Software Requirements Specification

for

Course Management System

Version 1.0

University of Western Ontario - CS 2212

Team 7

Winter 2018

Instructor

Kostas Kontogiannis

Version:	1.0
Print Date:	
Release Date:	Feb. 26, 2018
Release State:	Initial
Approval State:	Draft
Approved by:	
Prepared by:	Maryam Karimi, Jordan McInnis, Fatima Hasan
Reviewed by:	
Path Name:	
File Name:	SRS-G07.pdf
Document No:	

Document Change Control

Version	Date	Authors	Summary of Changes
0.1	Feb. 10	Maryam Karimi	Initial Write - Actor descriptions/ diagram
0.2	Feb. 14	Maryam Karimi	Added Use Case Diagrams
0.3	Feb. 16&17	Maryam Karimi	Added Use Case and Collaboration Diagrams and Domain Model Class Definitions
0.4	Feb. 18	Fatima Hasan	Initial Write - Introduction
0.5	Feb. 19	Fatima Hasan	Added Use Case Descriptions
0.6	Feb. 20&21	Maryam Karimi	Added the Gantt Diagram - made a couple changes to use case diagrams - Domain Model Diagram
0.7	Feb. 22	Fatima Hasan	Added NonFunctional Description
0.8	Feb.23&24	Fatima Hasan	Added Collaboration Diagram and Nonfunctional and Domain Class Diagrams
0.9	Feb. 24	Jordan McInnis	Sequence Diagrams
1.0	Feb. 25	Jordan McInnis	Domain Model Class Definitions

Document Sign-Off

Name (Position)	Signature	Date
Maryam Karimi	MK	Feb. 26, 2018
Jordan McInnis	JM	Feb. 26, 2018
Fatima Hasan	FH	Feb. 26, 2018

Contents

1 Introduction

1.1. Purpose

This document details the requirements for a simple course management system. It is meant to be used as a communication tool between the clients and the developers and give them a shared understanding of the system requirements. The features of the system and the constraints under which they must operate are explained in this document using diagrams and descriptions.

1.2. Overview

The project is about the specification, design, implementation, and unit testing of a simple course management system. In this system, there are three general types of users with three different roles. The system may be simple in the operations it offers but it provides ample opportunities to exercise design skills, and practice the material presented in class.

1.3. References

Eclipse:	http://www.eclipse.org/downloads/index.php
GANTT:	http://wiki.phprojekt.com/index.php/Gantt-diagram

2 Business Scenario Model

2.1. Actors

2.1.1. Overview

The course management system is an environment for course interactions. In this course management system, there are three types of general users with different roles. The system stores information of these actors and courses. In addition, it allows different users access different services and operations. Administrator actors have to start the system before it can be used by student and instructor actors. A more detailed description of these users can be found in section 2.1.2 and 2.1.3.

2.1.2. Actor Diagram

The main actors of the Course Management system are:

- ❖ Administrators
- ❖ Instructors
- ❖ Students
- ❖ The system

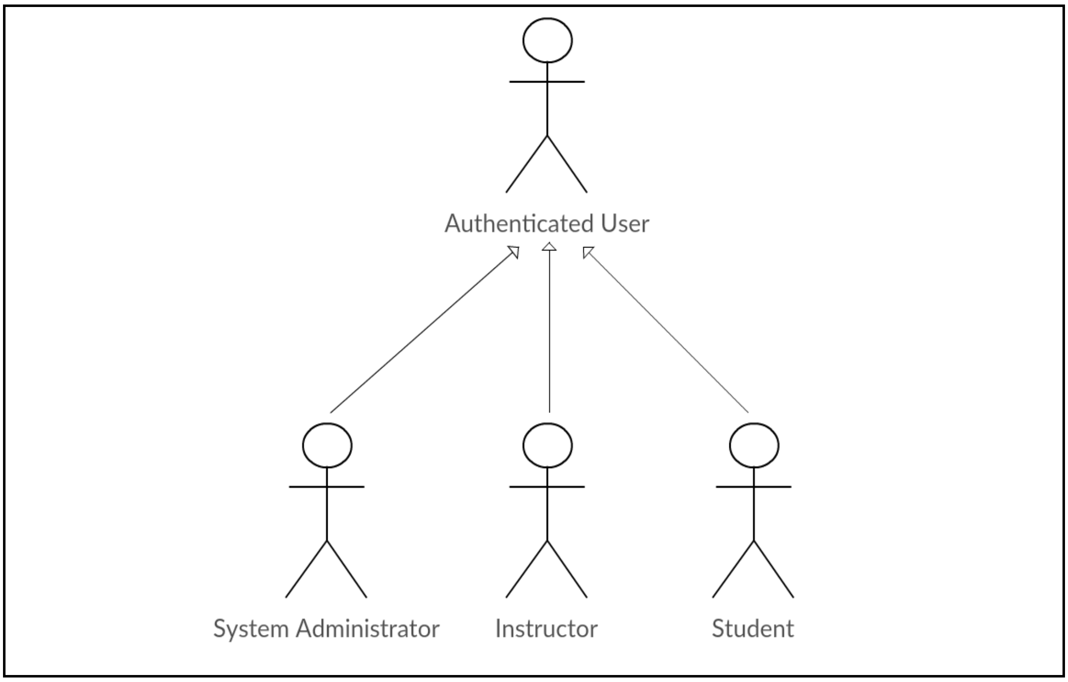


Fig. 1: Overview and structure of involved actors

Figure 1 represents the relationship between the different actors in our system. As demonstrated in the actor diagram, all actors are Authenticated Users, users who have credentials provided by a certified Identity provider. In this system, the identity provider is the Information and Technology Services (ITS) of the University.

2.1.3. Actor Definitions

System Administrator

Description	A System Administrator is a human actor who can login in the system by providing his/her credentials.He/ She starts or stops the system. He/She is responsible for creating courses and upon setting up the course allocates the percentage of marks for each of the EvaluationEntities(final exam, midterm, assignments) used to evaluate the students through the Course Management System. A System Administrator also has the ability to assign Student(s) or an Instructor to a course.
Aliases	Admin
Inherits	System User
Actor Type	Active - Person
Contact Person	Maryam Karimi
Contact Details	mkarimif@uwo.ca

Instructor

Description	An Instructor is a human actor who can login in the system by providing his/her credentials. His/Her roles include adding and modifying marks (for an assignment, midterm, or final exam for a student), selecting an Evaluation Strategy for every different type of student, calculating the final mark, printing the roster of the class, and printing the record for course(s) and for student(s).
Aliases	Tutor
Inherits	System User
Actor Type	Active - Person
Contact Person	Maryam Karimi
Contact Details	mkarimif@uwo.ca

Student

Description	A Student is a human actor who can login in the system by providing his/her credentials and has access to a list of courses he/she is attending. He/She can request enrollment in a course, declare his/her status of attendance/ enrollment in a course, and select a notification method when a mark is added or modified.
Aliases	None
Inherits	System User
Actor Type	Active - Person
Contact Person	Maryam Karimi
Contact Details	mkarimif@uwo.ca

2.2 Use Case Descriptions

This section documents the complete business scenarios within the scope of this project. Overall, we have identified 9 distinct scenarios.

2.2.1 XXXX-0001 Login

In this business scenario, one user(administrator, instructor, student) uses their credentials to request access to the system.

Description:

The users of the system (Students, Instructors, Administrators) have to provide their credentials and type of user to be able to access the system. The system allows each type of user access different operations.

Actors:

The actors associated with this business scenario are as follows:

1. Student
2. Instructor
3. Administrator

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

Before this scenario can be performed:

1. User must have username and password registered with the system (database).
2. The administrator must have started the system, before any other user of the system uses it. In other words, system administrator must be the first person to login.

Scenario Text:

1. User requests to log into the system.

   1.1 System prompts user for credentials.

   1.2 User enters credentials(username and password).

   1.3 Systems creates an authentication token for user.

2. User gains access to system based on the type of user.

Alternative Courses:

1. User requests to log into the system.

   1.1 System prompts user for credentials.

   1.2 User enters credentials(username and password).

   1.3 System displays error message for wrong credentials

2. User clicks on forgot password link.

   2.1 User changes password.

3. User tries logging in again.

Extends:

Error Message.

