_R1)
20. Detect conflict in course codes. (GSA\_SI\_M3)
21. Update what the student has finished (like finishing TOEFEL, making seminar, making committee, and published a paper). (GSA\_CR\_M2)
22. Check if a student is eligible to finish his study. (GSA\_CR\_M3)
23. Print a course report including number of enrolled students along their grades and the instructor for this semester. (GSA\_CSI\_R2)
24. Print report for professor including courses he has given in some semester. (P\_PR\_R2)
25. Print report for professor including students under his supervision. (P\_PR\_R3)
26. Find number of years for some student since registration. (GSA\_SI\_R1)
27. Find if a student has dropped a semester without taking any courses. (GSA\_CR\_R1)
28. Find if a student has got grade less than C. (GSA\_CR\_R2)
29. Find students who have not get a supervisor for 2 years. (GSA\_CR\_R3)
30. Give grade to a student for some course. (GSA\_CSI\_M4)



31. Calculate GPA for a student. (GSA\_R\_R1)
32. Print a student transcript. (GSA\_SI\_R2)
33. Convert credit hours. (GSA\_R\_M6)
34. Cancel student registration. (GSA\_R\_M7)
35. Add supervisor to a student. (GSA\_R\_M8)
36. Remove supervisor. (GSA\_R\_M9)
37. Extend registration to a fifth year. (GSA\_R\_M10)
38. Print mid-year report. (P\_PR\_R4)
39. Print goodness of thesis report. (P\_PR\_R5)
40. Enter courses and grades metadata. (GSA\_CI\_M5)

## 8.2 Activities

1. Maintain students' information. (GSA\_SI)
2. Registration. (GSA\_R)
3. Complementary rules for students. (GSA\_CR)
4. Maintain professors' information. (SA\_PI)
5. Maintain courses' information. (GSA\_CI)
6. Maintain courses in semester information. (GSA\_CSI)
7. Printing students' reports. (P\_PR)

## 8.3 Processes

1. Staff Affairs (SA)
2. Graduate Students Affairs (GSA)
3. Professors (P)

## 8.4 PATM in tabular format

Table 4 Process Activity Task Matrix

Process	Activity	Task
Staff Affairs (SA)	Maintain professors' information. (SA_PI)	Add a professor. (SA_PI_M1)
Graduate Students Affairs (GSA)	Maintain students' information. (GSA_SI)	Add new student. (GSA_SI_M1)
		Update student information. (GSA_SI_M2)
		Detect conflict in course codes. (GSA_SI_M3)
		Delete Student. (GSA_SI_M4)



Process	Activity	Task
		Find number of years for some student since registration. (GSA_SI_R1)
		Print a student transcript. (GSA_SI_R2)
	Registration. (GSA_R)	Register a student to a program. (GSA_R_M1)
		Register a student to a course inside the department. (GSA_R_M2)
		Register a student to a course outside the department. (GSA_R_M3)
		Remove a student from course. (GSA_R_M4)
		Withdraw a student from course. (GSA_R_M5)
		Calculate GPA for a student. (GSA_R_R1)
		Convert credit hours. (GSA_R_M6)
		Cancel student registration. (GSA_R_M7)
		Add supervisor to a student. (GSA_R_M8)
		Remove supervisor. (GSA_R_M9)
		Extend registration to a fifth year. (GSA_R_M10)
	Complementary rules for students. (GSA_CR)	Check number of hours which are taken by some student in some semester. It should be more than or equal to 3 and less than or equal to 12 hours. (GSA_CR_M1)
		Update what the student has finished (like finishing TOEFEL, making seminar, making committee, and published a paper). (GSA_CR_M2)
		Check if a student is eligible to finish his study. (GSA_CR_M3)
		Find if a student has dropped a semester without taking any courses. (GSA_CR_R1)
		Find if a student has got grade less than C. (GSA_CR_R2)
		Find students who have not get a supervisor for 2 years. (GSA_CR_R3)
	Maintain courses' information. (GSA_CI)	Add a course. (GSA_CI_M1)
		Delete a course. (GSA_CI_M2)
		Update course general information (like course code). (GSA_CI_M3)
		Add a complementary course. (GSA_CI_M4)
		Enter courses and grades metadata. (GSA_CI_M5)
	Maintain courses in semester information. (GSA_CSI)	Assign list of courses to some semester for some program. (GSA_CSI_M1)
		Remove course from some semester for some program. (GSA_CSI_M2)
		Print the list of courses offered for some program in some semester. (GSA_CSI_R1)



Process	Activity	Task
		Update course information for some semester (like the professor giving it). (GSA_CSI_M3)
		Print a course report including number of enrolled students along their grades and the instructor for this semester. (GSA_CSI_R2)
		Give grade to a student for some course. (GSA_CSI_M4)
Professors (P)	Printing students' reports. (P_PR)	Print a student (annual) report including courses taken along their grades and current status. (P_PR_R1)
		Print report for professor including courses he has given in some semester. (P_PR_R2)
		Print report for professor including students under his supervision. (P_PR_R3)
		Print mid-year report. (P_PR_R4)
		Print goodness of thesis report. (P_PR_R5)





## 9 Quality Assurance Plan

### 9.1 Introduction

This Software Quality Assurance Plan (SQAP) sets forth the process, methods, standards, and procedures that will be used to perform the Software Quality Assurance function for the graduate studies automation project. The SQAP follows the SPI modified to accommodate the project model adapted for the project.

#### 9.1.1 Purpose and Scope

This SQAP provides a foundation for managing the graduate studies automation project software quality assurance activities, and is based on project activities and work products as documented in the Project Plan.

This plan:

- Identifies the SQA responsibilities of the project team and the SQA team
- Defines graduate studies automation project reviews and audits and how they will be conducted
- Lists the activities, processes, and work products that the SQA team will review and audit
- Identifies the SQA work products

#### 9.1.2 References

Reference materials used to develop the graduate studies automation project SQAP include Software Process Improvement Guide, Product Suite Handbook V1.2

#### 9.1.3 Assumptions and Limitations

The following is a list of assumptions that will be used during performing this process and any associated constraints:

- The QA department and/or function are established in the organization.
- Adequate fund is provided to execute QA activities to the projects.
- The QA representative is a competent, trained person and has an adequate experience in the project domain.
- Audit checklist templates are properly understood and implemented.

### 9.2 QA Auditing Targets and Standards

#### 9.2.1 Processes to be audited

- Project management process
- Product development process
- Peer review process
- Configuration management



### 9.2.2 Work Products to be audited

- Architecture Design
- Baseline Release Reports
- Change Requests
- Change Requests Log
- Configuration Management Environment
- Configuration Management Plan
- Defects Summary Reports
- Detailed Design
- Functional Configuration Audit Reports
- Issue Tracking Log
- Minutes of Meetings
- Partial Project Management Plan
- Peer Review Checklists
- Peer Review Plan
- Physical Configuration Audit Reports
- Product Development Plan
- Product Release Report
- Project Closure Report
- Project Management Plan
- Project Schedule
- Requirements Elicitation Plan
- Requirements Management DB
- Requirements Traceability Matrix
- Reused Component Specification
- Risk Management Plan
- Risks Management Database
- Software Requirements Specification
- Status Accounting Reports
- System Installation Plan
- Test Cases
- Test Plan
- Test Procedures

### 9.2.3 Project Checkpoints (exit criteria)

Each stage of development will have at least one formal checkpoint called exit criteria. When a stage has been successfully exited, it indicates that all draft deliverables due to date have been completed, all outstanding issues have acceptable action plans, and there is a sound plan for the remainder of the project. The project's designated approvers (signoff authorities) must provide a written position at stage exit. All affected functional areas involved in the project also participate in and can provide input to the stage exit



## 9.2.4 Audit Standards

The graduate studies automation project SQAP will follow the SPI Methodology. All lifecycle work product standards are documented in the Project Plan.

## 9.3 Management and Organizations

### 9.3.1 SQAP Roles and Responsibilities

The following chart defines the SQAP roles and responsibilities of the members of the project team and their function at stage exit. If a particular person is unable to fulfill his/her responsibilities, it will be the responsibility of the manager overseeing that area to ensure a replacement, with appropriate skills and experience, on a timely basis.

Table 5 SQAP Roles and Responsibilities

Role	Name	Org.	SQA Responsibility	Exit criteria Function
QA Manager		Software Development	Manages the Quality Assurance function.	Approve
System Owner		Mr. Mohamed elsayy	Helps define product quality expectations. Represents procurement users. Determines final acceptance of the project	Approve
QA Consultant		Software Development	Audits and approves project deliverables from QA perspective. Reviews plans and deliverables for compliance with applicable standards. Provides guidance and assistance on process matters.	Approve
Project Manager		Software Development	Ensures implementation of quality activities. Coordinates resolution of issues. Provides regular and timely communications	Conduct
Project Manager's manager		Software Development	Monitors implementation of quality activities. Receives reports on SJ-RT quality efforts. Resolves conflict across organizations.	Approve

### 9.3.2 Required Skills

The Quality Assurance consultant must be able to review iterations of the Project Plan and lifecycle work products to determine adherence to industry standards, as modified and



documented in the Project Plan, and provide expert assistance on project management practices and software development process related matters.

This position will work independently from the development team to ensure objective audits of the work products as they are being developed and objective reviews of project management processes and stage exits.

### 9.3.3 Escalation Mechanism

Each deliverable and/or selected work product will be audited to make judgements as to the quality and validity of the deliverable or work product. The assessment will include any verification or validation activities performed since the last In-Stage Assessment. The reviewer will document the results of the assessment using the In-Stage Assessment Report described below. An issue will be logged if there is a problem without a visible plan for resolution. Once a list of issues has been compiled, it will be reviewed with the project manager to see if any new or additional information might mitigate or eliminate any of them. Remaining issues must be addressed with an action plan from the project manager. Issues from an In-Stage Assessment near the end of a stage might become "qualifications" to the current stage exit.

An assessment of risk to the schedule for both the next stage and the remainder of the project will also be provided. Risk categories are as follows

Table 6 Risk categories

Low	Potential or existing problems must be addressed to avoid an impact to the current Project Plan. This would also apply if no issues were identified.
Medium	Problems exist that have a high probability of impacting the current Project Plan or other dependencies
High	Serious problems exist (without an acceptable plan to resolve) that have a high probability of impacting user acceptance, the current Project Plan, or other dependencies

## 9.4 Reporting

### 9.4.1 Verification and Validation of Requirements

Verifying the project requirements at the end of the Preparation stage will establish the proper basis for initiating the Software Design procedures activities. The Functional Requirements Document (FRD) must contain, at a minimum, documentation on the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces



### 9.4.2 Verification

The following activities will be performed as part of requirements verification:

- Produce a traceability matrix tracing all FRD requirements back to system objectives in the project SOW and forward to Software Design elements.
- Evaluate FRD requirements and relationships for correctness, consistency, completeness, accuracy, readability and testability.
- Assess how well the FRD satisfies the SJ-RT system objectives.
- Assess the criticality of requirements to identify key performance or critical areas of software.

### 9.4.3 Validation

The following activities will be performed as part of requirements validation:

- Plan acceptance testing, including criteria for:
  - compliance with all requirements
  - adequacy of user documentation
  - Performance at boundaries and under stress conditions.
- Plan documentation of test tasks and results.
- Execute the Acceptance Test Plan.
- Document acceptance test results.

## 9.5 Approvals

No	QA Representative	QA Manager
1	Name:	Name:
2	Date:	Date:
3	Signature:	Signature:



## 10 QA Work Products Audit Checklists

The Graduate studies automation project's will be audited, taking the following readiness criteria into consideration.

