

# Software Requirements Specification

Version 1.2

23<sup>rd</sup> March, 2016

## Grade Management System

Submitted in partial fulfillment  
Of the requirements of  
CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223).

The students who submitted this team projects were

<< Ajeet Goyal B14CS003>>

<<Mahak Jain B14CS042>>

<<Mansi Mittal B14SS009>>

<<Shreyansh Chhajer B14SS017>>

# Table of Contents

[Table of Contents](#)

[List of Figures](#)

[1.0. Introduction](#)

[1.1. Purpose](#)

[1.2. Scope of Project](#)

[1.3 Constraints](#)

[1.4 Assumptions and Dependencies](#)

[1.3. Glossary](#)

[1.4. References](#)

[1.5. Overview of Document](#)

[2.0. Overall Description](#)

[2.1 System Environment](#)

[2.2 Functional Requirements Specification](#)

[2.2.1 Use case 1](#)

[Use case:](#)

[2.3 User Characteristics](#)

[2.4 Non-Functional Requirements](#)

[3.0. Requirements Specification](#)

[3.1 Functional Requirements](#)

[3.1.1 View Transcript](#)

[3.1.2 Award Grades](#)

[3.1.3 Edit Grade](#)

[3.1.4 Add Course](#)

[3.1.5 Add Student](#)

[3.1.6 Assign Instructor](#)

[3.1.7 User Login](#)

[3.1.8 Adding Instructor details in the Database](#)

[3.3 Detailed Non-Functional Requirements](#)

[3.4 \*Logical Structure of the Data\*](#)

[4.0 Supporting information](#)

[4.1 Table of contents and index](#)

[4.2 Appendixes](#)

## Document Version History

Date	Version	Description
17 <sup>th</sup> January, 2016	Version 1.0	Scope, Requirements and Use case Tables were added
28 <sup>th</sup> January, 2016	Version 1.1	Use case Diagrams, Class Diagrams, Sequence and Activity diagrams were added
23 <sup>rd</sup> March, 2016	Version 1.2	Class Diagram and Use cases were changed according to implementation

## List of Figures

Figure number	Description
2.1.1	System Use Case Diagram
2.1.2	System Class Diagram
2.2.1(a)(b)	View Transcript :Sequence Diagram,Activity diagram
2.2.2(a)(b)	Give Grades : Sequence Diagram, Activity Diagram
2.2.3	Edit Grades :Sequence Diagram
2.2.4	Add course:Sequence Diagram
2.2.5	Add Student :Sequence Diagram
2.2.6	Assign Instructor:Sequence Diagram
2.2.7	User Login :Sequence Diagram

--	--

2.2.8	Add Instructor:Sequence Diagram
2.2.9	Update User:Sequence Diagram

## **1.0. Introduction**

### ***1.1. Purpose***

The purpose of this document is to present a detailed description of the Course Management System. It will explain the purpose and features of the system, what the system will do and under what constraints the system will work effectively. The system's response to external stimuli and triggers will be detailed. This document is for both developers and stakeholders of the grade management system.

### ***1.2. Scope of Project***

This software system will be designed for educational institutes and schools with semester system to handle grade management system. This system will be designed to maximize the administrator's productivity by providing tools to assist in grade submission and management which would otherwise have to be performed manually. By maximizing the administrator's work efficiency and production the system will meet the administrator's needs while remaining easy to understand and use.

More specifically, this system is designed to allow an administrator to manage and communicate with a group of students and instructors to upload grades to a protected website. This software package will be used by administrator, faculty and students of various institutes. The registered students of the institute will be able to view his/her transcript. The administrator will be able to add courses, faculty and students. The faculty will be able to awards grades and edit them. Both the faculty and admin will be able to view transcript of students.

### ***1.3 Constraints***

- The course duration is one semester, due to the time constraint. The system must be ready in a period of 4 months.
- No hardware or software constraint applicable as such.

### ***1.4 Assumptions and Dependencies***

The database is maintained on the database server in a secure way.

The Admin knows and fills the course list, assigns professors and students to their respective course correctly

### ***1.3. Glossary***

<b>Term</b>	<b>Definition</b>
Grade	A mark indicating the quality of student's work
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Student	Persons currently studying in the target institute
Instructor	Person working in the institute currently teaching courses
Administrator	Person that administers the non-educational requirements of the students and Instructor
Transcript	A document with courses and grades displayed
User	Instructor or Student

### ***1.4. References***

Internet, Instructor and Mentor.

### ***1.5. Overview of Document***

The rest of the document is designed in the following way:

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.



## 2.0. Overall Description

### 2.1 System Environment

Use Case Diagram:

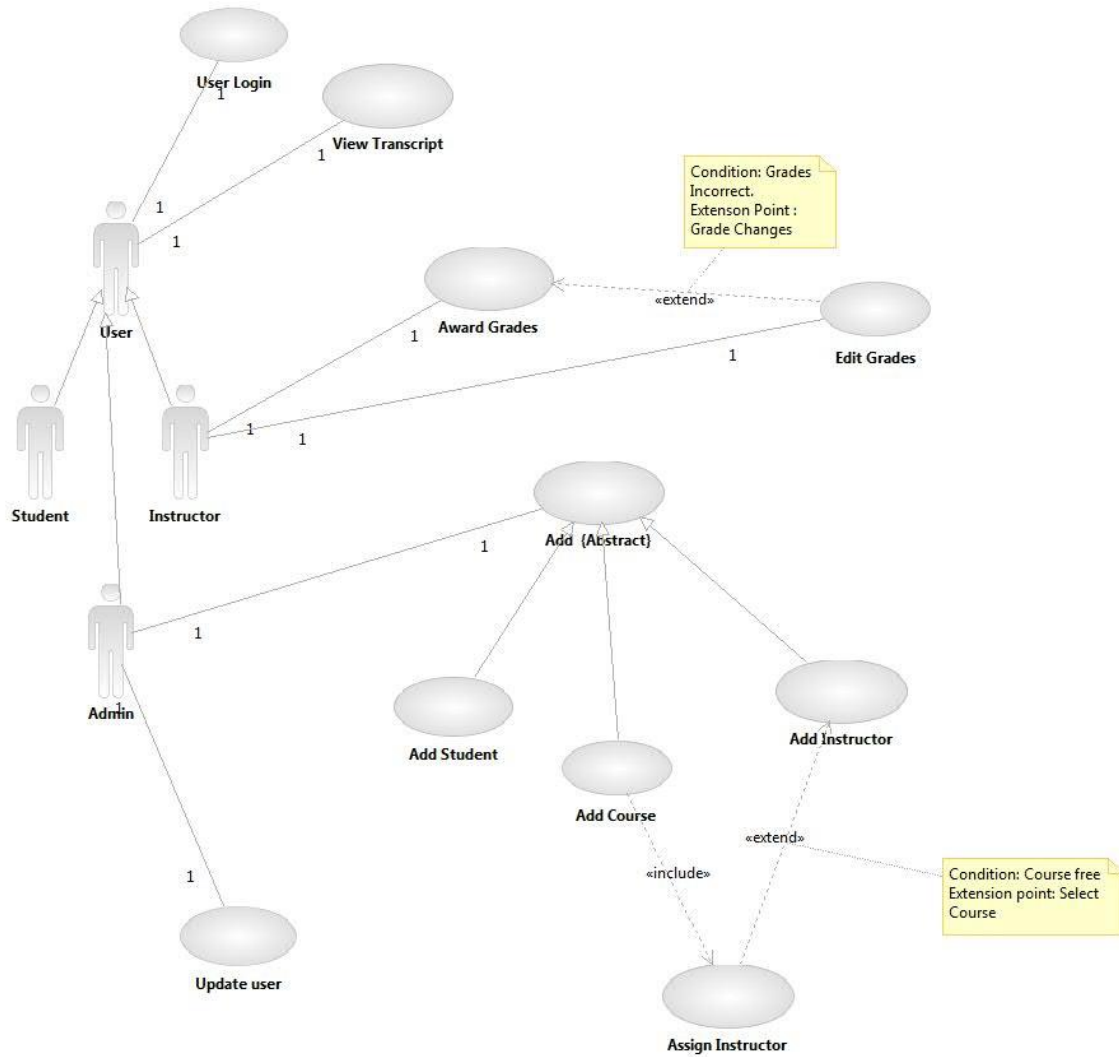
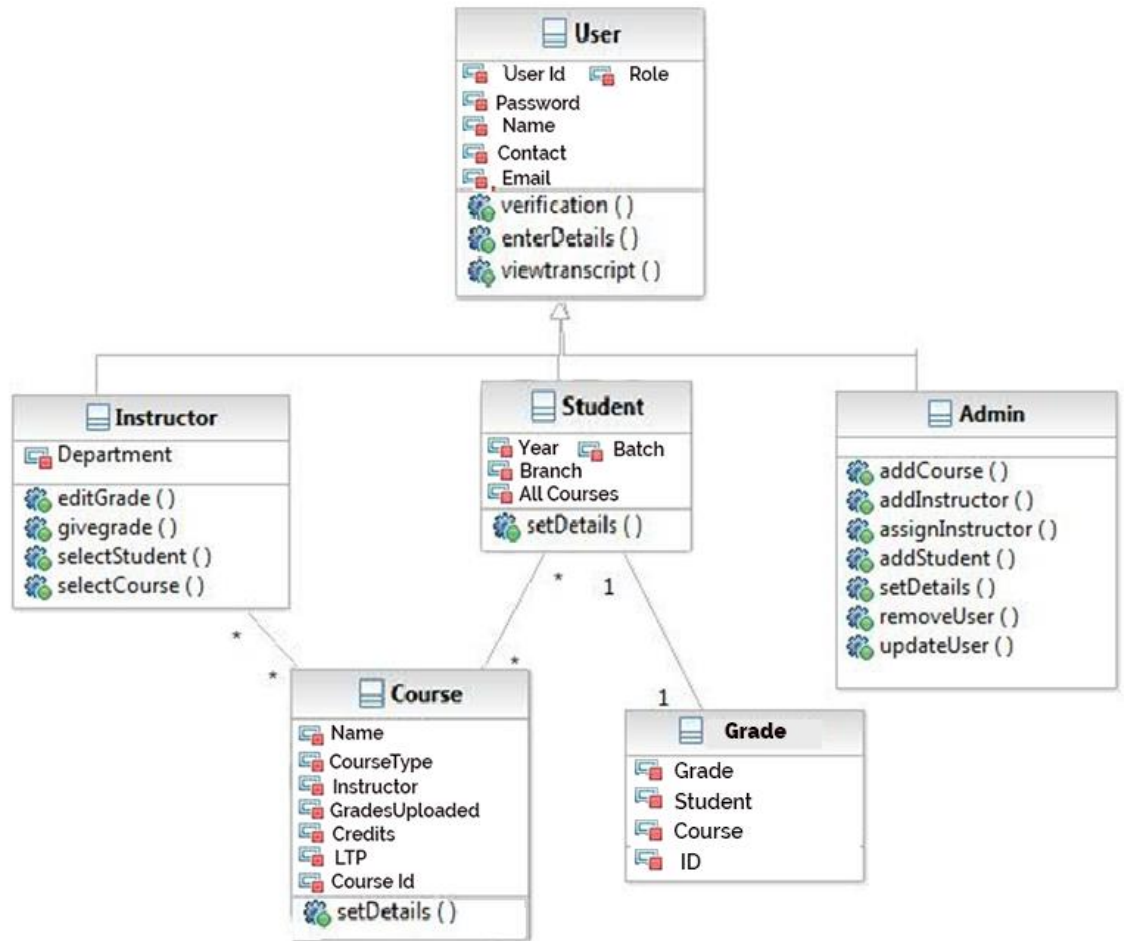


Fig. 2.1.1: System Use Case Diagram

Class Diagram:

Fig. 2.1.2: System Class Diagram

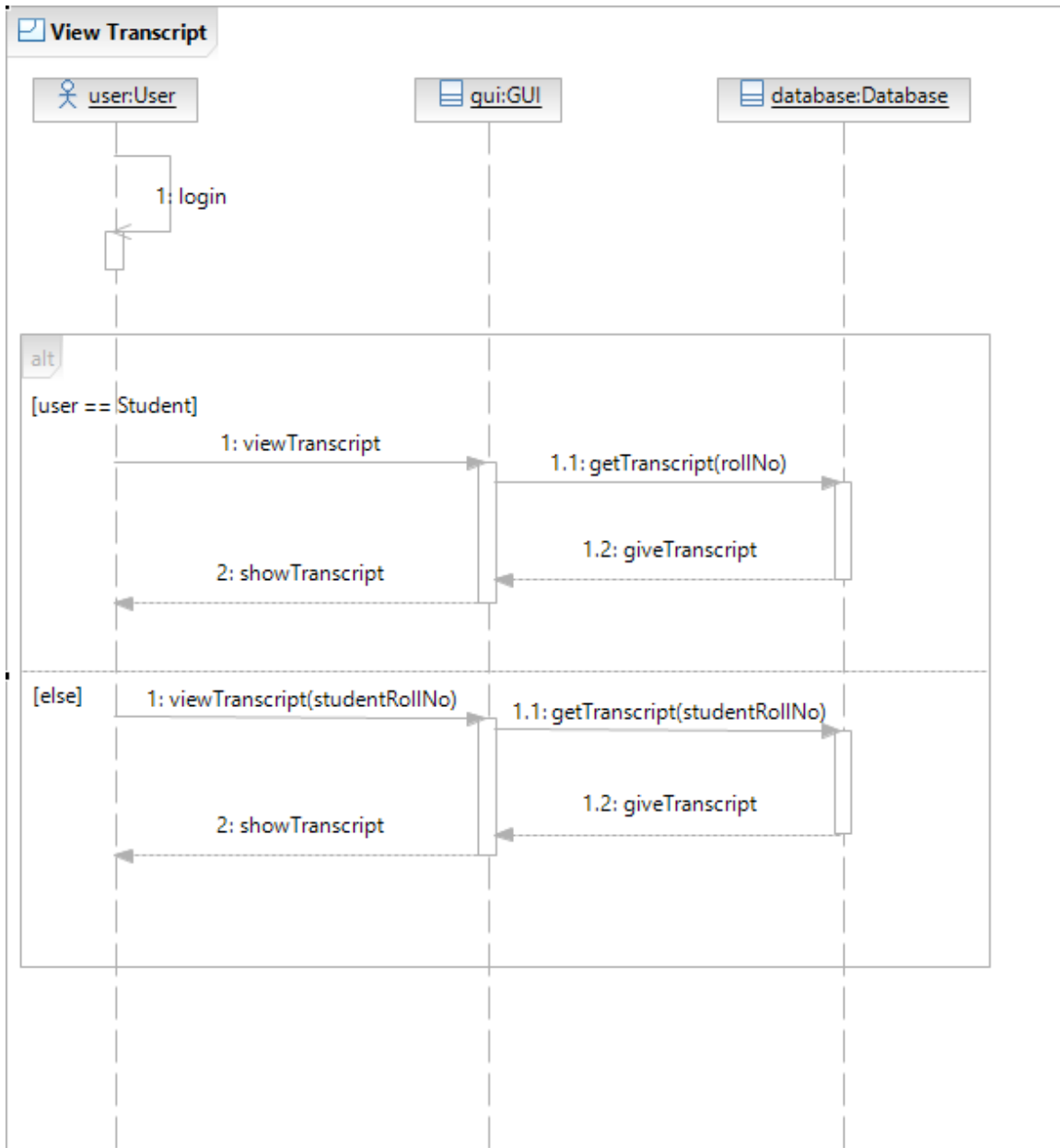


## 2.2 Functional Requirements Specification

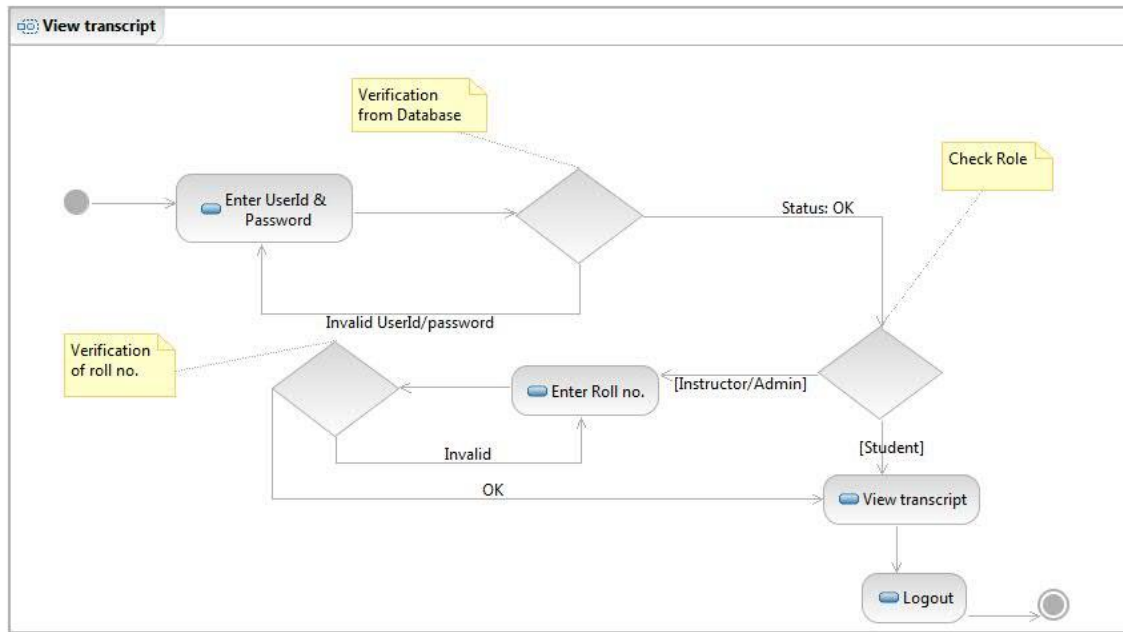
### 2.2.1 Use Cases

1. Use case: View Transcript

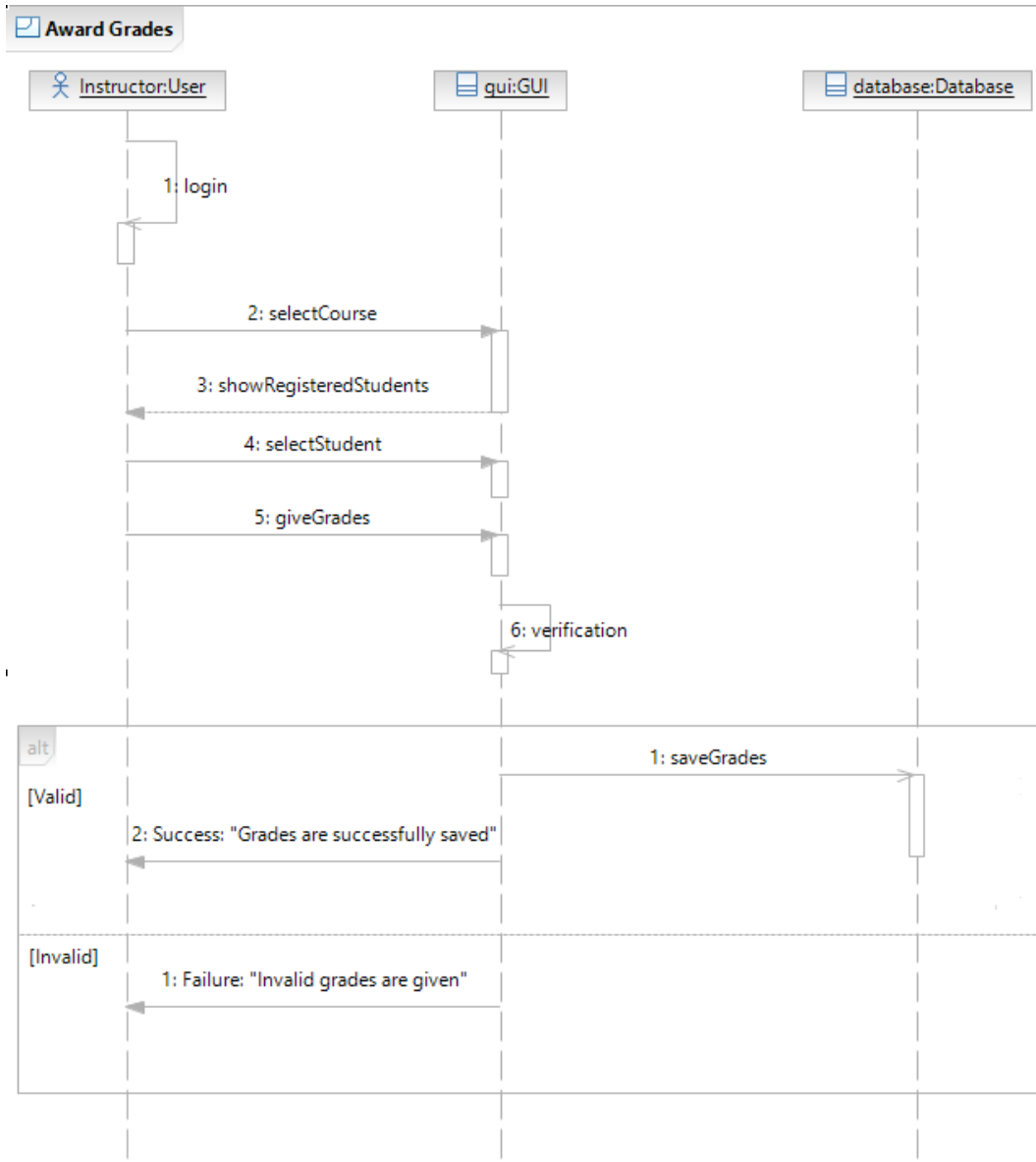
**Sequence Diagram:**



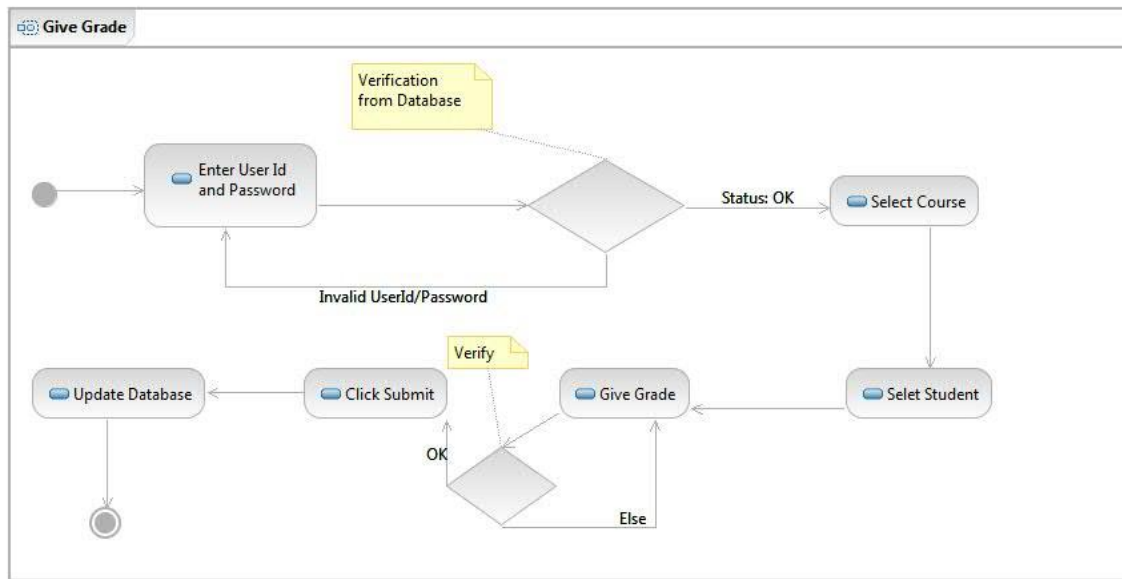
**Activity Diagram:**



2. Use case: Give Grades  
**Sequence Diagram:**

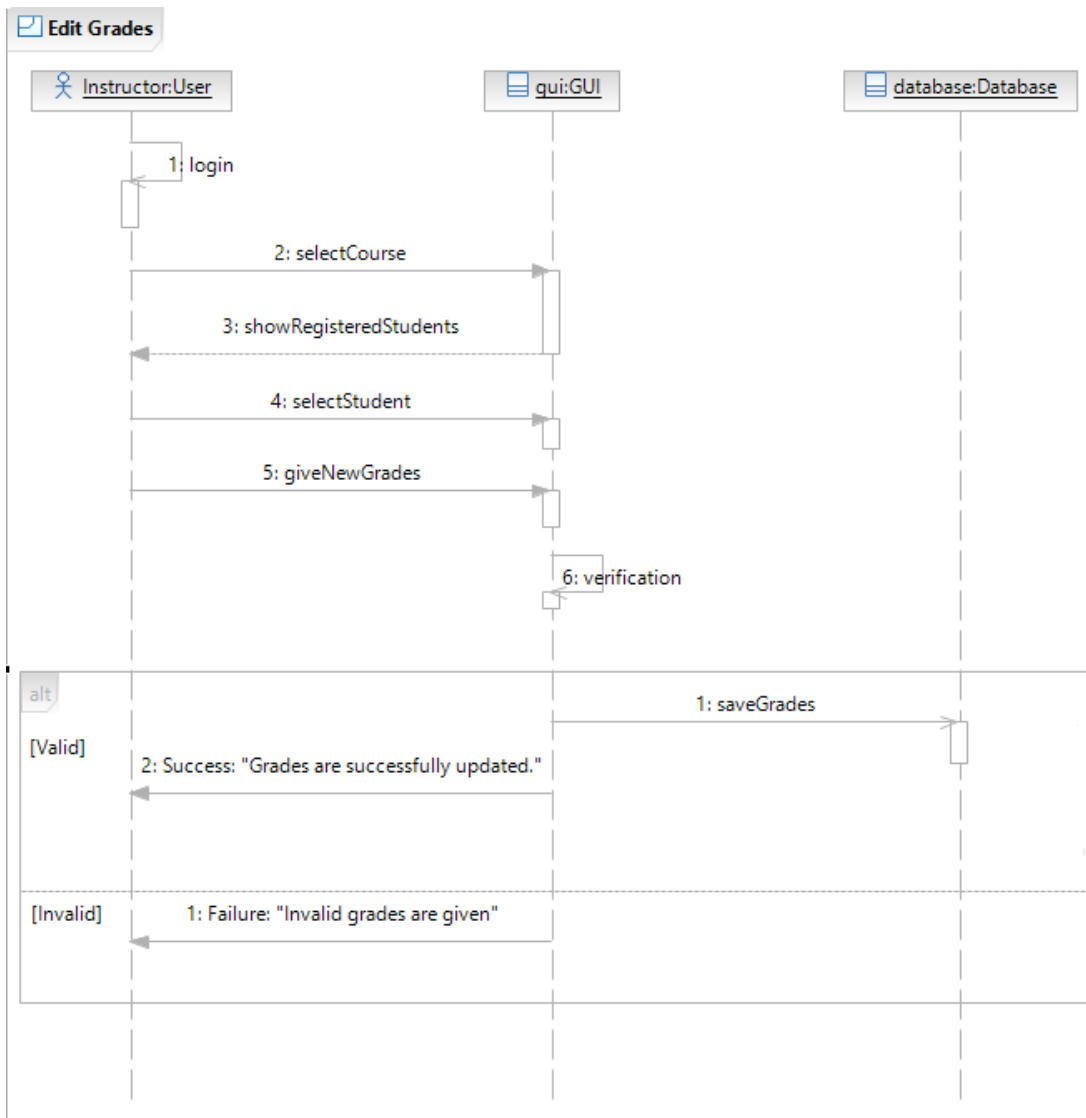


**Activity Diagram:**

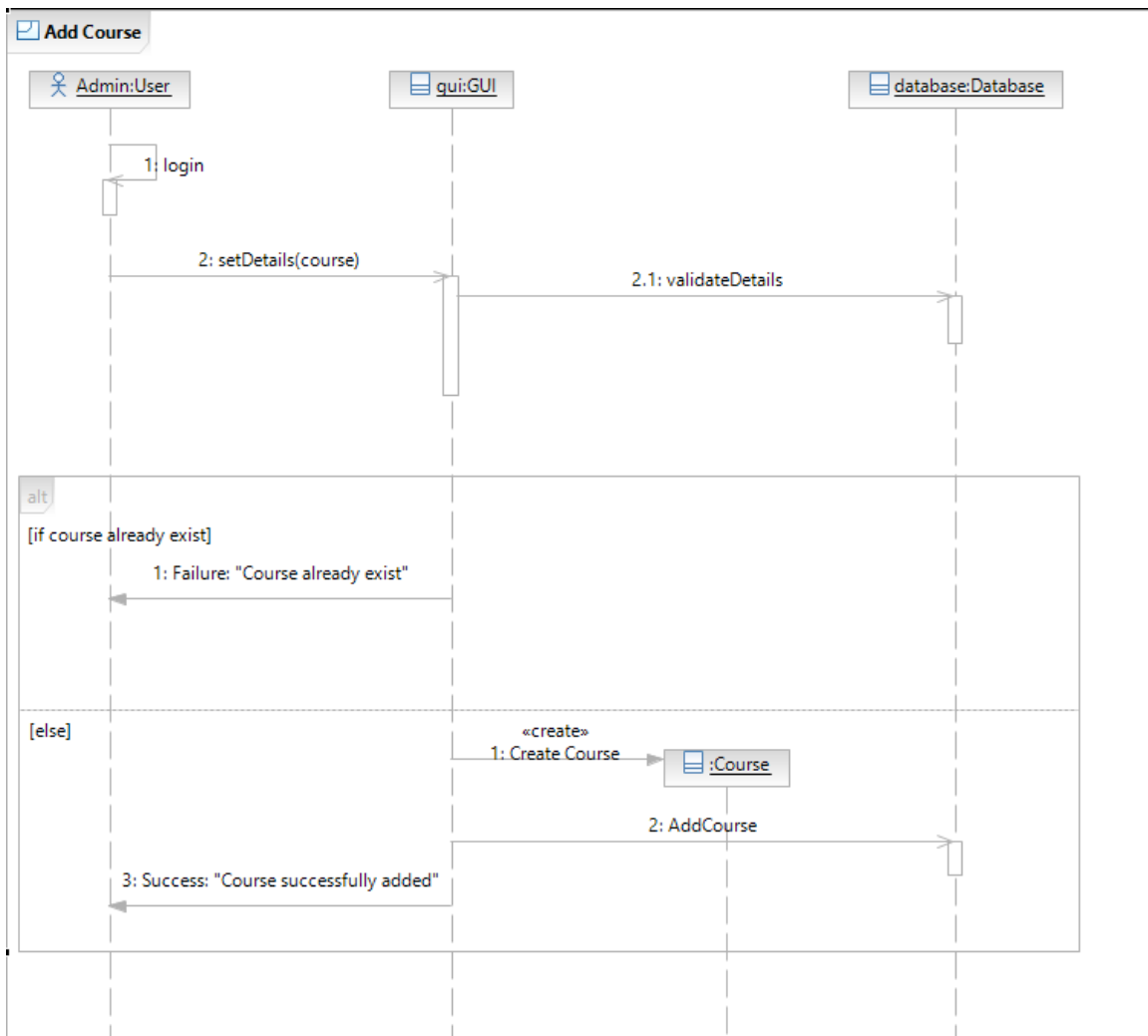


### 3. Use case: Edit Grades

**Sequence Diagram:**

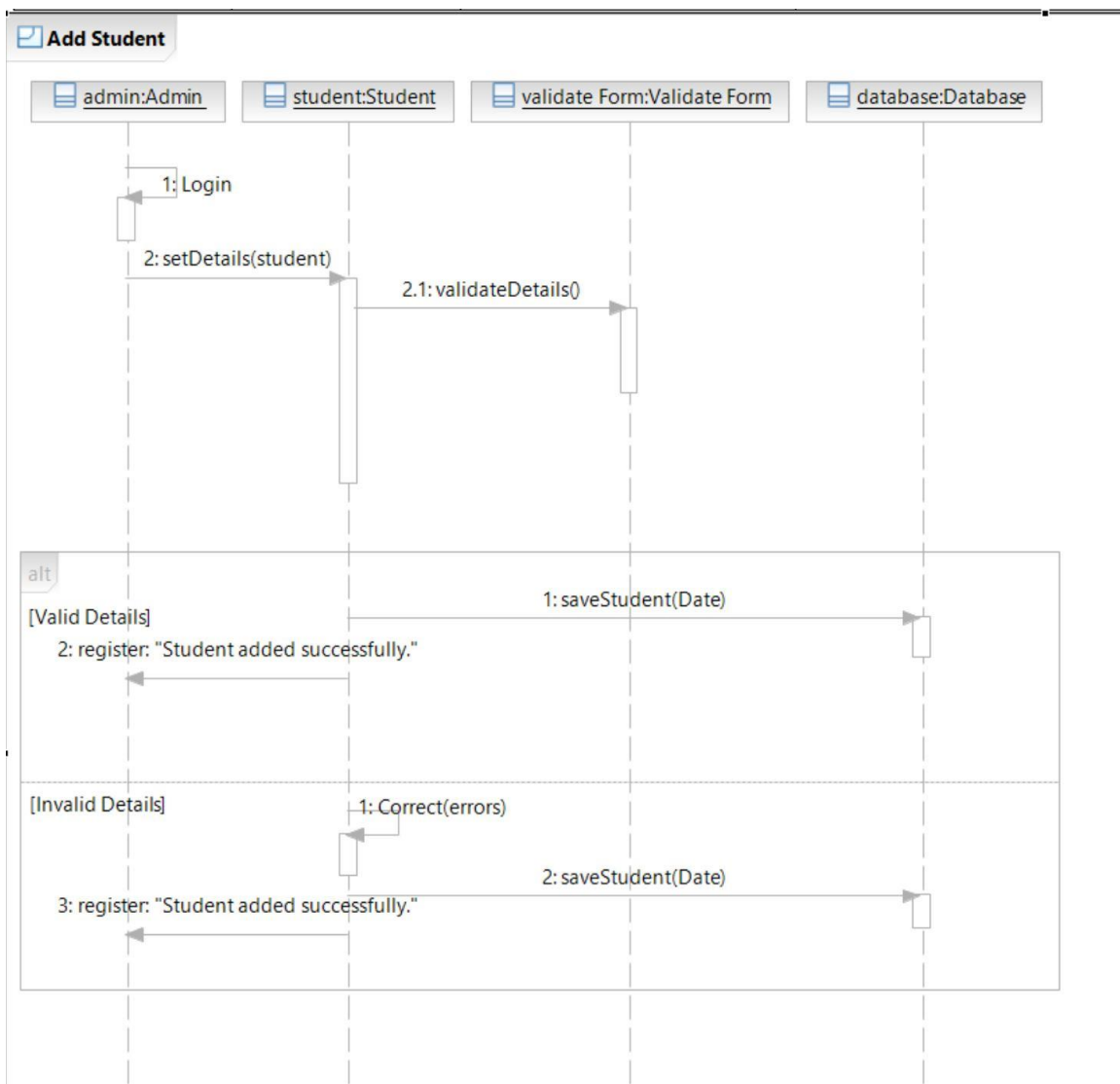


4. Use case: Add course  
**Sequence Diagram:**



5. Use case: Add student  
**Sequence Diagram:**

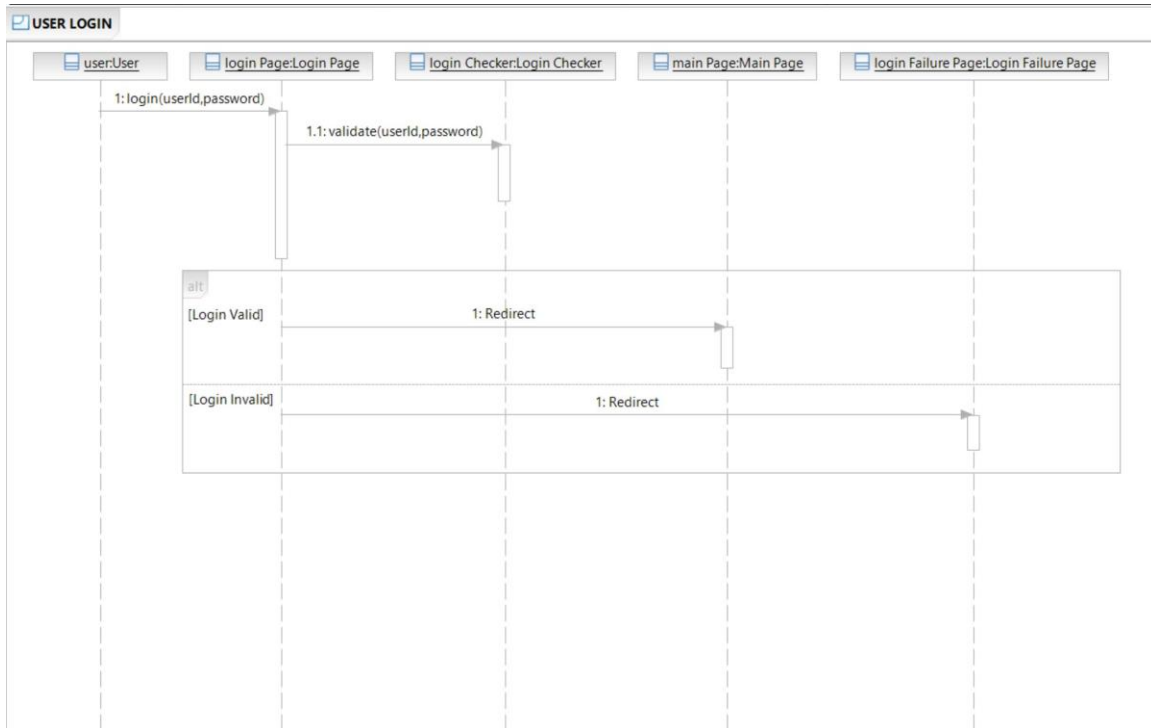