User Interfaces:

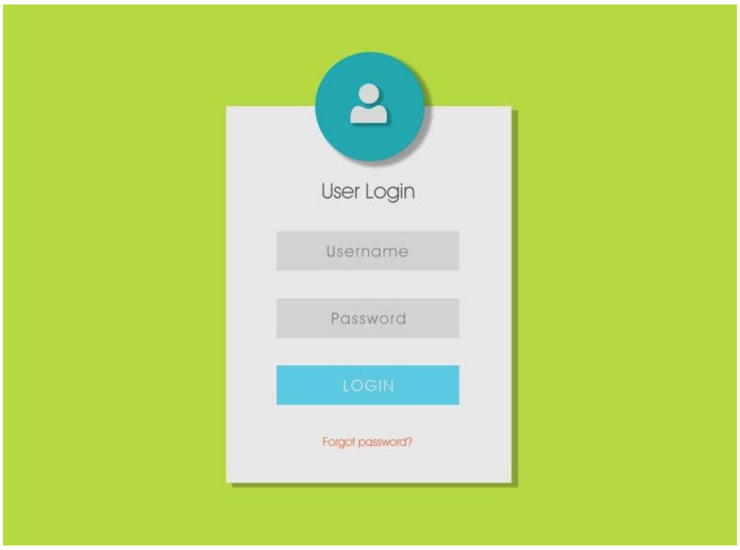


Fig. 2: User Interface for Use Case 2.2.1

Constraints:

1. The user must have login credentials
2. The system must be started already by administrator, before any other user attempts to use the system.

Questions:

Forgot password?

Notes:

This is an include and precondition for all other use cases.

When a user forgets her/his password and gets redirected to change her/his password, they are prompted to enter their birthdate in order to generate a new password for their account.

Authors:

Fatima Hasan

Source Documents:

1. Domain Model
2. User interface: https://dribbble.com/shots/1599405-User-Login-Interface

2.2.2 XXXX-0002 Start the system/Stop the system

In this business scenario model, the administrator is the only user that can start or stop the system.

Description:

The administrator starts the system by clicking on the "start" button and stops it by clicking the "stop" button.

Actors:

The actors associated with this business scenario are as follows:

1. Administrator

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1.Administrator must have username and password registered with the database.

2. Administrator must have signed in into the system.

Scenario Text:

1. *Use Login to gain access to system.*
2. Pushes "start" button to start system.

   2.1 Other authenticated users can use the system.

3. Pushes "stop button to terminate system.

   3.1 Other authenticated users can't use the system anymore.

Alternative Courses:

None.

Extends:

None.

User Interfaces:

None.

Constraints:

1. The authenticated user must be an administrator.

Questions:

None.

Notes:

This is precondition for all other use cases.

Authors:

Fatima Hasan

Source Documents:

1.Project Specification.

2.2.3 XXXX-0003 Adding a course

In this business model, an administrator adds a new course to the course management system.

Description:

The administrator adds a course by providing some informations about it.

The information includes, the course name, course id, course

Actors:

The actors associated with this business scenario are as follows:

1. Administrator

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Authenticated user must be an administrator.
2. System must be in a running state.

Scenario Text:

1. Use Login to access system as administrator.

2. Use Start system to start system.

3. Administrator clicks the "add course" button.

4. Administrator is prompted to enter the course name.

5. Administrator is prompted to enter the course id.

5.1 System checks to see that no other course id with the same id exists.

6. Administrator is prompted to enter the semester.

7. Administrator is prompted to enter the course start date.

7.1 System checks to see if the date entered is valid according to the semester selected.

8. Administrator is prompted to enter the course end date.

8.1 System checks to see if the date entered is valid according to the semester selected.

9.Administrator is prompted to enter the course evaluation entities list.

9.1 System checks to see if there is a final included.

10.Administrator is prompted to enter the number of instructors for the course.

11.Administrator is prompted to enter the course evaluation strategy for full-time credit and audit students, and the course evaluation strategy for part-time credit and audit students.

11.1 System checks to see that no total weight error exists.

Alternative Courses:

None.

Extends:

1. Total weight error.
2. Invalid date error.
3. Invalid course name.
4. Invalid course id.
5. No final error.

User Interfaces:

None.

Constraints:

1. User must be an administrator, in order to add a course.

Questions:

None.

Notes:

Evaluation Entities in this course management system include a final, which is mandatory, a midterm(s), and an assignment(s). The evaluation strategy for the evaluation entities is in the form of percentage for each evaluation entity.

Authors:

Fatima Hasan

Source Documents:

1. Project Description.

2.2.4 XXXX-0004 Adding/Modifying marks for a student in a course

In this business model, an instructor requests access to the system in order to add or modify a students mark.

Description:

The Instructors can add/modify the marks of a student for a course (i.e. his/her final exam mark, midterm mark, assignments marks). The Instructor has to select a course from the list of courses he/she is teaching and select a student by StudentID.

Actors:

The actors associated with this business scenario are as follows:

1. Instructor

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Instructor must have username and password registered with the database.
2. Instructor must have signed in into the system.
3. System must be in a running state.

Scenario Text:

1. *Use start system by administrator so that system is in running state.*
2. *Use Login to access system as an instructor.*
3. Instructor chooses the "add mark" button.
4. Instructor selects the course.
5. Instructor looks up student to add mark by student id.
6. Instructor chooses entity.
7. Instructor enters grade for student.

Alternative Courses:

None.

Extends:

None.

User Interfaces:

None.

Constraints:

Only an instructor can add/modify mark.

Questions:

None.

Notes:

To modify marks, the same scenario text is used, adding a mark for the second time overrides what was already stored.

Authors:

Fatima Hasan

Source Documents:

1. Project Specification

2.2.5 XXXX-0005 Printing records of students

In this business model, an instructor requests access to the system in order to print the records of students.

Description:

The Instructors can print the record of a student for a course (name, ID, and all marks including the final mark) for a course and for an individual student given his/her student ID. The Instructors may choose "all" to print the records of all the students in the class, or provide a list of StudentIDs to print the records of only these students.

Actors:

The actors associated with this business scenario are as follows:

1. Instructor

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Instructor must have username and password registered with the database.
2. Instructor must have signed in into the system.
3. System must be in running state.

Scenario Text:

1. Use Login to access system as administrator.

2. Use Start system to start system.

3. Use Login to access system as instructor.

4. Instructor clicks on the “print student record” button.

4.1 Instructor is then prompted to select a course.

5. Instructor clicks on the “print all records” button.

Alternative Courses:

5. Instructor selects “print record for selected students”

5.1 Instructor enters students IDs.

Extends:

None.

User Interfaces:

None.

Constraints:

1.Only an instructor can print records.

Questions:

None.

Notes:

In the alternative course, when an instructor selects print record for selected students, he/she will include a list of the student IDs for which the records are to be printed.

Authors:

Fatima Hasan

Source Documents:

1.Project Specification

2.2.6 XXXX-0006 Selecting/Changing the EvaluationStrategy

In this business model, an instructor requests access to the system in order to select or change the evaluation strategy.

Description:

An Instructor can change the evaluation strategy for a type of student enrollment for a given course he or she is teaching.

Actors:

The actors associated with this business scenario are as follows:

1. Instructor

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Instructor must have username and password registered with the database.
2. Instructor must have signed in into the system.
3. System must be in a running state.

Scenario Text:

1. Use Login to access system as administrator.

2. Use Start system to start system.

3. Use Login to access system as instructor.

4. Instructor clicks on the “Select/Change Evaluation Strategy” button.

5. Instructor chooses the student type.

6. Instructor chooses evaluation entity.

7. Instructor is then prompted to add weight.

8. Instructor pushes the “Done” button.

    8.1 System checks if the total weight of entities is equal to a 100%.

Alternative Courses:

None.

Extends:

1. Total weight error.

User Interfaces:

None.