Table 7 QA Work Products Audit Checklists

		Compliance			
REQUIREMENT		Yes	No	N/A	REVIEWER COMMENTS
1.0 GENERAL INFORMATION					
1	<b>Purpose:</b> Describe the purpose of the Quality Assurance Plan.				
1	<b>Scope:</b> Describe the scope of the Quality Assurance Plan as it relates to the project.				
1	<b>System Overview:</b> Provide a brief system overview description as a point of reference for the remainder of the document, including responsible organization, system name or title, system code, system category, operational status, and system environment and special conditions.				
1	<b>Project References:</b> Provide a list of the references that were used in preparation of this document.				
2	<b>Acronyms and Abbreviations:</b> Provide a list of the acronyms and abbreviations used in this document and the meaning of each.				
2	<b>Points of Contact:</b>				
	1.6.1 <b>Information:</b> Provide a list of the points of organizational contact (POCs) that may be needed by the document user for informational and troubleshooting purposes.				
	1.6.2 <b>Coordination:</b> Provide a list of organizations that require coordination between the project and its specific support function (e.g., installation coordination, security, etc.). Include a schedule for coordination activities.				
2.0 SCHEDULE OF TASKS AND RESPONSIBILITIES					



		Compliance			REVIEWER COMMENTS
REQUIREMENT		Yes	No	N/A	
	<p>Based on the tasks described in your Project Plan, provide a schedule of the QA activities for the duration of the project. Tasks may include, but are not limited to, those listed below:</p> <p>Identify Standards and Guidelines</p> <p>Evaluate Software Tools</p> <p>Evaluate Facilities</p> <p>Evaluate Software Products Review Process</p> <p>Evaluate Project Planning, Tracking, and Oversight Processes</p> <p>Evaluate Requirements Analysis Process</p> <p>Evaluate Design Process</p> <p>Evaluate Code and Unit Testing Process</p> <p>Evaluate Integration Testing Process</p> <p>Evaluate Acceptance Testing Process</p> <p>Evaluate Release Process</p> <p>Evaluate the Corrective Action Process</p> <p>Evaluate Media Certification Process</p> <p>Certify Non-deliverable Software</p> <p>Evaluate Storage and Handling Process</p> <p>Evaluate Deviations and Waivers Process</p> <p>Evaluate Configuration Management Process</p> <p>Evaluate Software Development Library Control Process</p> <p>Evaluate Non-developmental Software</p> <p>Perform Configuration Audits</p> <p>Evaluate Risk Management Process</p>				
<b>3.0 SYSTEM DOCUMENTATION</b>					
3	<p><b>Documents by Phase:</b> List and briefly describe the documentation expected to be produced during this project. Depending on the size of your project, this may include, but is not limited to, the documents in the following subsections:</p>				
	<p><b>3.1.1 Initiate Phase:</b></p> <p>Needs Statement</p>				



		Compliance			REVIEWER COMMENTS
REQUIREMENT		Yes	No	N/A	
<input type="checkbox"/>	Project Plan (including WBS) Configuration Management Plan Quality Assurance Plan Feasibility Study Cost/Benefit Analysis Risk Analysis				
<input type="checkbox"/>	<b>3.1.2 Planning Phase:</b> System Support and Acquisition Plan (initial) Functional Requirements Document Data Requirements Document System Security and Privacy Plan Internal Audit Plan Project Plan (updated)				
<input type="checkbox"/>	<b>3.1.3 Design Phase:</b> System/Subsystem Specifications Database Specifications Program Specifications System Support and Acquisition Plan (final) Validation, Verification, and Testing Plan (initial) Training Plan (initial) Project Plan (updated)				
<input type="checkbox"/>	<b>3.1.4 Build Phase:</b> Installation and Conversion Plan (initial) Test Plan User's Manual Operations Manual Maintenance Manual Validation, Verification, and Testing Plan (final) Training Plan (final) Project Plan (updated)				
<input type="checkbox"/>	Test Results and Evaluation Reports Installation and Conversion Plan (final) Project Plan (updated)				
<input type="checkbox"/>	Pilot Test Results				





		Compliance			REVIEWER COMMENTS
REQUIREMENT		Yes	No	N/A	
	Training Material Project Plan (updated)				
<b>4.0 REVIEWS AND AUDITS</b>					
4	<b>Review Process:</b> Define the steps of the review process and the procedures that will be used to conduct reviews.				
4	<b>Formal Reviews and Audits:</b>				
	<p><b>4.2.1 Lifecycle Reviews:</b> Identify the types of SDM associated lifecycle or non-lifecycle reviews that will be conducted within each project lifecycle phase. Include a schedule of reviews and include how the results of each review will be used on the project.</p> <p>Typically, the lifecycle reviews include:</p> <ul style="list-style-type: none"> <li>Requirements Review</li> <li>Design Review</li> <li>Specification Review</li> <li>Preliminary Design Review</li> <li>Critical Design Review</li> <li>Test Readiness Review</li> <li>Formal Qualification Review</li> <li>Production Readiness Review</li> <li>Acceptance Test Review</li> <li>Post-Implementation Review</li> </ul>				
	<b>4.2.2 Audits:</b> Describe the audits that will be conducted on the project and when they will be scheduled.				
4	<b>Informal Reviews:</b> Outline the types of informal reviews that will be conducted.				
4	<b>Review Reports:</b> Identify QA reports that will be produced throughout the project lifecycle and their uses.				



		Compliance			
REQUIREMENT		Yes	No	N/A	REVIEWER COMMENTS
4	<b>Review and Audit Metrics:</b> Describe the metrics that will be used to capture management information, and how they will be analyzed and reported.				
<b>5.0 PROBLEM REPORTING AND CORRECTIVE ACTION</b>					
	Discuss QA responsibilities and activities concerned with the reporting and tracking of project-related problems and resolutions.				
5	<b>Problem/Issue Documentation:</b> Explain how problems and issues will be documented.				
5	<b>Report Metrics:</b> Describe how metrics from the above reports will be collected and analyzed to determine and coordinate corrective actions.				



---

## 11 Test Plan

### 11.1 Introduction

This is the test plan of the post-graduate studies system implemented by students of Software Engineering course to be used in Faculty of Engineering, Alexandria University.  
This document is structured according to the template of SPIG.

The purpose of this document is to outline the test strategy and overall test approach for the project. This includes test methodologies, traceability, resources required, and estimated schedule. We will use the following plan to test our software and make sure it is robust.

The goal is to provide a framework that can be used by managers and testers to plan and execute the necessary tests in a timely and cost-effective manner.

### 11.2 Test objectives

The objective of the test suite is to provide adequate coverage metrics, requirements validation, and system quality data such that sufficient data is provided for those making the decision to release.

#### 11.2.1 Extent of tests

The tests referenced herein are written to validate use cases, requirements (both functional and non-functional), system architecture, and object design. The structured tests for object design will be run first as the components of the system are developed. The structured tests to validate the system architecture will be run next as the system is integrated in bottom-up fashion during integration test.

### 11.3 Test levels

In this section, we define different levels of testing including component test, integration test and system test. In component test, we define each component of the system and list its functionality and the expected behavior. This makes the testing procedure easier and can be traced easily. In integration test, we test the integration between different modules to see if it is working or not. We define all interfaces and the expected data to be transmitted between them to make sure that they are functioning together well. In system test, we test the whole system against the requirements. Our project has some functional and non-functional requirements that it should meet. In system test, we make sure that everything is going well.

#### 11.3.1 Black box tests

Black box tests relating to use cases are developed from the use case diagram(s) in the requirements analysis document. Black box tests derived from functional requirements are developed from the requirements



### 11.3.2 Component test

Structured Component (unit/white box) tests are generated from the Design Document.

### 11.3.3 Integration Test

Integration tests are developed from the System/Architecture Design Document. The integration tests generally come from the overall package diagram describing the architecture of the system. The architecture is also used to help in determining the integration test approach.

### 11.3.4 System Test

We should test here the entire system against the requirements declared in the Requirements Traceability Matrix (RTM). The functional requirements include the ability to register a student in a course with all required fields to do that and printing the reports concerning to the student history from the date of registration and professor's required reports. Non-functional requirements include security, efficiency, validation and verification.

### 11.3.5 Acceptance Test

This test should be done by Mr. Mohamed Elsayy (our customer) to make sure that the delivered system meets his requirements. We can prepare some test cases for him to clarify how our system works and leave him try what he wants.

## 11.4 Features to be Tested/Features Not to be Tested

In this section, we should define the features that we finished and ready for testing and those which are not implemented yet and not ready for testing. At the end of the project, all the required features should be finished and ready for testing. I think this section is concerned more to System and Acceptance test. All features (to be tested or not to be tested) should be extracted from the RTM (Requirements Traceability Matrix).

This section, focusing on the functional aspects of testing, identifies all features and combinations of features to be tested. It also describes all those features that are not to be tested and the reasons for not testing them.

### 11.4.1 Features to be tested

#### 11.4.1.1 Components developed in house

The components developed by this organization will be tested unless otherwise noted

#### 11.4.1.2 Components developed by outsource vendors

Components outsourced to be developed specifically for this project where this test team has primary responsibility to test and validate those components will be tested.

Components outsourced to be developed specifically for this project where the outsourced vendor is responsible for development as well as testing will not be tested as components by



the component testers. Rather, the test results from the vendor will be reviewed by the test leads, and if they pass review, the component will be tested in-house starting with integration test.

#### 11.4.2 Items that will not be tested

##### 11.4.2.1 3rd party and Off-The-Shelf components

It is assumed that 3rd party components were evaluated and the pros and cons properly weighed before choosing that component with our software. The interfaces to those components will be tested, but not the functionality or performance those components. This includes any 3rd party websites, devices or software.

##### 11.4.2.2 Infrastructure components

The actual database software utilized is assumed to work as designed and will not be directly tested for functionality. Performance tests will be done during system test with respect to GUI response time that will involve the database. However, no testing will be done directly against the database.

### 11.5 Approach

This section describes the general approach to the testing process. It discusses the reasons for the selected integration testing strategy. Different strategies are often needed to test different parts of the system.

#### 11.5.1 General Test Strategy

Unit testing and component testing will be performed on the components as they are developed. Test will be executed using test code in the form of either custom test tools or as an automated suite of tests run against the components in their individual sandboxes.

Integrations tests will be performed by both the component testers as well as the system testers. The BAT and the unit test suite will be used as a regression during the integration of components. However, as the integration begins to include GUI level functionality, the tests being run will utilize significantly more manual testing and less automated testing. Manual tests will start by validating functionality based on the requirements

#### 11.5.2 Integration Test Strategy

Because the components will be developed from the bottom-up and top-down, the test strategy will also align to the order of development of components. This will utilize a mostly bottom-up integration test approach, but will also involve the sandwich integration test approach

### 11.6 Pass/Fail Criteria

We should define here when to say that the system passed and when to consider it failed. It passes when it meets the functional and non-functional requirements we determine for the



project. Also, for errors done by users, error message should appear. We consider the system passed in this case and failed if something else happened.

This section specifies generic pass/fail criteria for the tests covered in this plan. They are supplemented by pass/fail criteria in the test design specification. Note that “fail” in the IEEE standard terminology means “successful test” in our terminology.

#### 11.6.1 Component Pass/Fail criteria

Tests executed on components only pass when they satisfy the signatures, constraints, and interfaces dictated by the Object Design Specification for that component. This includes positive tests, negative and stress tests, and boundary tests. If a test exhibits a product failure to meet the objectives of the object design specification, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team.

#### 11.6.2 Integration Pass/Fail criteria

Tests executed on integrated components only pass when they satisfy the signatures, constraints, and interfaces dictated by both the object design specification and the system architecture specification. This includes positive tests, negative and stress tests, boundary conditions, and tests that explicitly manipulate the interface environment (such as the physical connection to the database server).