6. Use case: Assign instructor  
**Sequence Diagram:**

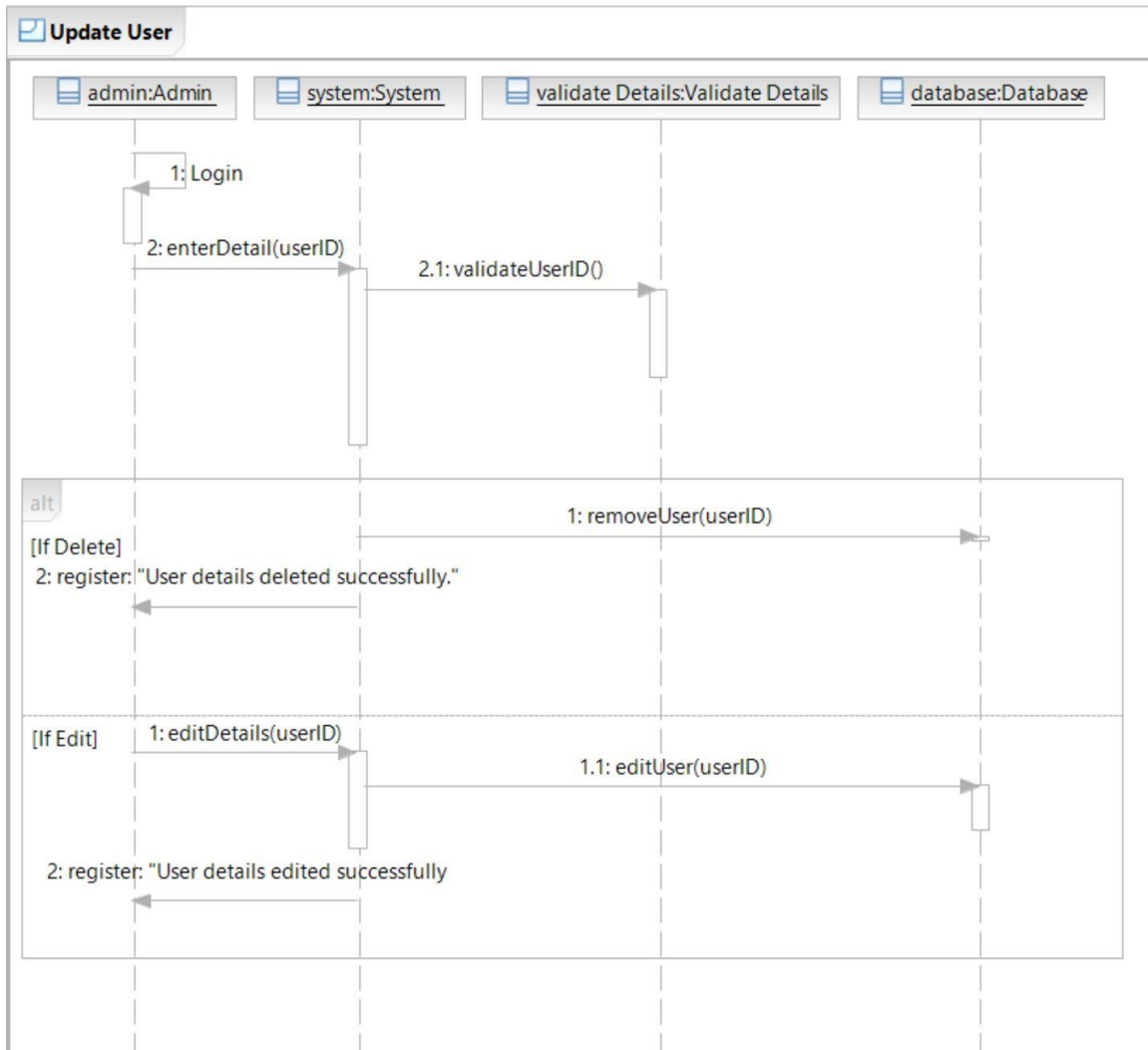


7. Use case: User Login  
**Sequence Diagram:**



8.Use case:Add Instructor  
**Sequence Diagram:**

9. Use case: Update User  
**Sequence Diagram:**



### **2.3     *User Characteristics***

The Student is expected to be Internet literate and be able to use a search engine. The Administrator and Instructor are expected to be Internet literate and to be able to upload files and edit documents. They are also expected to be able to use button, pull-down menus, and similar tools.

### **2.4     *Non-Functional Requirements***

The Online Portal will be on a server with high speed Internet capability. The system must be interactive and the delays involved must be less. The