Constraints:

1. Only an instructor can select or change the evaluation strategy for a specific course.

Questions:

None.

Notes:

In the scenario text, the instructor is prompted to choose one student type out of four different student types(full-time credit, full-time audit, part-time credit, and part-time audit). Furthermore, when the instructor chooses and evaluation entity and adds weight to it, the instructor is also expected to balance the rest of the entities, so that percentage of all evaluation entities is equal to one hundred. Failing to balance the evaluation entities will result in a total weight error, when trying to end the operation.

Authors:

Fatima Hasan

Source Documents:

1. Project Specifications.

2.2.7 XXXX-0007 Enrolling in a course

In this business model, a student requests access to the system to enroll into a course.

Description:

A Student can enroll in a course he or she is entitled to enroll in. A list of courses a student is allowed to enroll is kept on a data base or an ASCII file. A list of tuples of the form <CourseID, StudentID> is used to keep track of who is entitled to enroll in which class. The student selects a course from the list of available courses, and selects his/her enrollment status.

Actors:

The actors associated with this business scenario are as follows:

1. Student
2. Web Service

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Student must have username and password registered with the database.
2. Student must have signed in into the system.
3. System must be in running state.

Scenario Text:

1. *Use Login to gain access as a student.*
2. Student clicks the “enroll in courses” button.
3. Student selects a course from the list of courses available.

   3.1 System checks if student is entitled to enroll in the course selected.

4. Student enter*s* his/her enrollment status.

5. Student selects his/her notification preferences.

Alternative Courses:

None.

Extends:

1. Not Allowed To Enroll Error Message.

User Interfaces:

None.

Constraints:

1. Only a student can enroll in a course.

Questions:

None.

Notes:

When the student choose an enrollment status, it can be (**I**) either full-time, or part-time and (**II**) either for-credit, or for-audit (a student can be full-time for-audit, or full-time for-credit). A student cannot enroll in a course he/she is not entitled to enroll (e.g. is in a different year, doesn't have prerequisites). A student upon enrollment in a course has also to select his / her notification preferences from a list of predefined options (email, SMS, etc.)

Authors:

Fatima Hasan

Source Documents:

1. Project Specification

2.2.8 XXXX-0008 Changing notification preferences

In this business model, a student requests access to the system to change his/her notification preferences.

Description:

A Student can login into the system, select a course in which he/she is registered and change his/her notification choices or add a new one from the ones available.

Actors:

The actors associated with this business scenario are as follows:

1. Student

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Student must have username and password registered with the database.
2. Student must have signed in into the system.
3. System must be in a running state.

Scenario Text:

1. *Use Login to access system as a student.*
2. Student clicks on "change notification preferences" button.
3. Student selects course.
4. Selects notification preference.

Alternative Courses:

None.

Extends:

None.

User Interfaces:

None.

Constraints:

1. Only a student can change notification preferences for a course.

Questions:

None.

Notes:

The student selects a notification preference from a predefined list of notification preferences.

Authors:

Fatima Hasan

Source Documents:

1.Project Specification.

2.2.9 XXXX-0009 Student Notification

Description:

A Student will be notified in all of its notification targets (email, SMS) when a mark is added or modified.

Actors:

The actors associated with this business scenario are as follows:

1. System.
2. Student.

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

The necessary preconditions are:

1. Student must have username and password registered with the database.
2. Student must have signed in into the system.

Scenario Text:

1.Student will receive a notification based on the preference chosen.

Alternative Courses:

None.

Extends:

Not applicable.

User Interfaces:

None.

Constraints:

None.

Questions:

None.

Notes:

The system will always be updated.

Authors:

Fatima Hasan

Source Documents:

None

2.3 Use Case Diagrams

This section presents the business scenarios of the subject area in a graphical form.

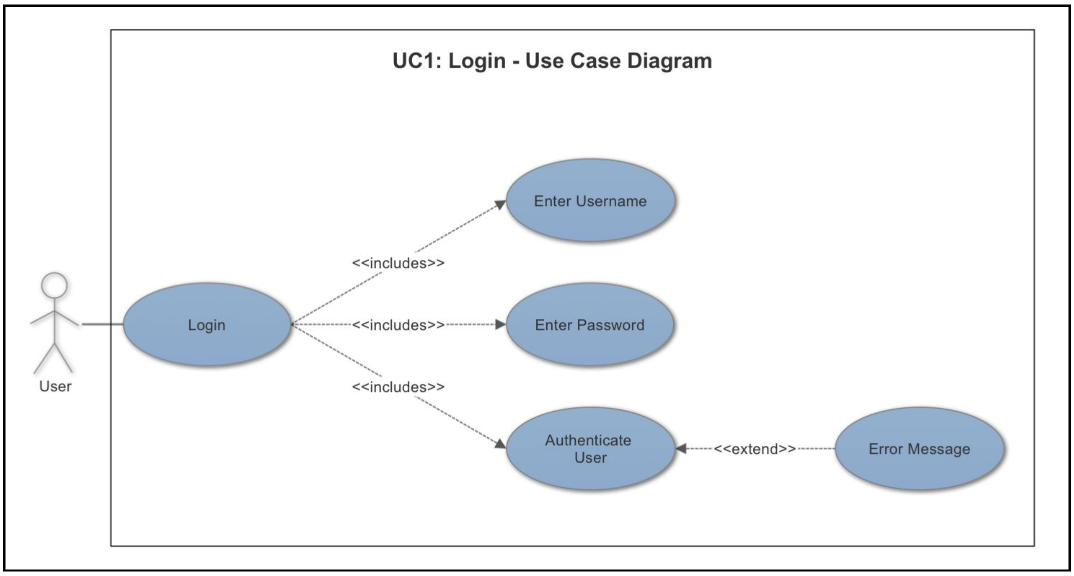


Fig. 3: Use Case Diagram for Scenario 2.2.1

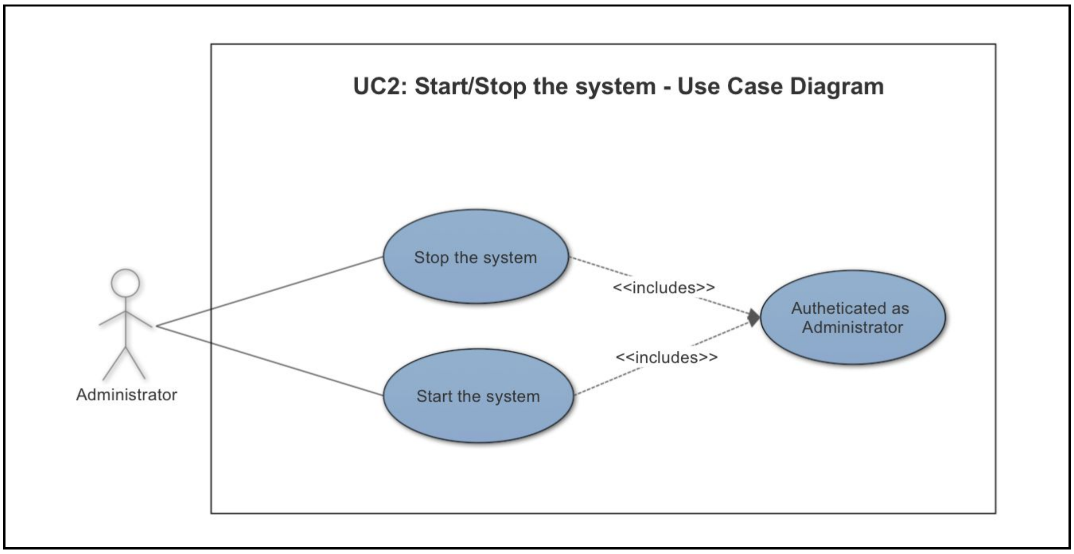

UC2: Start/Stop the system - Use Case Diagram
Administrator
Stop the system
Start the system
<<includes>>
<<includes>>
Autheticated as Administrator