If a test exhibits a product failure to meet the objectives of both the object design specification and the system architecture specification, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team.

#### 11.6.3 System Pass/Fail criteria

Tests executed against the system use the functional requirements, non-functional requirements, and use cases as the oracle to determine pass or fail.

If a test exhibits a product failure to meet the objectives of any of the functional requirements, non-functional requirements, or the use cases, it will fail and a defect/issue will be reported in the defect tracking system for review by the triage team.

### 11.7 Suspension/Resuming Criteria

The system should have a consistent database regardless of the point of suspension of the system. We should start from a stable and consistent point when resuming the system. We should make sure that the system will not crash if suspension is happened at any point and we can set a main page to be the starting point after resuming the system.

This section specifies the criteria for suspending the testing on the test items associated with the plan. It also specifies the test activities that must be repeated when testing is resumed.



### 11.7.1 Automated Unit Test Suite

As components are being developed, unit tests will be developed to test the interfaces of the components and low-level unit tests will be developed to test the methods of the underlying classes in the components.

As a prerequisite to the BAT, the automated unit test suite will be run by the build server on a per-build basis.

When the unit-test suite reports failures, testing will not occur on that build until the failures have been analyzed and resolved. Testing will resume on a build that passes the automated unit test suite.

### 11.7.2 Build Acceptance Test (BAT)

When a build is deemed ready to test by development, a build acceptance test will be run on the build. The BAT will consist of a broad but shallow set of tests to determine the overall stability of the build and decide if it is worth testing.

If the BAT fails on a particular build, testing will suspend until another build is created with any BAT failure issues fixed, verified by running the BAT again. Testing will resume on a build that passes the BAT.

Different build acceptance tests will be developed and used for the different test phases. Component BATs will be small and localized for each of the components. Integration BATs will vary based on the level of integration testing being performed. The System Test BAT will contain a set of tests that will utilize each of the components of the system.

### 11.7.3 Regression Testing

On a build by build basis, major bug fixes or code changes will be reviewed to determine the effects they may have on the system. If the changes are deemed to cause a sufficient amount of risk, regression test sets of the appropriately judged size will be created and executed.

A system-wide regression will also be run on the release candidate build to ensure incremental changes to the system have not altered the results of the tests that were run early in the test cycle.

### 11.7.4 System Design Changes

If at any point in time issues are submitted that require a design change to the system, all testing will be suspended. After the changes to the requirements, system architecture, and object design are made, a review and updates will be performed of the test specifications to ensure they properly align with the revised system changes. After updates are made, testing will resume. Tests in the vicinity of the change must all be rerun. A 20% regression of other tests must also be performed to ensure the changes did not adversely affect other parts of the system.



## 11.8 Test Deliverables

For each test done, the following should be reported:

1. Type of test.
2. Brief description.
3. Output of test (pass/fail).
4. Issues/bugs if exist.

## 11.9 Testing Tasks

I do not understand actually what that is but it may be the tasks a tester should do. The tester should do all the test cases that cover all types of tests declared in this document and deliver the test deliverables clarified in Section 8 in this document.

## 11.10 Test Environment

System test should be done using the PC of Mr. Mohamed Elsayy to make sure that all technologies and software we used are compatible with the environment from which he will work. So, we should try to work from the same environment from the beginning of the project.

## 11.11 Staffing and Training Needs

Although not applicable for our project. A standard testing team should comprise of the below

Test Manager

- responsible for finding and training the following test resources

Test Leads

- must be trained on the process being used for this project
- must be trained on the test specification format utilized
- must be trained on the defect/issue tracking system utilized

Component Testers

- must know SQL and Java
- must be familiar with NetBeans and Eclipse
- must be skilled at unit testing, API testing, and integration testing
- must be trained on the process being used for this project, the test specification format utilized, and the defect/issue tracking system utilized

System Testers

- must know how to use Load testing & Stress testing automation tools
- must be experienced in system testing and use case validation testing
- must be trained on the process being used for this project.
- must be trained on the test specification format utilized
- must be trained on the defect/issue tracking system utilized

## 11.12 Risks and Contingences

This task is related to the risk management. I think it can be also related to the component and the integration test. Depending modules are affected by wrong behaviors from other modules. So, we can determine here the issues that we should see in many modules depending on the wrong behavior from one module.





### 11.13 Approvals

We should have an approval on this document from the test manager and the project manager.

No	Test Representative	Test Manager
1	Name:	Name:
2	Date:	Date:
3	Signature:	Signature:



## 12 Test Procedures

### 12.1 Adding a student to some program

#### 12.1.1 Purpose

This test tests the steps should be taken to add a student.

#### 12.1.2 Test Procedure Identifier

GSA\_SI\_M1\_t1

#### 12.1.3 Input Specifications

Inputs will be entered through the web interface. Text fields are filled with appropriate values.

#### 12.1.4 Output Specifications

This test should NOT produce any errors and added student should appear in the database.

#### 12.1.5 Procedure Steps

1. Open the web interface.
2. Choose Student Registration tab.
3. Fill the text fields as follows:
  - a. Name: any valid name
  - b. National ID: any number of 14 character
  - c. Phone number: any number of 10 characters (starts with 203)
  - d. Mobile number: any number of 12 characters (starts with 201)
  - e. Program: Choose M.Sc.
  - f. Previous scientific degree: B.Sc. Alexandria University, Faculty of Engineering.
4. Go to registered students tab and list all registered students.
5. You should find the registered student you have added along with the information you have entered.

### 12.2 Testing the eligibility of some student to finish his study

#### 12.2.1 Purpose

This test tests the eligibility of student to finish his M.Sc.

#### 12.2.2 Test Procedure Identifier

GSA\_CR\_M3\_t1

#### 12.2.3 Input Specifications

No input for this example. Just check from the web interface the status of the student.

#### 12.2.4 Output Specifications

Student should be ineligible to finish his M.Sc.



### 12.2.5 Procedure Steps

1. Open the web interface.
2. Choose Students tab.
3. Choose the student that entered the system in example 1.
4. Press the button of check the eligibility of this student.
5. A message should appear telling that this student is not eligible to finish his studies now. It will be perfect if this message states why he is not eligible to finish his M.Sc. studies.

## 12.3 Updating student basic information

### 12.3.1 Purpose

This test checks the ability to update some information of some student after adding him/her.

### 12.3.2 Test Procedure Identifier

GSA\_SI\_M2\_t1

### 12.3.3 Input Specifications

Inputs will be given to the test through filling fields of the web interface views as discussed in the procedure steps.

### 12.3.4 Output Specifications

At the end, the information about that student should be updated according to the desired values.

### 12.3.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button "edit". You will be directed to the student basic information page. You can change one or more fields of these ones:
  - a. Phone number: 03xxxxxxx
  - b. Mobile number: 01xxxxxxxxxx
  - c. Email: [example@example.com](mailto:example@example.com)
4. Press "Save". You will be directed back to the students table. Find the changes you have done.

## 12.4 Updating student past grades information

### 12.4.1 Purpose

This test checks the ability to update some information of some student after adding him/her.



## 12.4.2 Test Procedure Identifier

GSA\_SI\_M2\_t2

## 12.4.3 Input Specifications

Inputs will be given to the test through filling fields of the web interface views as discussed in the procedure steps.

## 12.4.4 Output Specifications

At the end, the information about that student should be updated according to the desired values.

## 12.4.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button “edit”. You will be directed to the student basic information page.
4. In the scientific degree’s table, press “Add”. You will be directed to another page to add a scientific degree. Also, for this test, you can press the “edit” button for an already added degree.
5. In the scientific degree page, you can update one or more of the following:
  - a. Scientific degree class: B.Sc.
  - b. Faculty: Faculty of Engineering
  - c. University: Alexandria University
  - d. Department: Computer and Systems Engineering Department.
  - e. Graduation year: 2014.
  - f. Field of research (may be left empty): Computer vision.
  - g. GPA: 3.2
6. After finishing these values, click “Add”. It should be added to the list of degrees at the bottom of this page.

## 12.5 Updating student program information

### 12.5.1 Purpose

This test checks the ability to update some information of some student after adding him/her.

### 12.5.2 Test Procedure Identifier

GSA\_CR\_M2\_t1

### 12.5.3 Input Specifications

Inputs will be given to the test through filling fields of the web interface views as discussed in the procedure steps.



### 12.5.4 Output Specifications

At the end, the information about that student should be updated according to the desired values.

### 12.5.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button "edit". You will be directed to the student basic information page.
4. At the bottom of this page, find "Programs" tab.
5. Change any of the fields in this tab like:
  - a. Finished Toefel.
  - b. Has research point.
  - c. VTMENA program.
  - d. Published a paper.
6. Click save will direct you to the original page of student information. Find the student attributes updated.

## 12.6 Register a student to a program

### 12.6.1 Purpose

This test checks the ability to register a student to some program

### 12.6.2 Test Procedure Identifier

GSA\_R\_M1\_t1

### 12.6.3 Input Specifications

Inputs will be given to the test through filling fields of the web interface views as discussed in the procedure steps.

### 12.6.4 Output Specifications

At the end, the information about that student should be updated according to the desired values.

### 12.6.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button "edit". You will be directed to the student basic information page.
4. At the bottom of this page, find "Programs" tab.
5. Click the "Add" button. You will be directed to another page.



6. Choose the type of the program (let it be “Master”) and set its starting date (by default it will be the date of the day you are doing this task).
7. Click “save” then you can find the new information added to the “Programs” tab for this student.

## 12.7 Get number of years since student registration in some program

### 12.7.1 Purpose

This test checks the number of years of registration of some student in some program. This can be known through getting the start date of registration of some student in some program. The employee can then take some decision based on this information.

### 12.7.2 Test Procedure Identifier

GSA\_SI\_R1\_t1

### 12.7.3 Input Specifications

No inputs using keyboard. Such some steps to get the required information.

### 12.7.4 Output Specifications

The starting date of student registration in some program. It should be correct.

### 12.7.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button “show”. You will be directed to the student basic information page.
4. At the bottom of this page, find “Programs” tab. In this tab, you can find all the programs that this student is registered in.
5. Find the start date of each program recorded in this tab.

## 12.8 Deleting a student from the whole system

### 12.8.1 Purpose

This test checks the ability to delete some student. This may not be required but it is supported feature in our software.

### 12.8.2 Test Procedure Identifier

GSA\_SI\_M4\_t1



### 12.8.3 Input Specifications

No inputs using keyboard. Such some steps to get the required information.

### 12.8.4 Output Specifications

The student should be removed from the system by the end of this test.

### 12.8.5 Procedure Steps

1. Navigate to the students tab.
2. Search on some student using name or **preferably** ID.
3. For the required student (any one), press the button “delete”.
4. The student should be deleted from the whole system and cannot be found again if searched on.

## 12.9 Maintaining Courses Category

### 12.9.1 Purpose

This test checks the ability to add, edit and delete a course category. Current categories of our department are: Software, Hardware, Control and Application.

### 12.9.2 Test Procedure Identifier

GSA\_CI\_M5\_t1

### 12.9.3 Input Specifications

Course category name and description. They will be given through the web interface provided by us.

### 12.9.4 Output Specifications

The added, removed or edited category should appear in the table of categories.

### 12.9.5 Procedure Steps

1. Navigate to the course categories page.
2. Find the “Add” button and click on it. You will be directed to another screen to add the new course category.
3. Write the name (let it be “SW”) and the description (let it be “Software”) of the new category.
4. Click “save” then you will be directed back to the table. Find the course category you have added in the table.
5. To edit it, click on “edit” button beside it. You will be directed again to the screen of editing this category.
6. Edit the name to “S-Ware” for example and click save. You will be directed again to the categories table. Find your data updated.