physical machine to be used will be determined by the Educational Institute. Information transmission should be securely transmitted to server without any changes in information the speed of the User's connection will depend on the hardware used rather than characteristics of this system.

## 3.0. Requirements Specification

### 3.1 Functional Requirements

#### 3.1.1 View Transcript

<b>Use Case Name</b>	View Transcript
<b>Trigger</b>	User wants to view transcript.
<b>Precondition</b>	User must be logged in.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. User clicks <b>View Transcript</b> option after login.</li><li>2. For a student, the system will display the respective transcript.</li><li>3. For Instructor or admin,<ol style="list-style-type: none"><li>a.) The system will prompts the user to enter the user-id of a student.</li><li>b.) Instructor or admin enters the required information.</li><li>c.) The system will search the user-id of the student in the database.</li><li>d.) The system will display the required transcript.</li></ol></li></ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	Grades of the registered courses would be displayed.
<b>Exception Paths</b>	<p>If the user-id entered by the instructor or admin is not valid.</p> <ol style="list-style-type: none"><li>c.) The system displays the screen with error being highlighted.</li><li>d.) Instructor or admin will correct the error and resubmit.</li><li>e.) The system will display the required transcript.</li></ol>
<b>Other</b>	The user can be administrator, instructor or student.

#### 3.1.2 Award Grades

<b>Use Case Name</b>	Award Grades
<b>Trigger</b>	Instructor decides to award the grades.
<b>Precondition</b>	<ol style="list-style-type: none"><li>1. The instructor must be logged in.</li><li>2. The instructor must have corrected the answer scripts.</li></ol>
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. Instructor login to the system.</li><li>2. Instructor chooses a course to upload the grades.</li><li>3. The system will display the list of registered students for the course.</li><li>4. Instructor assign grades to the students.</li><li>5. Instructor click on <b>Submit</b> option.</li></ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	Grades are uploaded.