Fig. 4: Use Case Diagram for Scenario 2.2.2

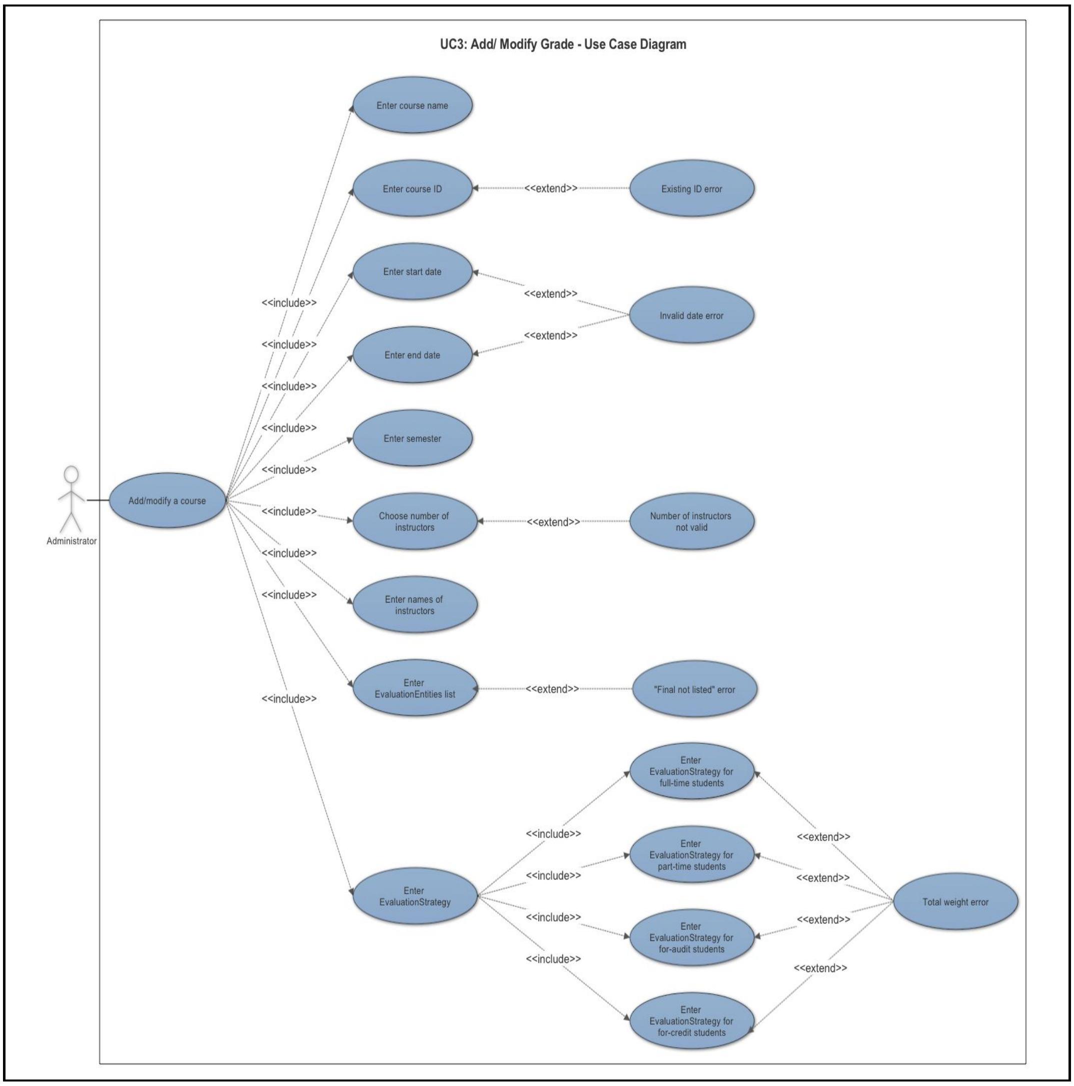

UC3: Add/ Modify Grade - Use Case Diagram
Administrator
Add/modify a course
<<include>>
Enter course name
Enter course ID
<<extend>>
Existing ID error
Enter start date
Enter end date
<<extend>>
<<extend>>
Invalid date error
Enter semester
Choose number of instructors
<<extend>>
Number of instructors not valid
Enter names of instructors
Enter EvaluationEntities list
<<extend>>
"Final not listed" error
Enter EvaluationStrategy
<<include>>
Enter EvaluationStrategy for full-time students
Enter EvaluationStrategy for part-time students
Enter EvaluationStrategy for for-audit students
Enter EvaluationStrategy for for-credit students
<<extend>>
Total weight error