7. To delete it, click on “delete” button beside it. Your entry will be deleted from this table.

## 12.10 Maintaining Courses Grades Categories

### 12.10.1 Purpose

This test checks the ability to add, edit and delete a course grade category. Current categories of our department are: A, A-, B+, B, B-, C, C+, C-, F, FW, I, L and others.

### 12.10.2 Test Procedure Identifier

GSA\_CI\_M5\_t2

### 12.10.3 Input Specifications

Course grade category name and description. They will be given through the web interface provided by us.

### 12.10.4 Output Specifications

The added, removed or edited category should appear in the table of categories.

### 12.10.5 Procedure Steps

1. Navigate to the course grades categories page.
2. Find the “Add” button and click on it. You will be directed to another screen to add the new grade category.
3. Write the name (let it be “I”) and the description (let it be “Incomplete”) of the new category.
4. Click “save” then you will be directed back to the table. Find the grade category you have added in the table.
5. To edit it, click on “edit” button beside it. You will be directed again to the screen of editing this category.
6. Edit the name to “Inc.” for example and click save. You will be directed again to the categories table. Find your data updated.
7. To delete it, click on “delete” button beside it. Your entry will be deleted from this table.

## 12.11 Maintaining Professors Titles Categories

### 12.11.1 Purpose

This test checks the ability to add, edit and delete a professor title category. Current categories of professors are: Teaching Assistant, Doctor, Prof. Dr., Associate Prof. and others.





---

### 12.11.2 Test Procedure Identifier

GSA\_CI\_M5\_t3

### 12.11.3 Input Specifications

Professor title category name. It will be given through the web interface provided by us.

### 12.11.4 Output Specifications

The added, removed or edited category should appear in the table of categories.

### 12.11.5 Procedure Steps

1. Navigate to the professors' titles categories page.
2. Find the "Add" button and click on it. You will be directed to another screen to add the new professor title category.
3. Write the new professor title (let it be "Prof.").
4. Click "save" then you will be directed back to the table. Find the professor title category you have added in the table.
5. To edit it, click on "edit" button beside it. You will be directed again to the screen of editing this category.
6. Edit the name to "Professor" for example and click save. You will be directed again to the categories table. Find your data updated.
7. To delete it, click on "delete" button beside it. Your entry will be deleted from this table.

## 12.12 Adding/Updating a course

### 12.12.1 Purpose

This test checks the ability to add or update a course to the system. This is used for any program and even in continuing courses.

### 12.12.2 Test Procedure Identifier

GSA\_CI\_M1\_t1

### 12.12.3 Input Specifications

Course attributes like name, credit hours, total grade, etc. They will be given through the web interface provided by us.

### 12.12.4 Output Specifications

The added course along its attributes should appear in the table of courses.



### 12.12.5 Procedure Steps

1. Navigate to the courses page.
2. Find the "Add" button and click on it. You will be directed to another screen to add the new course.
3. In this page, you can enter the following information about the course:
  - a. Course Name: Software Engineering
  - b. Course Class: choose "SW" from the drop down menu. This menu contains the categories that you have already added in the courses categories before.
  - c. Course code: 147892
  - d. Credit Hours: 3
  - e. Total Grade: 100
  - f. Min number of Students: 5
4. Click "save" then you will be directed back to the courses tables. Find your course added to the table.
5. To update any of the fields of the added course, click on "edit" button beside it and update the required field (let it be the course code and update it to 586721) then click "save".
6. You can find the course with the updated attributes in the table.

## 12.13 Adding a course with errors

### 12.13.1 Purpose

This test checks the ability to catch errors when adding a new course to the system.

### 12.13.2 Test Procedure Identifier

GSA\_CI\_M1\_t2

### 12.13.3 Input Specifications

Course attributes like name, credit hours, total grade, etc. They will be given through the web interface provided by us.

### 12.13.4 Output Specifications

Any erroneous field should be caught by the system and a detailed error message should appear.

### 12.13.5 Procedure Steps

1. Navigate to the courses page.
2. Find the "Add" button and click on it. You will be directed to another screen to add the new course.
3. In this page, you can enter the following information about the course. Those in red are erroneous, you can use one or more of them to test error checking:



- a. Course Name: 123 (or any number)
  - b. Course Class: choose "SW" from the drop down menu.
  - c. Course code: Software (or any text)
  - d. Credit Hours: any text
  - e. Total Grade: any text
  - f. Min number of Students: any text
4. Click "save" then an error message should appear to show the errors one have done in the data entry.

## 12.14 Adding a complementary course

### 12.14.1 Purpose

This test checks the ability to add a complementary course to the system. This is used for prerequisites before some programs.

### 12.14.2 Procedure Identifier

GSA\_CI\_M4\_t1

### 12.14.3 Input Specifications

Course attributes like name, credit hours, total grade, etc. They will be given through the web interface provided by us.

### 12.14.4 Output Specifications

The added course along its attributes should appear in the table of courses.

### 12.14.5 Procedure Steps

1. Navigate to the courses page.
2. Find the "Add" button and click on it. You will be directed to another screen to add the new course.
3. In this page, you can enter the following information about the course:
  - a. Course Name: Data Mining
  - b. Credit Hours: 3
  - c. Total Grade: 150
  - d. Check the complementary course checkbox to mark this course as complementary one.
4. Click "save" then you will be directed back to the courses tables. Find your course added to the table.



## 12.15 Deleting a course

### 12.15.1 Purpose

This test checks the ability to delete a course to the system. This is used for both default and complementary courses.

### 12.15.2 Test Procedure Identifier

GSA\_CI\_M2\_t1

### 12.15.3 Input Specifications

Some clicks on specific buttons to delete the course. Just through the web interface we provide.

### 12.15.4 Output Specifications

The deleted course should not appear any more in the table of courses.

### 12.15.5 Procedure Steps

1. Navigate to the courses page.
2. Find the course you would like to delete and click on “delete” button beside it.
3. The page will refresh automatically and the course should not appear any more.

## 12.16 Adding course to some program in some semester

### 12.16.1 Purpose

Test the ability to add a program class for some semester.

### 12.16.2 Test Procedure Identifier

GSA\_CSI\_M1\_t1

### 12.16.3 Input Specifications

Inputs will be given through the web interface including the program, semester, course code, year and other parameters.

### 12.16.4 Output Specifications

The input data should be shown at the end in the table of program classes.

### 12.16.5 Procedure Steps

1. Navigate to the program class tab. At the first, it should be empty.
2. Find the button “Add” and click on it. It will take you to another window to add course for a program class.
3. In the program courses window, enter appropriate values for the required parameters as follows:
  - a. Choose “Master” for “Program” field.



- b. Choose any semester from the drop down menu (let it be "Fall" for example). The year should be by default the current year. You can change it to 2016 for example.
  - c. Choose the course class from the drop down menu. It should have all of the courses you have added before in the courses tab. Choose "SWE" for example.
  - d. Choose the general name of the course (like "Directed reading in Computer Science").
  - e. Write down the Specific course name (You can leave this empty without getting errors too).
  - f. Give a valid course code in the appropriate field (let it be 7035614). This field should have default value. Make sure of that.
  - g. Assign minimum number of students to 6.
  - h. Assign credit hours to 3. This field should have a default value. Make sure of that.
  - i. Assign total grade to 100. This field should have a default value. Make sure of that.
  - j. Don't mark the directed reading for this test.
  - k. Choose any 2 professors for this class. The list should contain all of the professors you have added before. Make sure of that.
  - l. Click "Add course". You should be directed to the table of program class.
  - m. Make sure that the course you have added to this semester is recorded.
4. Press the "Save" button. You should be directed to a page that contains the courses you add. Find your course along with all of its attributes added.

## 12.17 Getting courses of some program in some semester

### 12.17.1 Purpose

Test the functionality of searching for all courses provided by some program in some semester.

### 12.17.2 Test Procedure Identifier

GSA\_CSI\_R1\_t1

### 12.17.3 Input Specifications

Inputs will be given through the web interface.

### 12.17.4 Output Specifications

List of courses of some program that are offered in some semester.

### 12.17.5 Procedure Steps

1. Navigate to the program class tab. At the first, it should be empty.



2. For some saved program (let it be "Masters") click "Show." It will take you to another window to search for courses in this program.
3. In the semester field, choose one from the drop down menu (let it be "Fall") and in year write "2015". It should have default value of current year.
4. Press the button "Search". You should see all courses that you have added for this program and semester along with all accompanying attributes.

## 12.18 Delete some course in some program and some semester

### 12.18.1 Purpose

Test the functionality of deleting a course provided by some program in some semester.

### 12.18.2 Test Procedure Identifier

GSA\_CI\_M2\_t1

### 12.18.3 Input Specifications

Inputs will be some commands through the web interface.

### 12.18.4 Output Specifications

List of courses of some program that are offered in some semester.

### 12.18.5 Procedure Steps

1. Navigate to the program class tab. At the first, it should be empty.
2. For some saved program (let it be "Masters") click "Show." It will take you to another window to search for courses in this program.
3. In the semester field, choose one from the drop down menu (let it be "Fall") and in year write "2015". It should have default value of current year.
4. Press the button "Search". You should see all courses that you have added for this program and semester along with all accompanying attributes.
5. For some course, press "delete". It should be removed from the list after searching again.

## 12.19 Update some course in some program and some semester

### 12.19.1 Purpose

Test the functionality of updating a course provided by some program in some semester.

### 12.19.2 Test Procedure Identifier

GSA\_CSI\_M3\_t1



### 12.19.3 Input Specifications

Inputs will be some commands through the web interface.

### 12.19.4 Output Specifications

Attributes of the course should be changed in the courses list.

### 12.19.5 Procedure Steps

1. Navigate to the program class tab. At the first, it should be empty.
2. For some saved program (let it be "Masters") click "Show." It will take you to another window to search for courses in this program.
3. In the semester field, choose one from the drop down menu (let it be "Fall") and in year write "2015". It should have default value of current year.
4. Press the button "Search". You should see all courses that you have added for this program and semester along with all accompanying attributes.
5. For some course, press "edit". It should take you to another screen that contains the attribute of this course in this semester for the chosen program.
6. Update some of the listed attributes like:
  - a. Professors giving this course: choose one more.
  - b. Course code: change the last number.
  - c. Check/uncheck the directed reading field.
7. Finally, press "Save". You should be directed again to the courses window. Find your updated course attribute.

## 12.20 Adding course to some program in some semester with some errors

### 12.20.1 Purpose

Test the ability to check erroneous inputs and catching it in the process of adding a program class for some semester.

### 12.20.2 Test Procedure Identifier

GSA\_CSI\_M1\_t3

### 12.20.3 Input Specifications

Inputs will be given through the web interface including the program, semester, course code, year and other parameters.

### 12.20.4 Output Specifications

Declarative error message should appear to the user letting him to correct his error.

### 12.20.5 Procedure Steps

1. Navigate to the program class tab. At the first, it should be empty.



2. Find the button “Add” and click on it. It will take you to another window to add course for a program class.
3. In the program courses window, enter appropriate values for the required parameters. All red steps are errors. You can try one of them or multiple ones to check the error generation:
  - a. Choose “Master” for “Program” field.
  - b. Choose any semester from the drop down menu (let it be “Fall” for example).
  - c. In the year field, write “abc” or any other text (not a number) to generate an error.
  - d. Choose the course class from the drop down menu. It should have all of the courses you have added before in the courses tab. Choose “SWE” for example.
  - e. Choose the general name of the course (like “Directed reading in Computer Science”).
  - f. Write down the Specific course name as numbers. This should be invalid.
  - g. Give an invalid course code in the appropriate field (let it be INVALID).
  - h. Assign minimum number of students to any text like INVALID.
  - i. Assign credit hours to any text like INVALID.
  - j. Assign total grade to any text like INVALID.
  - k. Don’t mark the directed reading for this test.
  - l. Don’t choose any professor for this course.
4. Click “Add course”. A declarative error message should appear that describes the exact error that happened in any case.
5. The user should be directed again to the same page to remove the errors from attributes entries.