<b>Exception Paths</b>	None
<b>Other</b>	None

### 3.1.3 Edit Grade

<b>Use Case Name</b>	Edit Grades
<b>Trigger</b>	Increment or decrement in marks of student(s).
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. The instructor must be logged in.</li> <li>2. The instructor must have re-corrected the answer scripts.</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Instructor login to the system.</li> <li>2. Instructor chooses a course to change the grades.</li> <li>3. The system will display the list of registered students for the course.</li> <li>4. Instructor make the necessary changes.</li> <li>5. Instructor click on <b>Save Changes</b> option.</li> </ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	Updated grades are uploaded.
<b>Exception Paths</b>	None
<b>Other</b>	None

### 3.1.4 Add Course

<b>Use Case Name</b>	Add Course
<b>Trigger</b>	Administrator needs to add a new course to the Academic System.
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. Administrator must be logged in.</li> <li>2. Course must not already exist in the database.</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Administrator chooses <b>Add Course</b> option from the list of options available on the access page.</li> <li>2. The system will display the form to enter the course details.</li> <li>3. Administrator will enter all the details of the new course including course-id.</li> <li>4. The system will validate all the fields.</li> <li>5. The system will display a message that “Course has been added to the database successfully”.</li> <li>6. Use case ends.</li> </ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	New course is added with a unique course ID.
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>4. The system will display the screen with error being highlighted.</li> <li>5. Administrator will correct the errors and will resubmit the course details.</li> <li>6. The system will display a message that “Course has been</li> </ol>



	added to the database successfully”. 7. Use case ends.
<b>Other</b>	None

### 3.1.5 Add Student

<b>Use Case Name</b>	Add Student
<b>Trigger</b>	New admission to the Academic System.
<b>Precondition</b>	1. Administrator must be logged in. 2. Student profile must not already exist in the database.
<b>Basic Path</b>	1. Administrator chooses <b>Register Student</b> option from the list of option available on the access page. 2. The system will display the form to enter the details of the student. 3. Administrator will enter all the details of the new student including user-id. 4. The system will validate all the fields and default password will be generated which is same as the user-id. 5. The system will display a message that “Student details has been added to the database successfully”. 6. Use case ends.
<b>Alternative Paths</b>	None
<b>Postcondition</b>	New student is registered with a unique student ID.
<b>Exception Paths</b>	If the user-id entered by the administrator is already used. 4. The system will display the screen with errors being highlighted. 5. Administrator will correct the errors and resubmit the student details. 6. The system displays a message that “Student details has been added to the database successfully”. 7. Use case ends.
<b>Other</b>	None

### 3.1.6 Assign Instructor

<b>Use Case Name</b>	Assign Instructor
<b>Trigger</b>	New course is added.
<b>Precondition</b>	1. Administrator must be logged in. 2. Course and Instructor profile must exist in the database.
<b>Basic Path</b>	1. Administrator chooses “Assign Instructor” option from the list of options available on the access page. 2. The system prompts for Course-Id and Instructor-Id. 3. Administrator enters the respective information.

	4. The system validates the Course-Id and the Instructor-Id. 5. The system displays a message that “Instructor has been assigned to the course successfully”. 6. Use case ends.
<b>Alternative Paths</b>	None
<b>Postcondition</b>	Instruction is assigned to a particular course.
<b>Exception Paths</b>	If the Course-Id or Instructor-Id entered by the administrator are not valid. 4. The system displays the screen with error being highlighted. 5. Administrator will correct the errors and resubmit. 6. The system displays a message that “Instructor has been assigned to the course successfully”. 7. Use case ends.
<b>Other</b>	None

### 3.1.7 User Login

<b>Use Case Name</b>	User Login
<b>Trigger</b>	User wishes to login.
<b>Precondition</b>	1. User is not logged into a profile. 2. Input profile exists in database. 3. User password matches profile.
<b>Basic Path</b>	1. The system displays the login page that asks the user to enter username and password. 2. The user submits their username and password to login to the system. 3. The system verifies the username and password from the database. 4. The system displays access page for the respective user. 5. Use case ends.
<b>Alternative Paths</b>	None
<b>Post condition</b>	User is logged in successfully and can use respective functions of the system.
<b>Exception Paths</b>	If the user does not enter the correct username and password. 4. The system does not display access page for the respective user. 5. Use case ends.
<b>Other</b>	The user can be administrator, instructor and student.

### 3.1.8 Adding Instructor details in The Database

<b>Use Case Name</b>	Add Instructor
<b>Trigger</b>	Administrator wishes to add instructor detail.

<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. Administrator must be logged in.</li> <li>2. Instructor profile must not already exist in the database.</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Administrator chooses the “Add Faculty Information” option from the list of options available on the access page.</li> <li>2. The system will display the form (see Appendix - A for a list of fields) to enter the details of the instructor.</li> <li>3. Administrator will enter all the details of the new instructor including user-id.</li> <li>4. The system will validate all the fields (see Appendix - B for all the validation rules) and default password will be generated which is same as the user-id.</li> <li>5. The system will display a message that “Instructor details has been added to the database successfully”.</li> <li>6. Use case ends.</li> </ol>
<b>Alternative Paths</b>	None
<b>Post condition</b>	Instructor details added successfully in the database.
<b>Exception Paths</b>	<p>If the user-id entered by the administrator is already used.</p> <ol style="list-style-type: none"> <li>4. The system will display the screen with error being highlighted.</li> <li>5. Administrator will correct the error and will resubmit the instructor details.</li> <li>6. The system will display a message that “Instructor details has been added to the database successfully”.</li> <li>7. Use case ends.</li> </ol>
<b>Other</b>	

### 3.1.9 Update user

<b>Use Case Name</b>	Update user.
<b>Trigger</b>	<ol style="list-style-type: none"> <li>1. When a student or instructor leaves the academic system.</li> <li>2. When a user information needs to be changed.</li> </ol>
<b>Precondition</b>	1. Input profile exists in database.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. He chooses which type of user to update.</li> <li>3. Enter his unique ID.</li> <li>4. Chooses whether to delete or edit information,</li> </ol>
<b>Alternative Paths</b>	None

<b>Post condition</b>	Information of that user is updated in database.
<b>Exception Paths</b>	1. The user is already removed. 2. The user never existed.
<b>Other</b>	1. The user can be instructor and student only. 2. Update can be either delete user or edit user information.

### ***3.3 Detailed Non-Functional Requirements***

### ***3.4 Logical Structure of the Data***

<< Keep this blank for the time being>>

## **4.0 Supporting information**

### ***4.1 Table of contents and index***

### ***4.2 Appendixes***