Fig. 5: Use Case Diagram for Scenario 2.2.3

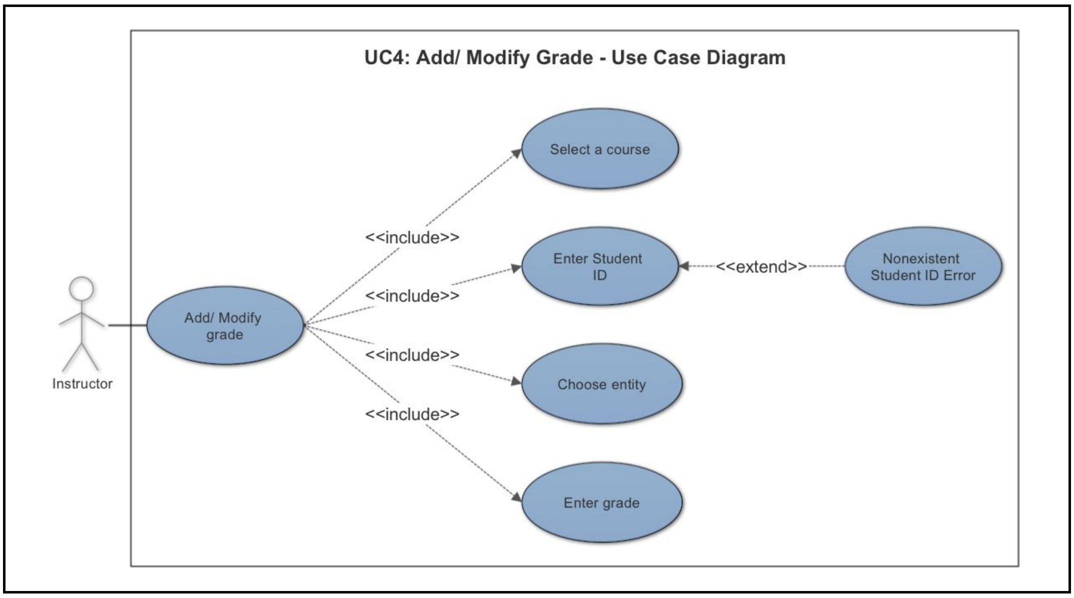


Fig. 6: Use Case Diagram for Scenario 2.2.4

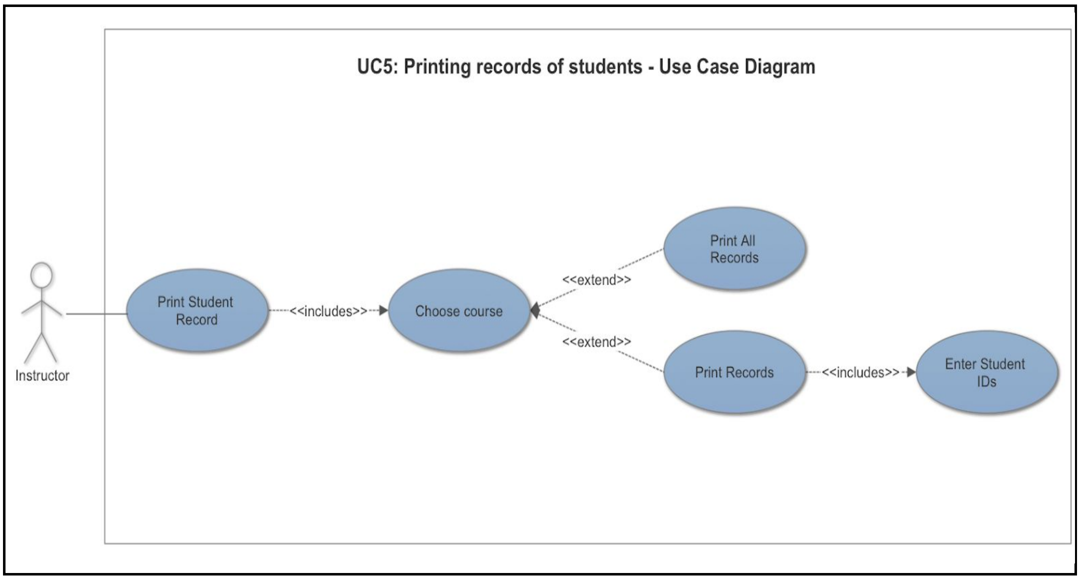


Fig. 7: Use Case Diagram for Scenario 2.2.5

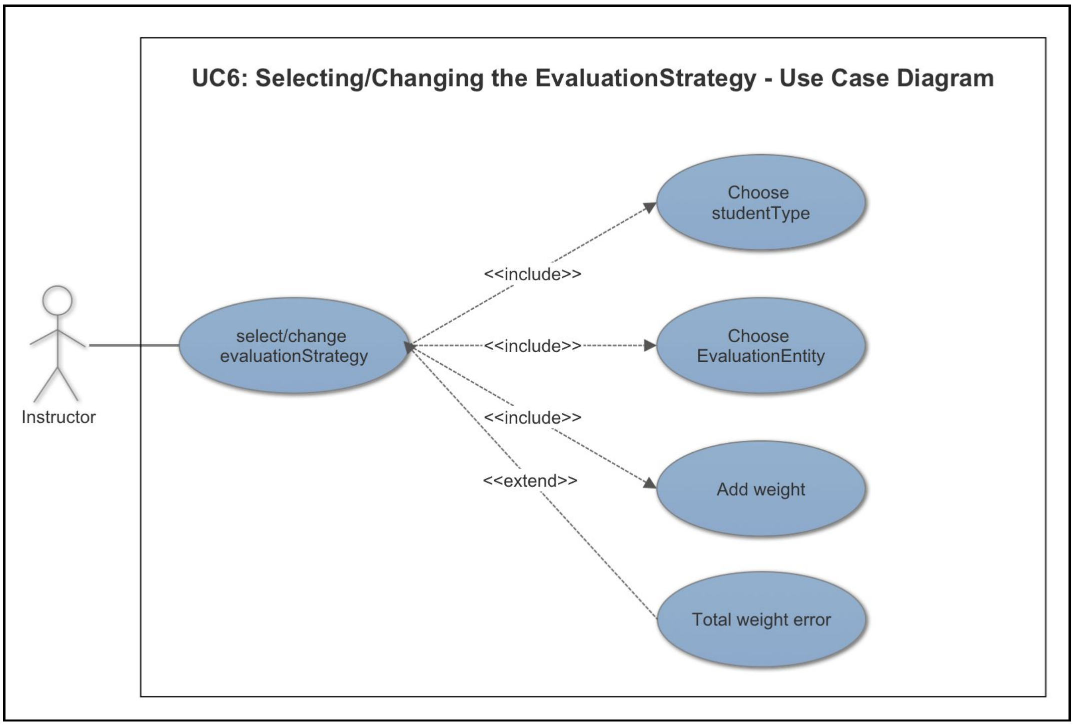


Fig. 8: Use Case Diagram for Scenario 2.2.6

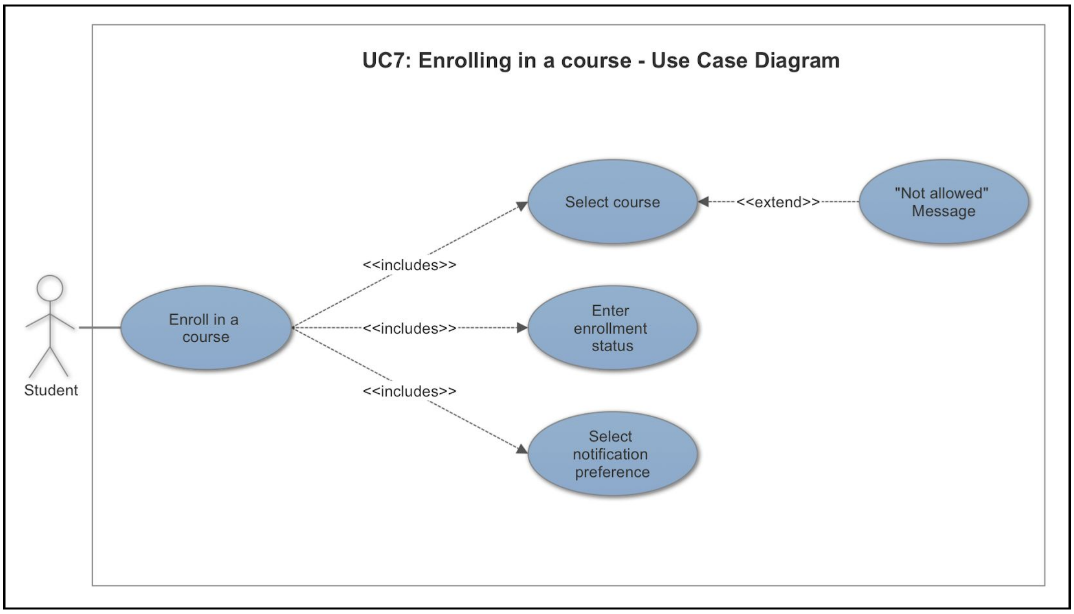


Fig. 9: Use Case Diagram for Scenario 2.2.7

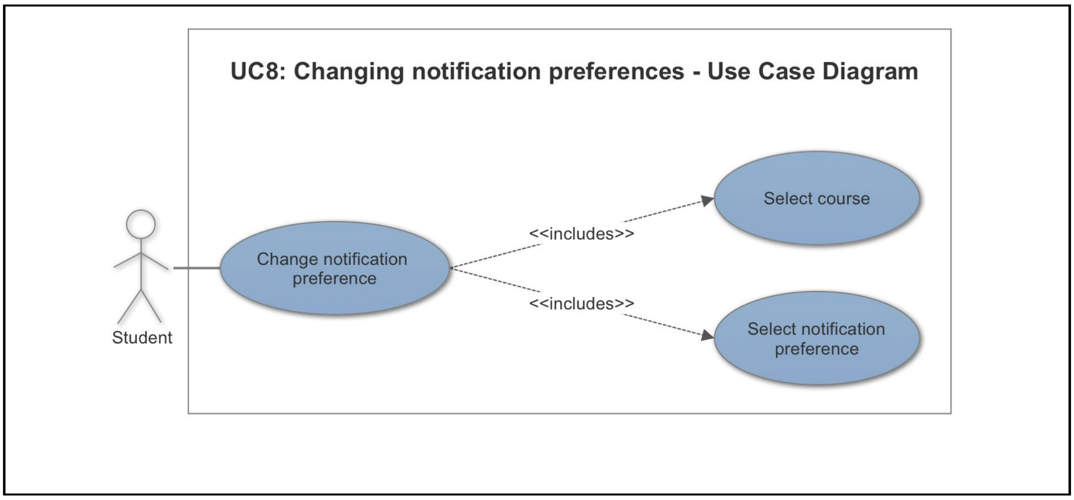

UC8: Changing notification preferences - Use Case Diagram
Student
Change notification preference
<<includes>>
<<includes>>
Select course
Select notification preference