## 13 Configuration Management Plan

### 13.1 Introduction

In order to effectively manage the project, a coordinated Configuration Management (CM) Plan is needed. This plan will establish CM roles and responsibilities and describe how the project team will track, implement, and communicate configuration items (CIs) and changes throughout the project lifecycle.

### 13.2 Roles and Responsibilities

The following roles and responsibilities pertain to the CM Plan for the project.

#### 13.2.1 Configuration Control Board (CCB)

The CCB is comprised of the project Sponsor, Project Manager, Configuration Manager, and the Lead Engineer for the configuration item (CI) under consideration. The CCB is responsible for the following:

- Review and approve/reject configuration change requests
- Ensure all approved changes are added to the configuration management database (CMDB)
- Seeking clarification on any CIs as required

#### 13.2.2 Project Sponsor

The Project Sponsor is responsible for:

- Chairing all CCB meetings
- Providing approval for any issues requiring additional scope, time, or cost

#### 13.2.3 Project Manager

The Project Manager is responsible for:

- Overall responsibility for all CM activities related to the project
- Identification of CIs
- All communication of CM activities to project stakeholders
- Participation in CCB meetings
- Re-baselining, if necessary, any items affected by CM changes

#### 13.2.4 Configuration Manager

The Configuration Manager will be appointed by the Program Management Office (PMO).

The Configuration Manager is responsible for:

- Overall management of the CMDB



- Identification of CIs
- Providing configuration standards and templates to the project team
- Providing any required configuration training

### 13.2.5 Lead Engineers

All identified CIs will be assigned to a Lead Engineer. The assigned Lead Engineer is responsible for:

- Designating a focus group to develop the change request
- Ensure all change requests comply with organizational templates and standards prior to the CCB
- Identification of CIs

### 13.2.6 Engineers

Each CI will be assigned to a focus group consisting of several engineers. Each member of the focus group will provide input to the change request prior to submitting the change request to the lead engineer for review and presentation at the CCB

## 13.3 Configuration Control

The project will use a standardized configuration control process throughout the project lifecycle in order to ensure all CIs are handled in a consistent manner and any approved changes are fully vetted regarding impact and communicated to stakeholders.

As CIs are identified by the project team, the Configuration Manager will assign a CI name and the CI will be entered into the CMDB in an “initiate” status. The CI will then be assigned to an engineer focus group. Each member of a CIs focus group will have the ability to access the CI through the CMDB, make changes and edits, and enter the CI back into the CMDB with a description of the change/edit annotated in the CMDB log.

It is imperative that for any software changes testing is conducted by the focus group in order to validate any changes made. The Lead Engineer assigned to manage the focus group is responsible for ensuring that testing has been conducted, changes are entered into the CMDB log, and that all changes/edits are saved properly into the CMDB. The Lead Engineer is also responsible for assigning new version numbers and CMDB status for any changes made by his/her assigned focus group.

Many times a CI will have a relationship with one or more other CIs within a project. The Lead Engineer, CM, and Project Manager will work together to ensure these relationships are fully understood. The Lead Engineer and CM will then be responsible for illustrating these relationships and co-dependencies in the CMDB to ensure a full understanding of each CI and how they relate to one another.



Any configuration changes which are identified by the project team or stakeholders must be captured in a configuration change request (CCR) and submitted to the CCB. The CCB will review, analyze, and approve/deny the request based on the impact, scope, time, and cost of the proposed change. If the change is approved, the project requirements will be re-baselined (if necessary) and all changes will be communicated to the project team and stakeholders by the Project Manager. Denied CCRs may be re-submitted with additional or new information for re-consideration by the CCB.

### 13.4 Configuration Management Database (CMDB)

The CMDB will be the centralized repository for all configuration information for the project. The CMDB provides a common platform for the project team to edit, change, revise, and review CIs and also to ensure all documents and data are updated with the latest revision and release formats.

Access to the CMDB will be granted and governed by standard UNIX permissions. Two types of CMDB access will be granted for the project:

- 1) Full read and write access will be granted to the CM, Project Manager, Lead Engineers, and Engineers. These individuals will be authorized to access the CMDB to make changes, edit documents and data, and review and approve versions and CI status.
- 2) Read only access will be granted to the Project Sponsor and all other stakeholders. This access will allow these individuals to view all CIs and CI data but they will not be authorized to make any changes. If these individuals identify the need for a change or edit they will notify the CM who will review the notification and provide feedback.

The CMDB will provide assurance that members of the project team are always working off of the latest version of software, data, and documentation. However, it is important to maintain the history of these assets throughout the project lifecycle. As these assets are changed and updated, the Lead Engineer of the CI's assigned focus group will be responsible for updating the status of the CI and providing new revision numbering. This numbering will be done in accordance with Smith Company's standard revision control numbering process wherein higher version numbers indicate more recent versions of the software, data, or documentation.



## 13.5 Configuration Status Accounting

It is important that for the Project, the Project Sponsor and Vice President of Technology have the ability to review configuration status at any given time. The Project Manager will also submit weekly reports, to include configuration status, every Friday. These reports will consist of the following information as part of the configuration status section:

- 1) Change requests
  - a. Aging - How long change requests have been open
  - b. Distribution – number of change requests submitted by owner/group
  - c. Trending – what area(s) are approved changes occurring in
- 2) Version Control
  - a. Software
  - b. Hardware
  - c. Data
  - d. Documentation
- 3) Build Reporting
  - a. Files
  - b. CI relationships
  - c. Incorporated Changes
- 4) Audits
  - a. Physical Configuration
  - b. Functional Configuration

Prior to any new software releases, the CM will work with each Lead Engineer to ensure all CIs are updated with latest release versions.

## 13.6 Configuration Audits

Configuration audits will be an ongoing part of the project lifecycle. The purpose of the configuration audit is to ensure all team members are following the established procedures and processes for configuration management. Project audits for the Project will occur prior to any major software release or at the Project Manager or Sponsor's discretion if they determine the need for one.

All project configuration audits will be performed by the CM. Throughout the project the CM works closely with Lead Engineers to ensure that all configuration processes and procedures are being followed. As part of the configuration audit the CM will perform the following tasks:

- 1) Establish an audit environment in the CMDB
- 2) The CM will copy all of the latest software, data, and document versions into the audit environment
- 3) The CM will ensure all versions are correctly numbered and that version control has been performed properly



- 4) The CM will analyze historical versions and timestamps of all software, data, and documents to ensure all changes/edits were properly recorded and captured
- 5) The CM will copy latest software versions and conduct software testing to ensure requirements are being met
- 6) The CM will ensure all required artifacts are present and current in the CMDB
- 7) The CM will ensure all approved CCRs have been incorporated into the project and are recorded in the CMDB

Once the audit has been performed, the CM will compile his/her audit findings. For each finding, the CM must work with the Project Manager/Team to identify the corrective action(s) necessary to resolve the discrepancy and assign responsibility for each corrective action.

Upon completion of the project audit and findings, the CM will note all discrepancies and compile a report to be presented to the Project Manager, Sponsor, and VP of Technology.



## 14 Configuration Management Environment

**Project** Graduate studies **Date**  
**Title:** automation System **Prepared:** 27 Nov. 2015

Version Management:

<https://github.com/> will be utilized to maintain all CIs of the project.

Each team member has roles and responsibilities during the project life cycle regarding the configuration management.

Configuration Change Control:

<https://www.bugzilla.org/> will be used for reporting bugs and new requirements

Change request reports could be used

Table 8 Configuration Management Roles and Responsibilities

Role	Responsibility
Program Manager/ Project Management <i>Islam Moursy</i>	<ul style="list-style-type: none"> <li>Develops and maintains artifacts following proper version control procedures using the SCM Procedures and work instructions for each VA Product being worked as part of the Program/Project.</li> <li>Ensures proper execution of the SCM Plan Standard.</li> <li>Oversees the SCM process.</li> <li>Assesses and evaluates all other change requests.</li> <li>Establish appropriate Change Control Board (CCB).</li> <li>Submit CCB baseline information.</li> <li>Identify dependent projects.</li> <li>Establish/revise required artifacts.</li> <li>Creation of SCM Procedures and work instructions for each VA product they are assigned.</li> </ul>
Software Configuration Manager <i>Samar Abozeed</i>	<ul style="list-style-type: none"> <li>Educates project team members in SCM "best practices."</li> </ul>



	<ul style="list-style-type: none"> <li>• Develops and maintains SCM Procedures and work instructions for each VA product they are assigned.</li> <li>• Establishes, promotes, and releases baselines.</li> <li>• Works with the Enterprise Tools Team to create and administer the SCM repositories.</li> <li>• Works with the Enterprise Tools Team to build and manage the SCM tool environment.</li> <li>• Performs or validates interim and final builds.</li> <li>• Prepares release package, release archives and Version Description Documents (VDD).</li> <li>• Accountable for instituting the established processes and reporting progress statistics based on change requests.</li> <li>• Attends CCB meetings.</li> <li>• Identifies product baselines as necessary of all products within their assigned Projects.</li> <li>• Responsible for SCM audits and necessary status accounting related to the product.</li> <li>• Conducts audits at scheduled milestones.</li> </ul>
Development Manager/Leads <i>Ahmed Badie</i>	<ul style="list-style-type: none"> <li>• Develops and maintains artifacts following proper version control procedures using the SCM Procedures and work instructions.</li> <li>• Submits build/release requests.</li> <li>• Coordinates development activities and assigns tasks.</li> <li>• Ensures all SCM Procedures and work instructions are implemented and followed for all software, documentation, and/or any other components for which they are responsible.</li> <li>• Ensures all developers' work within the specified SCM process and related guidelines as specified in the SCM Procedures and work instructions.</li> <li>• Attends the CCB meetings and provide technical details, as required.</li> </ul>
Developers/System Administration/Functional/ Technical Analysts/DBAs/System Administration	<ul style="list-style-type: none"> <li>• Develops and maintains artifacts following proper version control procedures using the SCM Procedures and work instructions.</li> </ul>



*Arsany Guirguis*

- Maintain accurate, detailed information for all assigned change requests (CRs), in the CR database, related to the applicable development detail of the CRs lifecycle.
- Provide impact analysis reporting for the CCB approved problems or changes, including documentation of suggested solutions to facilitate CCB disposition activities.
- Documentation of build, release, and installation instructions.

### 14.1 Configuration management environment:

This section illustrates the configuration management environment for the project

Github repo https url: <https://github.com/mmadian/SWProject.git>

Github repo ssh command: <git@github.com:mmadian/SWProject.git>

git clone -b develop https://github.com/mmadian/SWProject.git

#### 14.1.1 Repo structure:

The project repo constructs of 4 main folders:

1. Documents: for tracking the documents of the project.
2. Project: for tracking project source code.
3. Graphics: for tracking project graphics (logo, images,).
4. Extra: for extra needed stuff.



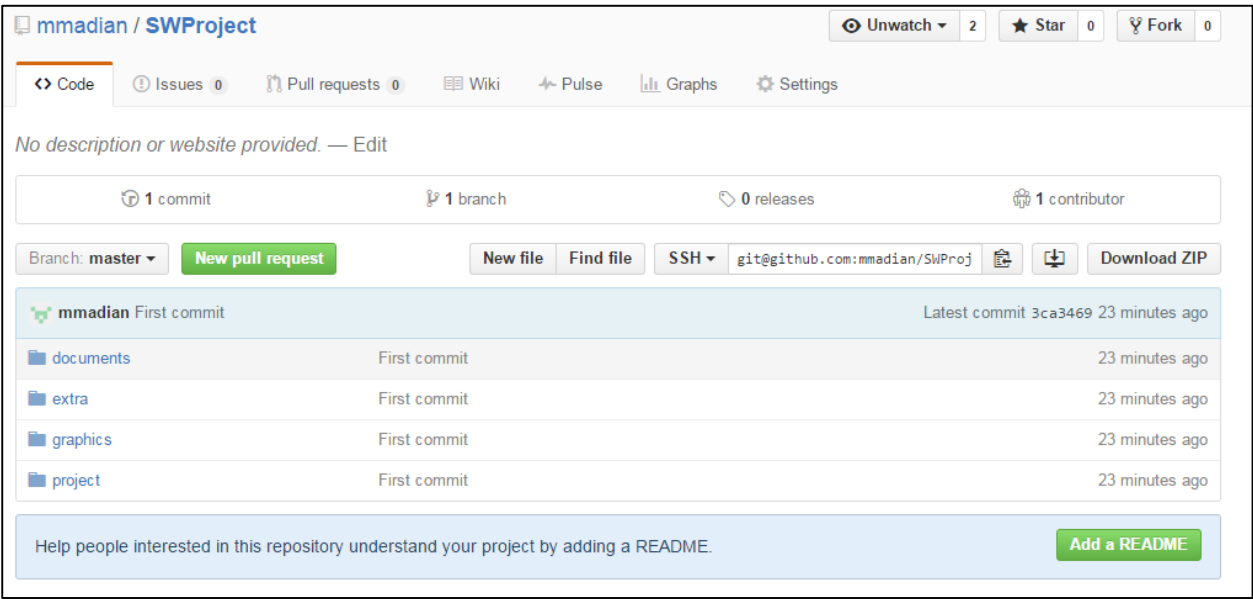


Figure 8 Configuration Management Repository



## 15 Templates

### 15.1 Quality Assurance Audit Reports

Project Title:		Date Prepared:	
Project Auditor:		Audit Date:	

#### Area Audited:

<input type="checkbox"/> Project	<input type="checkbox"/> Project processes	<input type="checkbox"/> Project documents
<input type="checkbox"/> Product	<input type="checkbox"/> Product requirements	<input type="checkbox"/> Product documents
<input type="checkbox"/> Approved change implementation	<input type="checkbox"/> Corrective or preventive action implementation	<input type="checkbox"/> Defect/deficiency repair
<input type="checkbox"/> Quality Management Plan	<input type="checkbox"/> Organizational policies	<input type="checkbox"/> Organizational procedures

#### Description of Good Practices to Share:

#### Description of Areas for Improvement:



**Description of Deficiencies or Defects:**

ID	Defect	Action	Responsible Party	Due Date

**Comments:**



## 15.2 Change Request Template

Change request forms are the primary project management tool used for requesting any changes to a specific project and are one piece of the change management process. All project managers must manage change carefully and implement a thorough change control process to ensure projects remain within their approved constraints. Some projects have many stakeholders with varying levels of interest in the project and change is an inevitable part of any project lifecycle.

The change request form is filled out by the individual who identifies the need for a change and submitted to the project team in accordance with the change control process. The project manager then leads the team in identifying the impacts of the change, whether or not it will benefit the project, and if it will allow the project to proceed within its approved constraints. The request is then submitted to the change control board with the project team's findings where it is reviewed and either approved, rejected, or deferred until clarification can be sought.

If the change is approved, all project documentation must be updated accordingly and the change must be communicated to all stakeholders. Some changes may also require re-baselining of the costs, schedule, or scope. There are many formats for change requests depending on the organization. Below is one example of a thorough change request form.

The following table shows a common format used for a project change request (next page):

### Standard Change Request Template:

<b>Change Request</b>	
<b>Project:</b>	<b>Date:</b>
<b>Change Requestor:</b>	<b>Change No:</b>
<b>Change Category (Check all that apply):</b> <input type="checkbox"/> Schedule <input type="checkbox"/> Cost <input type="checkbox"/> Scope <input type="checkbox"/> Requirements/Deliverables <input type="checkbox"/> Testing/Quality <input type="checkbox"/> Resources	



**Does this Change Affect (Check all that apply):**

- ☐ Corrective Action    ☐ Preventative Action    ☐ Defect Repair    ☐ Updates
- ☐ Other

**Describe the Change Being Requested:**

**Describe the Reason for the Change:**

**Describe all Alternatives Considered:**

**Describe any Technical Changes Required to Implement this Change:**

**Describe Risks to be Considered for this Change:**

**Estimate Resources and Costs Needed to Implement this Change:**

**Describe the Implications to Quality:**

**Disposition:** ☐ Approve ☐ Reject ☐ Defer

**Justification of Approval, Rejection, or Deferral:**

**Change Board Approval:**

Name	Signature	Date



## 15.3 Change Log Template

**Project**

**Title:**

**Date Prepared:**

Change ID	Category	Description of Change	Submitted by	Submission Date	Status	Disposition



## 15.4 Risk Register

Risk Identification		Qualitative Rating				Risk Response		
Risk	Risk Category	Probability	Impact	Risk Score	Risk Ranking	Risk Response	Trigger	Risk Owner
Delay in Schedule				0	1			
New Development environment				0	2			
				0	3			
				0	4			
				0	5			
				0	6			
				0	7			
				0	8			
				0	9			
				0	10			

### Key Terms

- **Risk:** The risk stated in a complete sentence which states the cause of the risk, the risk, and the effect that the risk causes to the project.
- **Risk Category:** Categorization of risks by area of project affected, source of risk or other useful category.
- **Probability:** The likelihood that a risk or opportunity will occur (on a scale from 0 to 10 with 10 being the highest).
- **Impact:** The impact of the risk on the project if the risk occurs (scale from 0 to 10 with 10 being the highest).
- **Risk Score:** Determined by multiplying probability and impact (scale from 0 to 100)
- **Risk Ranking:** A priority list which is determined by the relative ranking of the risks (by their scores) within the project with the number one being the highest risk score.
- **Risk Response:** The action which is to be taken if this risk occurs.
- **Trigger:** Something which indicates that a risk is about to occur or has already occurred.
- **Risk Owner:** The person who the project manager assigns to watch for triggers, and manage the risk response if the risk occurs.



## 16 References

- [1] Software Process Improvement Guid (SPIG) Product Suite Handbook V1.2, Cario:  
Software Engineering Competence Center - SECC, 2010.





---

## Appendix A Installation Guide

We will be setting up a Ruby on Rails development environment on Ubuntu. The reason we're going to be using Ubuntu is because the majority of code you write will run on a Linux server. Ubuntu is one of the easiest Linux distributions to use with lots of documentation so it's a great one to start with.

You'll want to download the latest Desktop version here: <http://releases.ubuntu.com/>

### Installing Ruby

The first step is to install some dependencies for Ruby.

```
sudo apt-get update
sudo apt-get install git-core curl zlib1g-dev build-essential libssl-dev libreadline-dev
libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libcurl4-openssl-dev python-
software-properties libffi-dev
```

Next we're going to be installing Ruby using rvm which is pretty simple:

```
sudo apt-get install libgdbm-dev libncurses5-dev automake libtool bison libffi-dev
curl -L https://get.rvm.io | bash -s stable
source ~/.rvm/scripts/rvm
rvm install 2.2.3
rvm use 2.2.3 --default
ruby -v
```

Now we tell Rubygems not to install the documentation for each package locally and then install Bundler

```
echo "gem: --no-ri --no-rdoc" > ~/.gemrc
gem install bundler
```

### Configuring Git

We'll be using Git for our version control system so we're going to set it up to match our Github account. If you don't already have a Github account, make sure to register. It will come in handy for the future.



Replace my name and email address in the following steps with the ones you used for your Github account.

```
git config --global color.ui true
git config --global user.name "YOUR NAME"
git config --global user.email "YOUR@EMAIL.com"
ssh-keygen -t rsa -C "YOUR@EMAIL.com"
```

The next step is to take the newly generated SSH key and add it to your Github account. You want to copy and paste the output of the following command and paste it here.

```
cat ~/.ssh/id_rsa.pub
```

Once you've done this, you can check and see if it worked:

```
ssh -T git@github.com
```

You should get a message like this:

```
Hi excid3! You've successfully authenticated, but GitHub does not provide shell access.
```

### *Installing Rails*

Since Rails ships with so many dependencies these days, we're going to need to install a Javascript runtime like NodeJS. This lets you use Coffeescript and the Asset Pipeline in Rails which combines and minifies your javascript to provide a faster production environment.

To install NodeJS, we're going to add it using the official repository:

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
sudo apt-get install -y nodejs
```

And now:

```
gem install rails -v 4.2.4
```

If you're using rbenv, you'll need to run the following command to make the rails executable available:

```
rbenv rehash
```

Now that you've installed Rails, you can run the rails -v command to make sure you have everything installed correctly:



```
rails -v  
# Rails 4.2.4
```

If you get a different result for some reason, it means your environment may not be setup properly.

### Setting Up MySQL

Rails ships with sqlite3 as the default database. Chances are you won't want to use it because it's stored as a simple file on disk. You'll probably want something more robust like MySQL or PostgreSQL.

There is a lot of documentation on both, so you can just pick one that seems like you'll be more comfortable with. If you're coming from PHP, you may already be familiar with MySQL. If you're new to databases, I'd suggest skipping to setting up PostgreSQL.

You can install MySQL server and client from the packages in the Ubuntu repository. As part of the installation process, you'll set the password for the root user. This information will go into your Rails app's database.yml file in the future.