Fig. 10: Use Case Diagram for Scenario 2.2.8

3 Domain Model

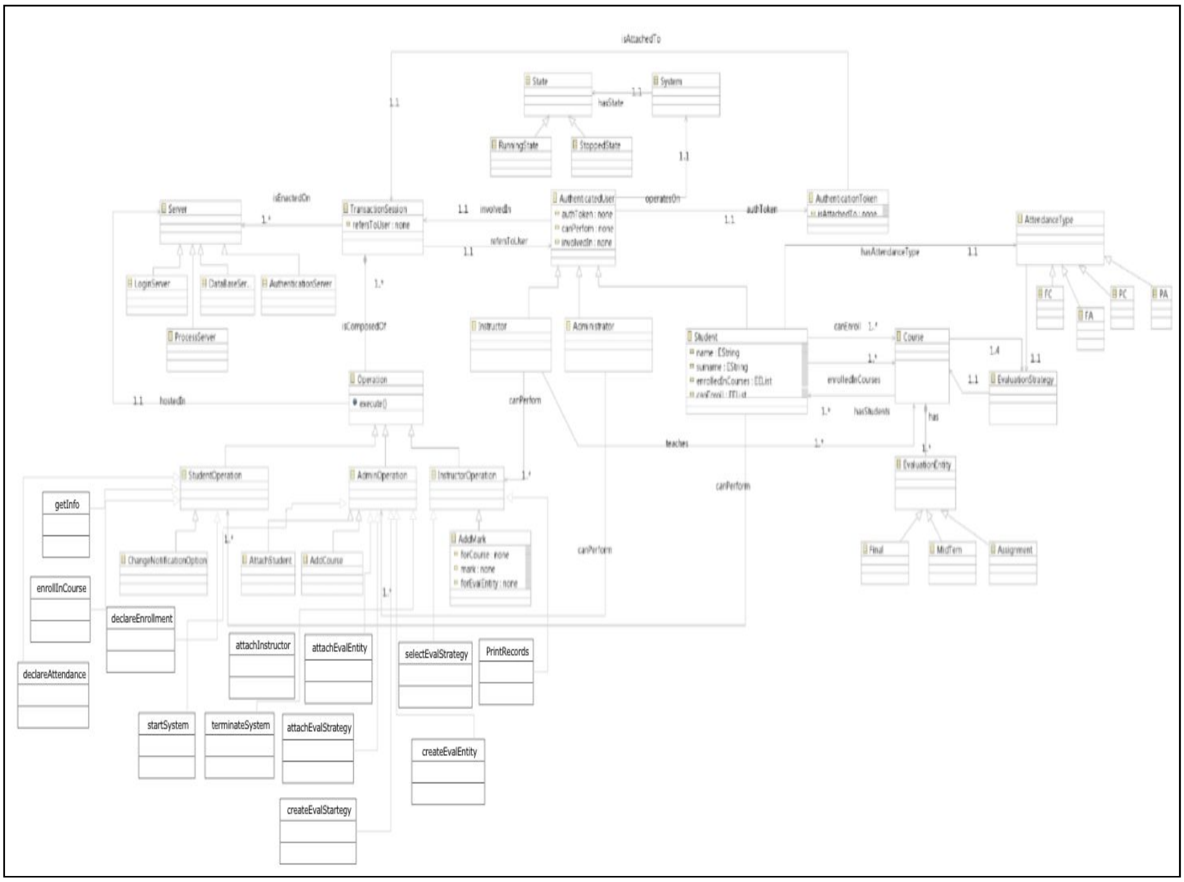


Fig. 11: Domain Model

3.1 Domain Model Class Diagram

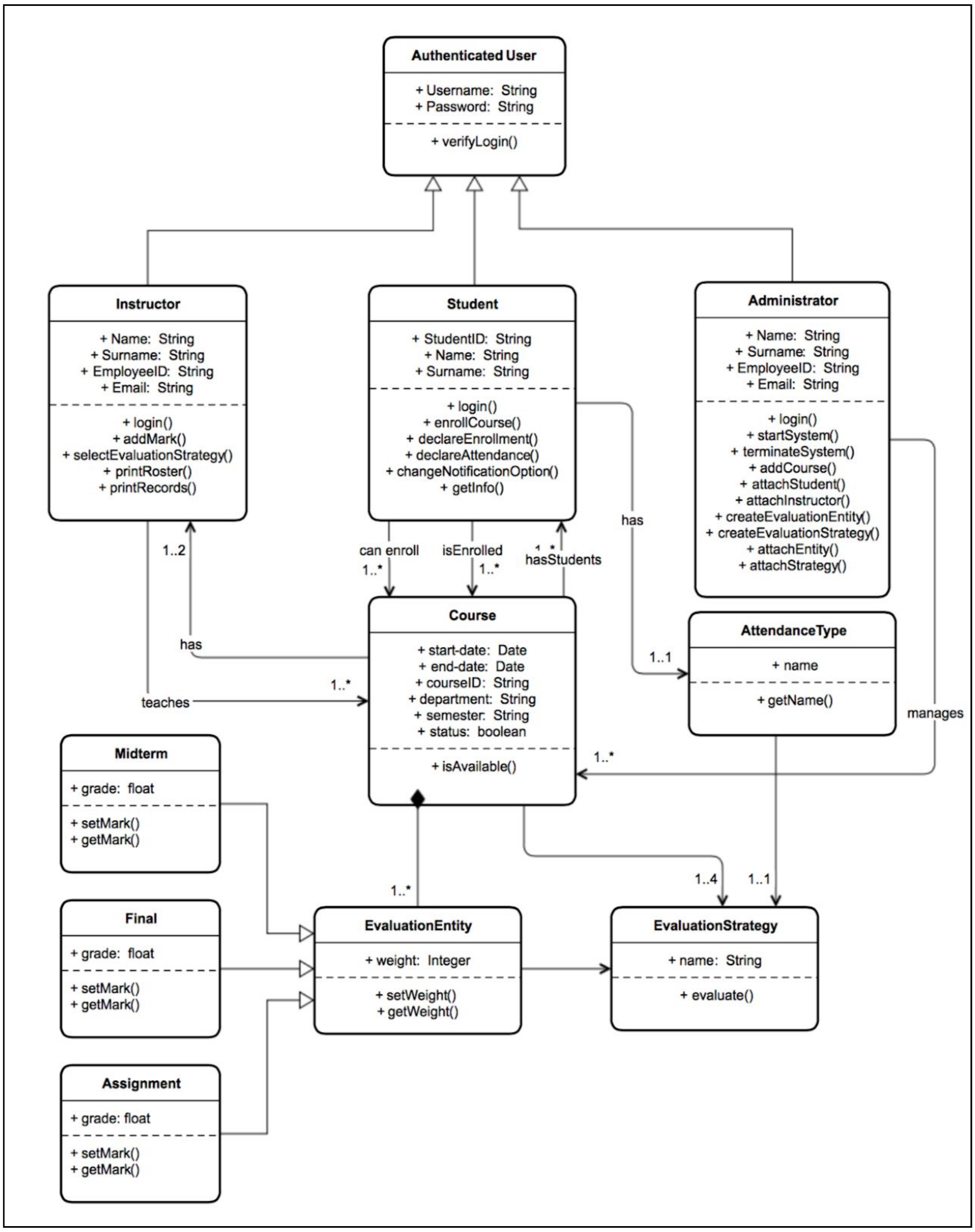


Fig. 12: Domain Model Class Diagram

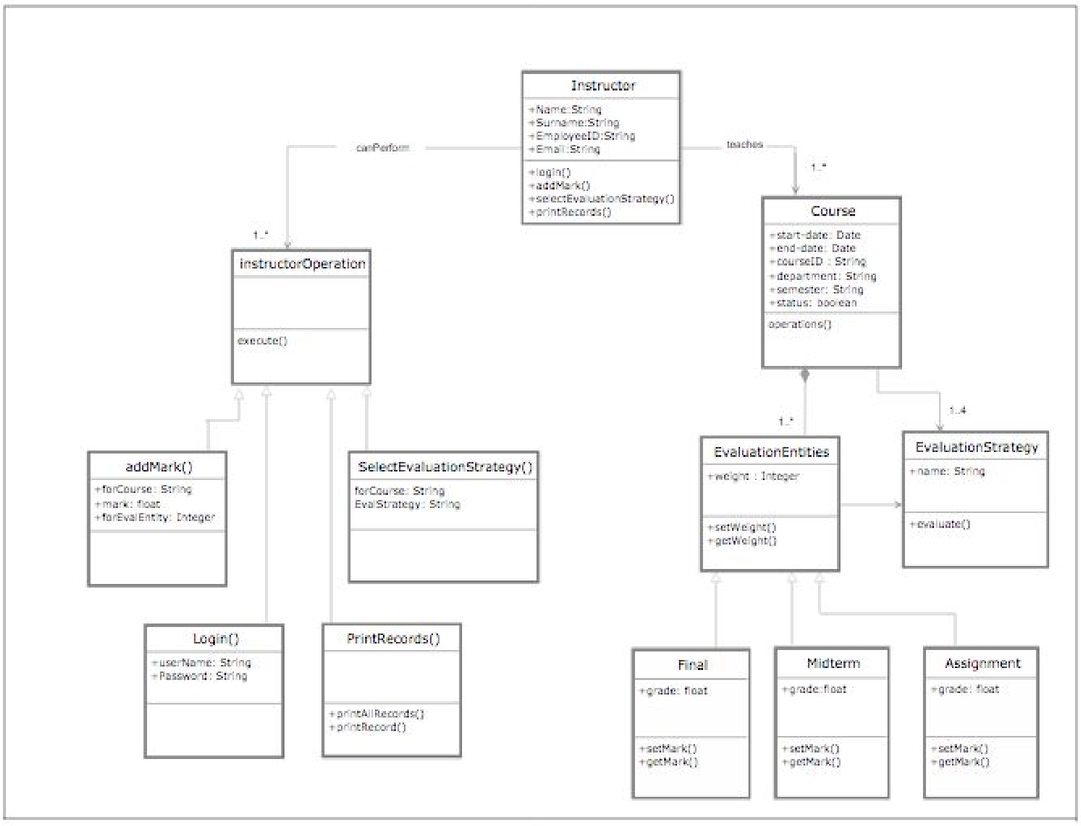


Fig. 13: Domain Model Class Diagram

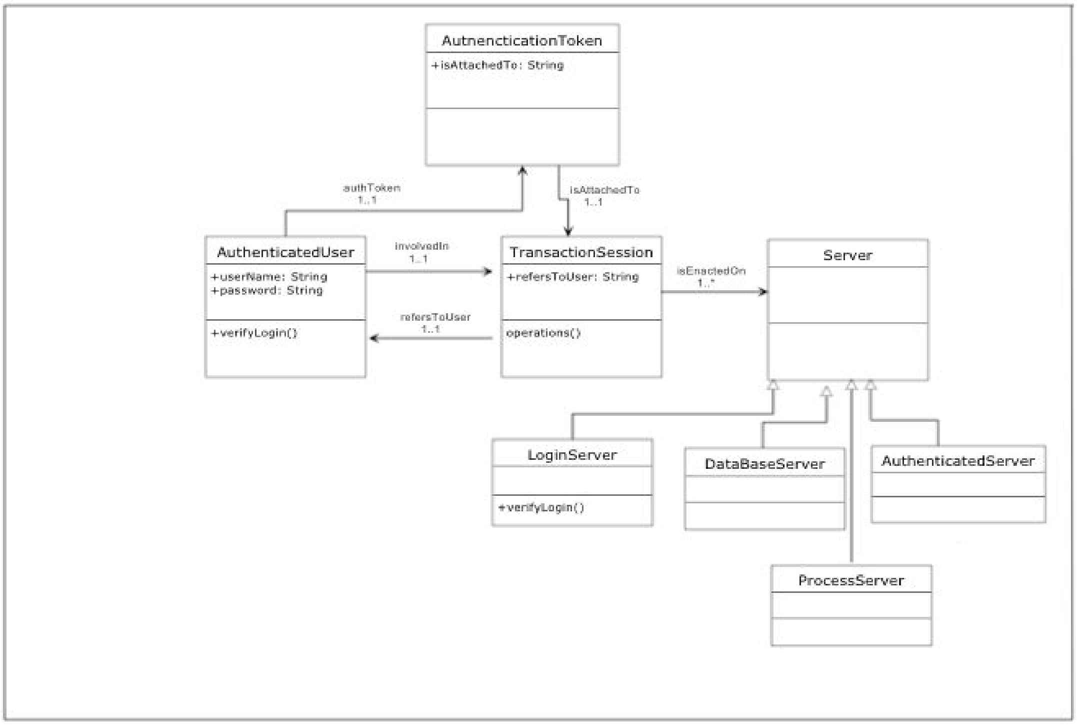


Fig. 14: Domain Model Class Diagram

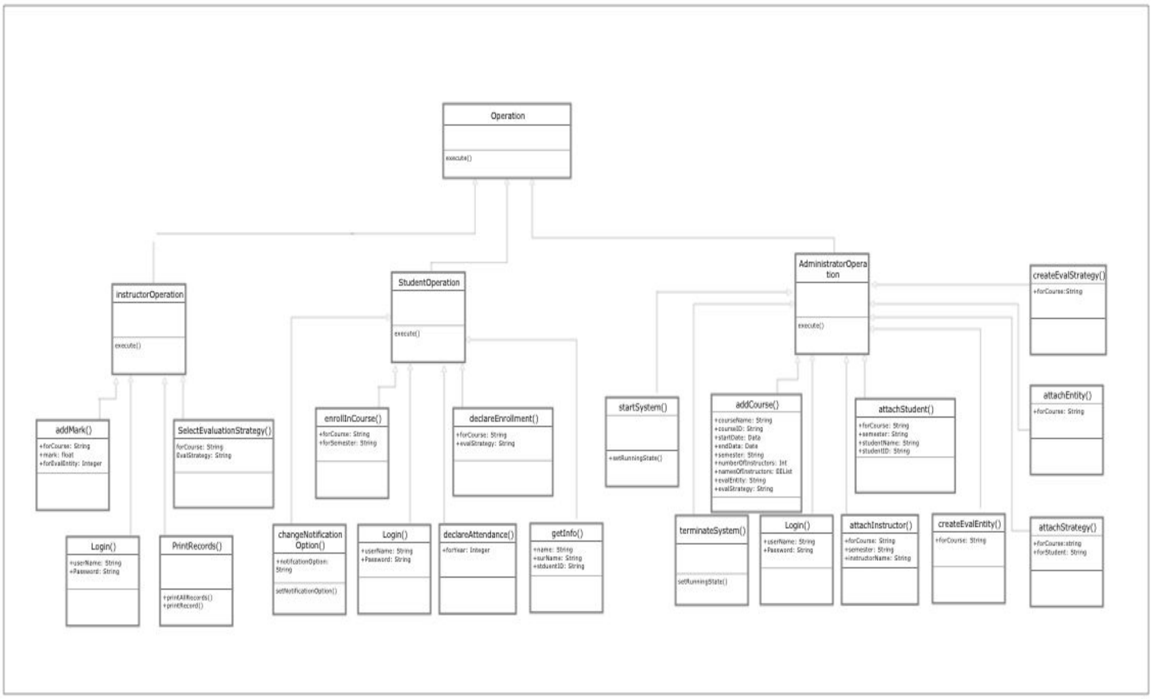


Fig. 15: Domain Model Class Diagram

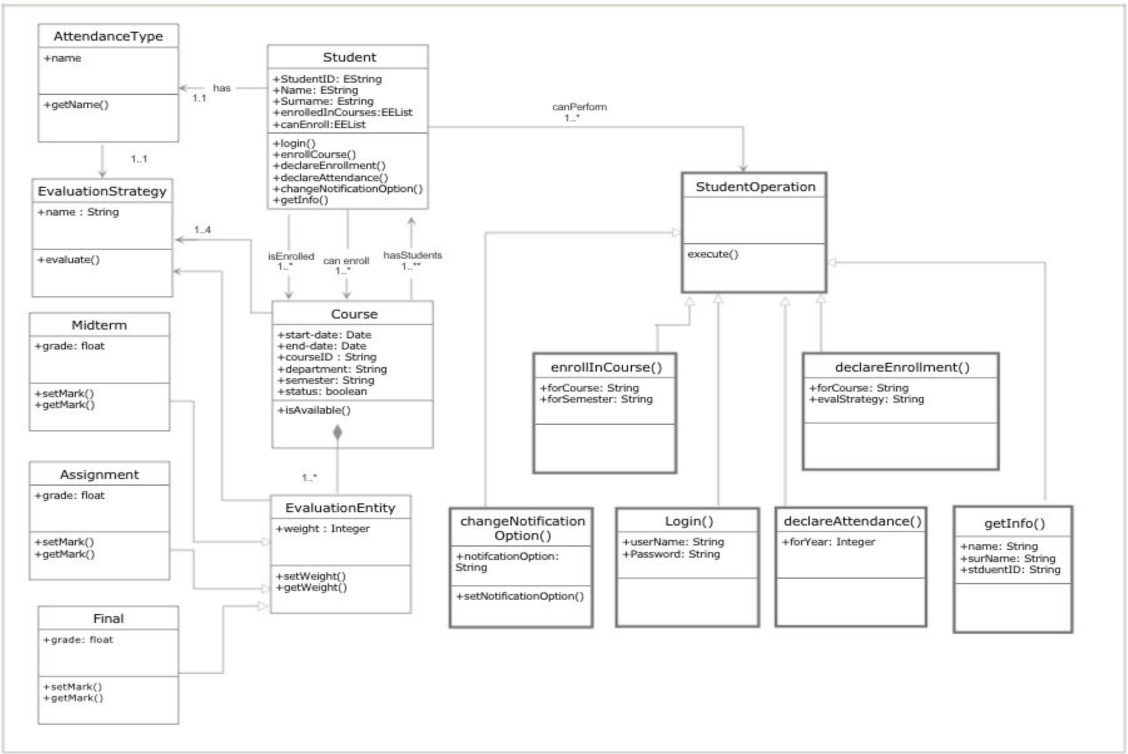


Fig. 16: Domain Model Class Diagram

3.2 Domain Model Class Definitions

3.2.1 Instructor

Description	This class represents an instructor using the system
Attributes	● Login ● Name ● Surname ● EmployeeID ● Email
Responsibilities	1. Keeps the information for a physical actor (Instructor)
Business Rules	*Validation of data is required.* *The entered information must relate to an instructor at the university. (physical entity)*