```
sudo apt-get install mysql-server mysql-client libmysqlclient-dev
```

Installing the libmysqlclient-dev gives you the necessary files to compile the mysql2 gem which is what Rails will use to connect to MySQL when you setup your Rails app. When you're finished, you can skip to the Final Steps.

### Final Steps

- get into project path
- download the latest version via
  - o "git clone -b develop https://github.com/mmadian/SWProject.git"
- open "config/database.yml" change username and password of database of mysql
- run these commands:
  - o `bundle install`
  - o `rake db:create`
  - o `rake db:migrate`
- run the server with `rails s`
- open browser and got to localhost:3000 or 0.0.0.0:3000



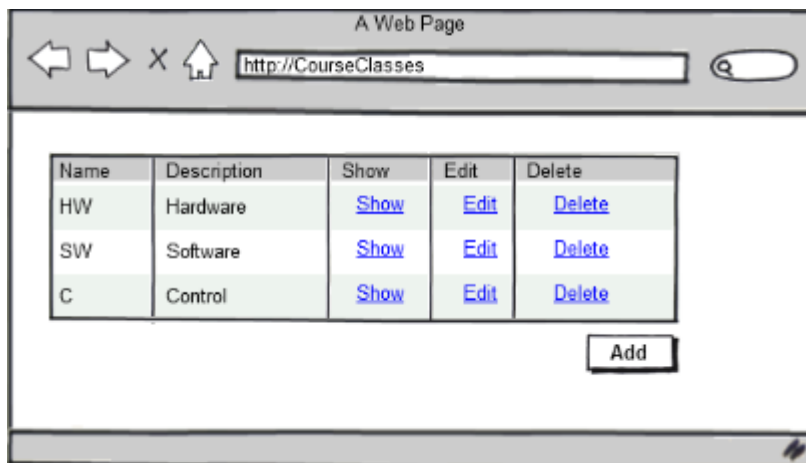
## Appendix B User Guide

### Overview of User Interface

In this section we will illustrate the screens that the developed application provided for the user in order to automate the whole process and give the user of the application the whole information needed to manage and track the students across whole programs and also to adjust the meta data for the application and to provide for the user more convenient way to manage the daily tasks from the CRUD operations, or even the needed reports.

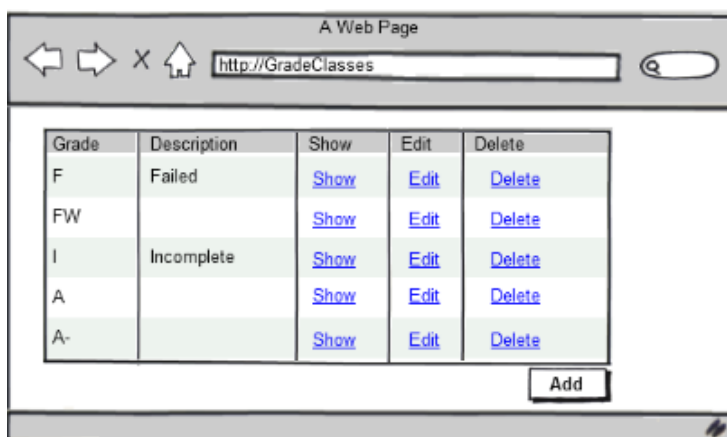
#### Course Manage Screen

In Course Manage Screen User Can List Courses types (HW, SW, C, APP ) , and also Can Add, Update or Delete the selected course type.



#### Grade Manage Screen

In Grade Manage Screen User Can List Grades available (F, FW, I, A, A- ,...etc.) , and also Can Add, Update or Delete the selected grade.





### Course Manage Screen

In Courses Manage Screen User Can List Courses (Data Mining, Software Engineering , ... etc. ) , and also Can Add, Update or Delete the selected course.

Name	Course Class	Course Code	CreditHours	TotalGrade	Min students number	Show	Edit	Delete
Data Mining	SW	123456	3	150	5	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add](#)

### Professor Titles Manage Screen

In Professor Titles Manage Screen User Can List all available titles (TA, DR ) , and also Can Add, Update or Delete the selected Title.

Name	Show	Edit	Delete
TA	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Dr.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add](#)

### Scientific Degree Manage Screen

in Scientific Degree Manage Screen User Can List Scientific Degree available (M.Sc, PHD, ,...etc.) , and also Can Add, Update or Delete the selected one.

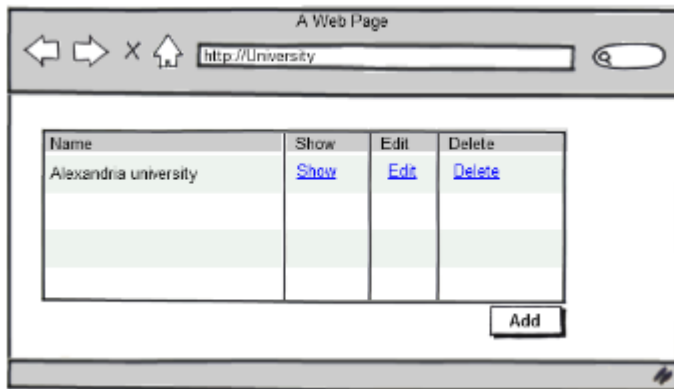
Name	Show	Edit	Delete
B.Sc.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
M.Sc.	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
Diploma of graduate studies	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
PHD	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add](#)



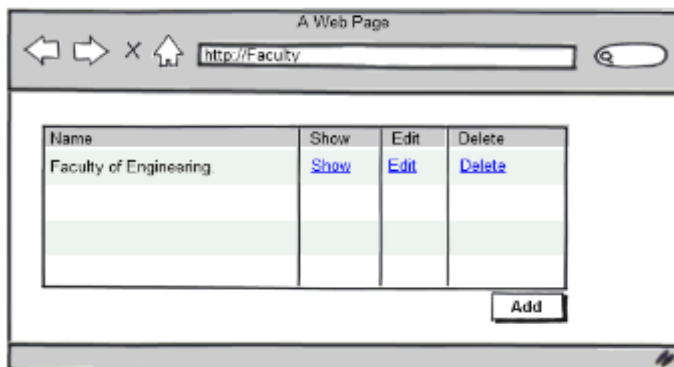
### University Manage Screen

in University Manage Screen User Can List Universities available (Alexandria University ,...etc.) , and also Can Add, Update or Delete the selected one.



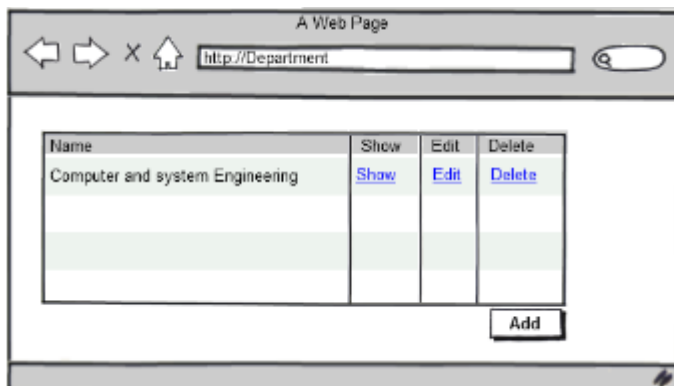
### Faculty Manage Screen

in Faculty Manage Screen User Can List Faculties available (Engineering Faculty ,...etc.) , and also Can Add, Update or Delete the selected one.



### Department Manage Screen

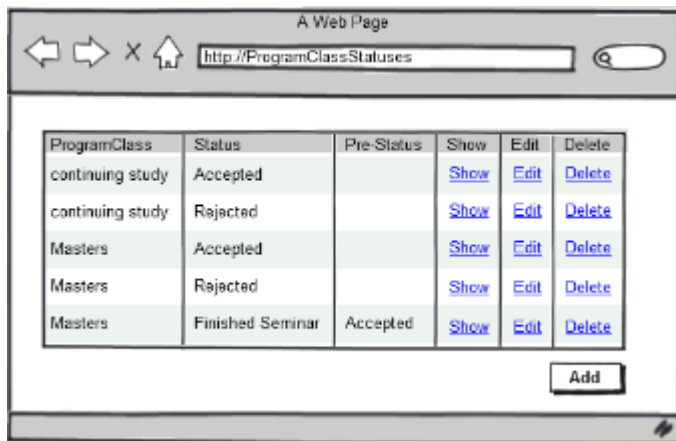
in Department Manage Screen User Can List all Departments available (Computer Science ,...etc.) , and also Can Add, Update or Delete the selected one.





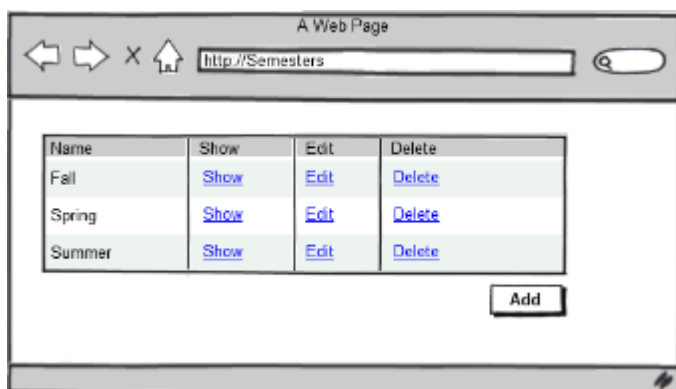
### Program Manage Screen

in Program Manage Screen User Can List all Programs available (Masters,...etc.) , and also Can Add, Update or Delete the selected one.



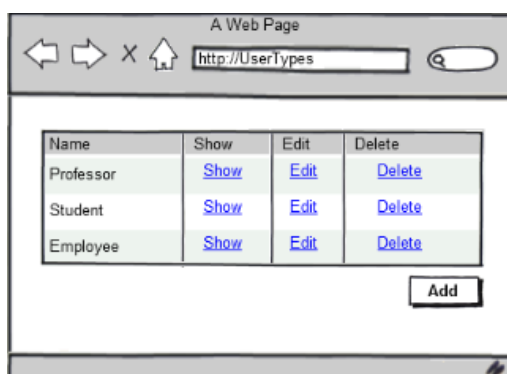
### Semesters Manage Screen

in Semesters Manage Screen User Can List all Semesters available (Fall, Spring,...etc.) , and also Can Add, Update or Delete the selected one.



### User Type Manage Screen

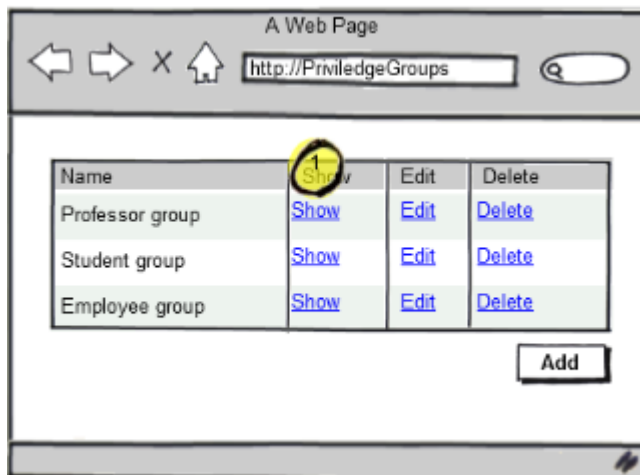
in User Type Manage Screen User Can List all User Type available (Professor, Student, Employee ,...etc.) , and also Can Add, Update or Delete the selected one.





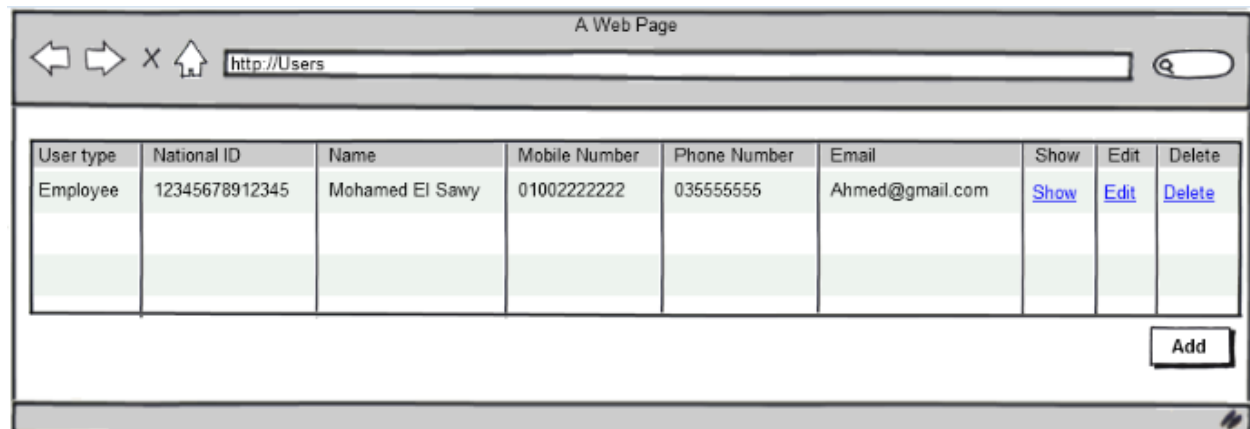
### *Privilege Groups Manage Screen*

in Privilege Groups Manage Screen User Can List all Privilege Groups available (Professor Group, Student Group, Employee Group ,...etc.) , and also Can Add, Update or Delete the selected one.



### *Users Manage Screen*

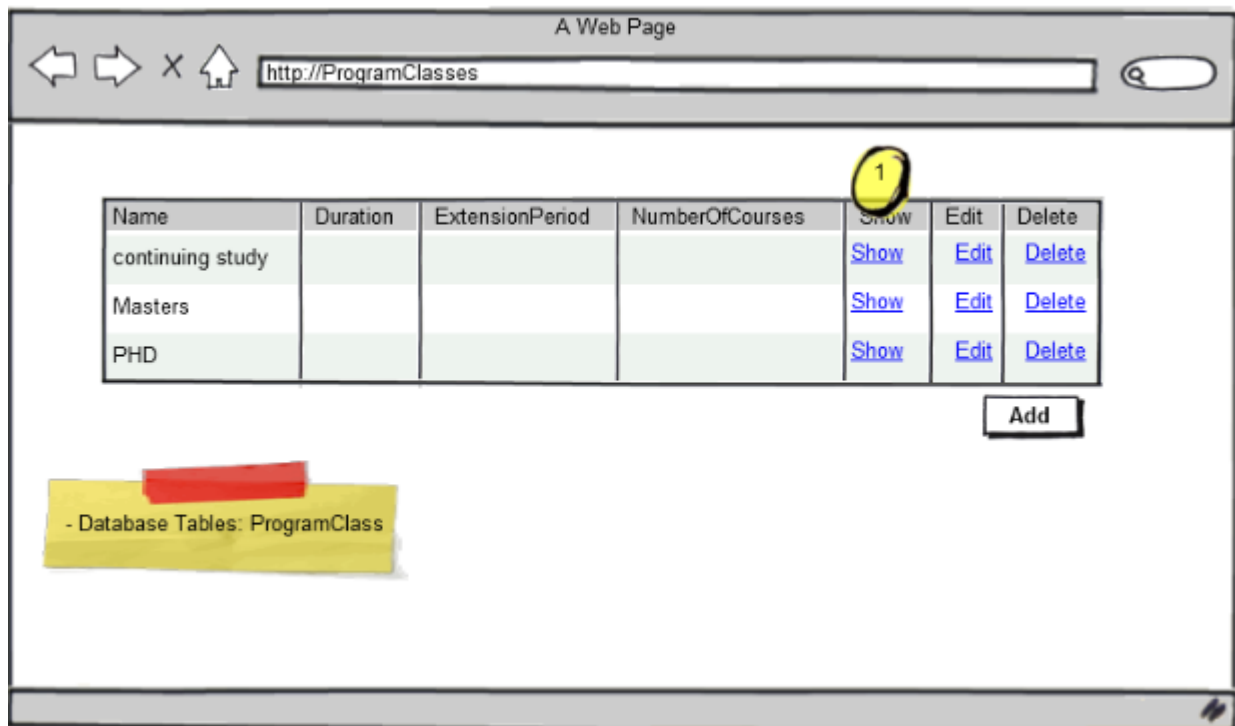
in Users Manage Screen User Can List all Users available , and also Can Add, Update or Delete the selected one.



### *Program Type Manage Screen*

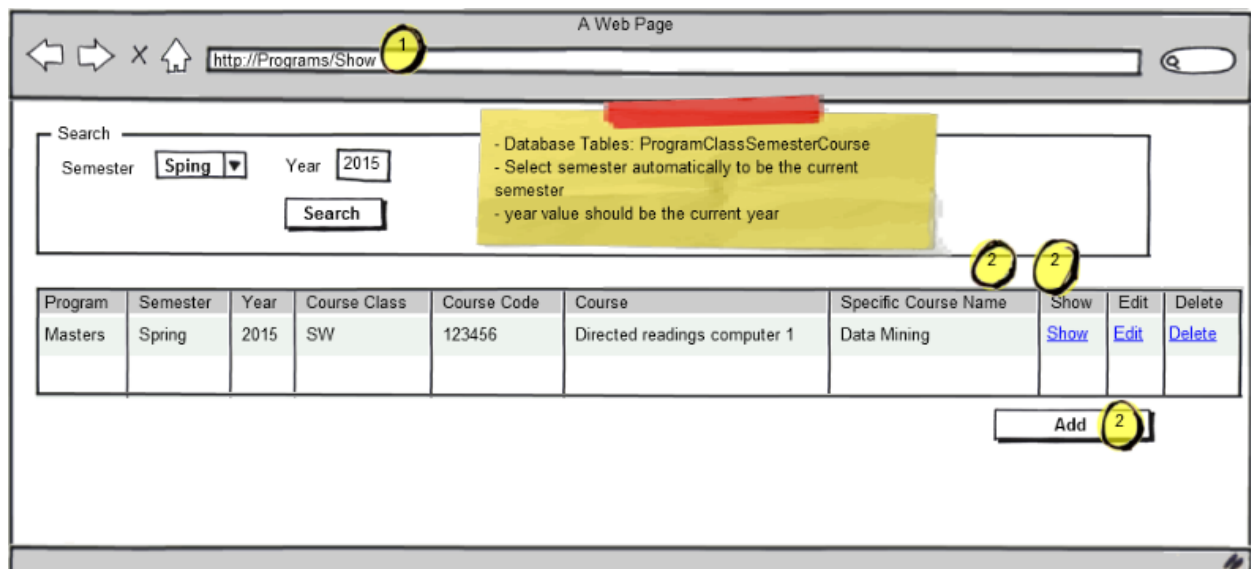
in Program Type Manage Screen User Can List Program Types available (Masters, PHD,...etc.) , and also Can Add, Update or Delete the selected one.





### Programs Manage Screen

in Programs Manage Screen User Can List all Programs available and finished one(Master Spring 2014,...etc.) , and also Can Add, Update or Delete the selected one.





### Program Edit/Add Screen

in Program Edit/Add Screen User Can Change the details of selected program or just add new one.

### Students Manage Screen

in Students Manage Screen User Can List all Students available, and also Can Add, Update or Delete the selected one.

National ID	Name	Mobile Number	Phone Number	Email	Show	Edit	Delete
12345678912345	Ahmed Abd-Allah	01002222222	035555555	Ahmed@gmail.com	Show	Edit	Delete



### Students Edit/Add Screen

in Students Edit/Add Screen User view the selected user with all fields to edit, and also list the registered program and the scientific degree for this user.

The screenshot shows a web page titled "A Web Page" with the URL "http://Students/Add-Show-Edit". The page is divided into three main sections:

- Personal Info:** Contains fields for National ID (12345678912345), Name (Ahmed Abd-Allah), Mobile Number (01002222222), Email (Ahmed@gmail.com), and Phone Number (035555555). A yellow sticky note indicates the database tables: Student, User.
- Scientific Degrees:** A table with columns: Scientific Degree Class, University, Faculty, Department, Graduation Year, Field Of Research, GPA, Show, Edit, Delete. A yellow sticky note indicates the database table: StudentScientificDegree. An "Add" button is at the bottom right.
- Programs:** A table with columns: Program, Start Date, Finished Toefl, Toefl Date, Has Research Point, Pulpished A Paper, Is VTMINA program, Show, Edit, Delete. A yellow sticky note indicates the database table: StudentProgram. An "Add" button is at the bottom right.

### Scientific Degree Edit/Add Screen

in Scientific Degree Edit/Add Screen User view the selected Scientific Degree Edit/Add with all fields to edit, and also list the grades under that Scientific Degree.

The screenshot shows a web page titled "A Web Page" with the URL "http://Student/Scientific Degrees/Add-Show-Edit". The page contains the following elements:

- Form Fields:** Scientific Degree Class (B.Sc.), University (ComboBox), Faculty (ComboBox), Department (ComboBox), GraduationYear (Some text), Field Of Research (Some text), and GPA (Some text). A yellow sticky note indicates the database table: StudentScientificDegree.
- Grades Table:** A table with columns: Name, Year, Grade, Show, Edit, Delete. It contains one row: Prep, 2010, 50. A yellow sticky note indicates the database table: StudentScientificDegreeGrade and states: "According to the type of Scientific Degree class, the section of Scientific Degree Grade will be visible only for B.Sc." An "Add" button is at the bottom right.



### Student Programs Courses Mange Screen

in Student Programs Courses Manage Screen User Can Update the selected one, and also view the professors who teach that course.

### Student Supervisors Manage Screen

in Student Supervisors Manage Screen User Can Update the selected one.

Professor	Is Academic supervisor	Is Primary	Addition D
Ahmed Ali	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 20
Sara Hamdy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	22 Sep. 20

### Student Program Committee Manage Screen

in Student Program Committee Manage Screen User Can Update the selected one.



A Web Page  
http://Student/Program/Committee/Add or Edit

☐ AutoComplete Professor list
 ☐ Is External
 ☐ Has Sent Individual Report
 ☐ Has Sent Satisfaction Report

Add

Professor	Is External	Has Sent Individual Report	Has Sent Satisfaction Report	Addition Date
Ahmed Ali	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2015
Sara Hamdy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015
Soheir Fahmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015

Save

- Database Tables: All Professor left join StudentProgramProfessor where the "Select" column = true for those in StudentProgramProfessor.IsDefenseCommittee = true

### Students Programs Manage Screen

in Students Programs Manage screen user can view the courses, and add the too, also see the Status, warnings, Supervisors and the given grades for that student and also can update, add or delete them.

A Web Page  
http://Student/Programs/Show

Courses

Semester	Year	Course Class	Course Code	Course	Specific Course Name	Course Grade	Student Grade	Student Grade Class	Show	Edit	Delete
Spring	2015	SW	123456	Directed readings computer I	Data Mining	150	111	A	Show	Edit	Delete

Add

- Database Tables: StudentProgramCourse, ProgramClass, SemesterCourse

Status

Status	Date	Show	Edit	Delete
Accepted	Sep. 2010	Show	Edit	Delete
Extension Accepted	Sept 2015	Show	Edit	Delete

Add

- Database Tables: StudentProgramStatusHistory

Warnings

Date	Show	Edit	Delete
Sep. 2011	Show	Edit	Delete
Jan. 2011	Show	Edit	Delete

Add

- Database Tables: StudentProgramWarningHistory

Supervisors

Professor	Is Academic supervisor	Is Primary	Addition Date
Ahmed Ali	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2010
Sara Hamdy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2010

Add/Edit

- Database Tables: StudentProgramProfessor Where IsDefenseCommittee = false

Giving Grades

Thesis Title Arabic:  Some text

Thesis Title English:  Some text

Thesis General Field:  Some text

Thesis Specific Field:  Some text

Save

- Database Tables: StudentProgramGivingGrade

Committee

Professor	Is External	Has Sent Individual Report	Has Sent Satisfaction Report	Addition Date
Ahmed Ali	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	22 Sep. 2015
Sara Hamdy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015
Soheir Fahmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 Sep. 2015

Add/Edit

- Database Tables: StudentProgramProfessor Where IsDefenseCommittee = true



*Appendix C Chapter mapping to SPIG process areas*

	<b>Chapter</b>	<b>Corresponding SPIG phase</b>	<b>Corresponding SPIG Process Area</b>
1	Statement of work	Initiation	PM
2	Minutes of Meetings	Initiation	PM
3	Project Management Plan	Planning	PM
4	Risk Management Plan	Execution, Monitoring and Control Phase	PM
5	Detailed Design	Execution, Monitoring and Control Phase	PD
6	Requirements Elicitation	Planning	PD
7	Requirements Traceability Matrix	Planning	PD
8	Process Activity Task Matrix	Planning	PD
9	Quality Assurance Plan	Execution, Monitoring and Control Phase	QA
10	QA Work Products Audit Checklists	Execution, Monitoring and Control Phase	QA
11	Test Plan	Execution, Monitoring and Control Phase	PR
12	Test Procedures	Execution, Monitoring and Control Phase	PR
13	Configuration Management Plan	Initiation	CM
14	Configuration Management Environment	Closure	CM