3.2.2 System Administrator

Description	This class represents a system administrator using the system
Attributes	● Login ● Name ● Surname ● EmployeeID ● Email
Responsibilities	1. Keeps the information for a physical actor (System Administrator)
Business Rules	*Validation of data is required.* *The entered information must relate to a system administrator of university. (physical entity)*

3.2.3 Student

Description	This class represents a student using the system
Attributes	● login ● Name ● Surname ● StudentID
Responsibilities	1. Keeps the information for a physical actor (Student)
Business Rules	*Validation of data is required.* *The entered information must relate to a student of university. (physical entity)*

3.2.4 Course

Description	This class contains the all the relevant information about the course and it's operation
Attributes	● start-date ● end-date ● courseID ● department ● Semester ● Student list ● instructor
Responsibilities	1. Keeps the course information regarding type of course, department and run time 2. Keeps track of all students that are actively enrolled in the course 3. Holds the active instructor for the course
Business Rules	*None.*

3.2.5 Authenticated User

Description	This class contains a user whom's information and role has been validated
Attributes	● login ● password ● permissions
Responsibilities	1. Associates a user with a role and allows them access to the system
Business Rules	*None.*

3.2.6 Authenticated Token

Description	A token included in the request of an authenticated user that can be easily confirmed as valid by the server
Attributes	● key
Responsibilities	1. Contain a valid key recognizable to the server
Business Rules	*None.*

4 Interaction Diagrams

4.1 Sequencing Diagrams

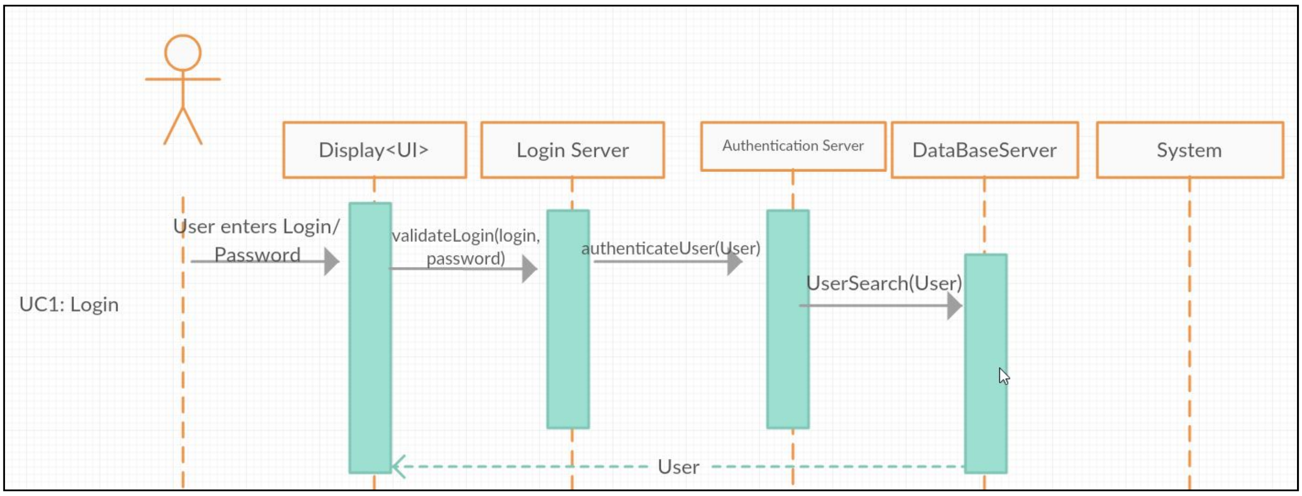


Fig. 17: Sequence Diagram for UC1(Login)

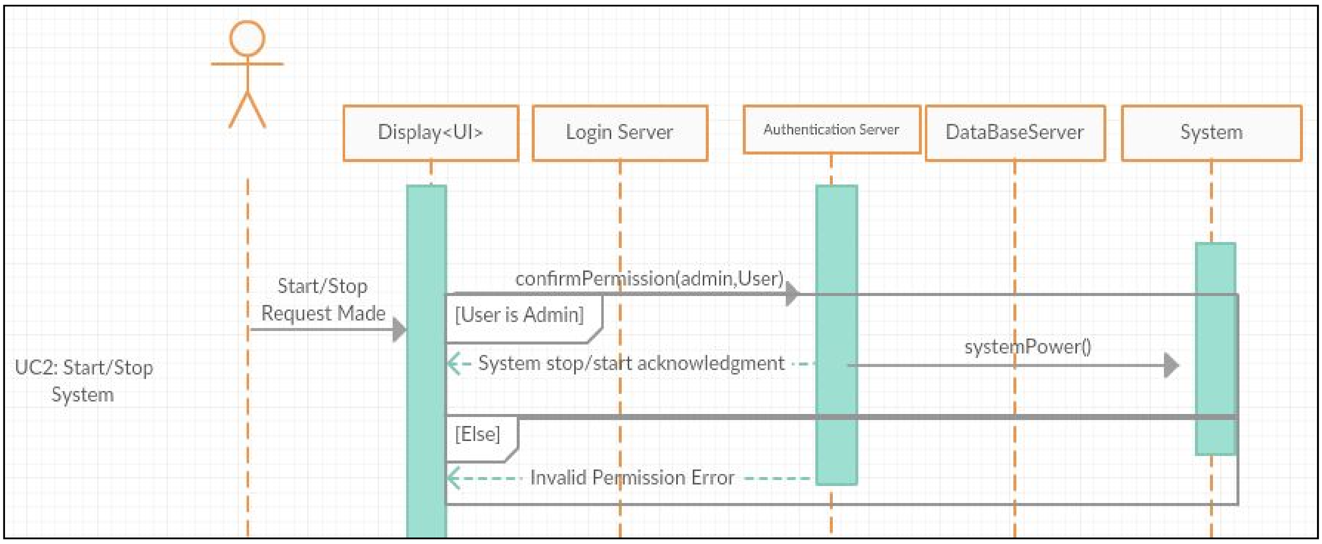


Fig. 18: Sequence Diagram for UC2(Start/Stop System)

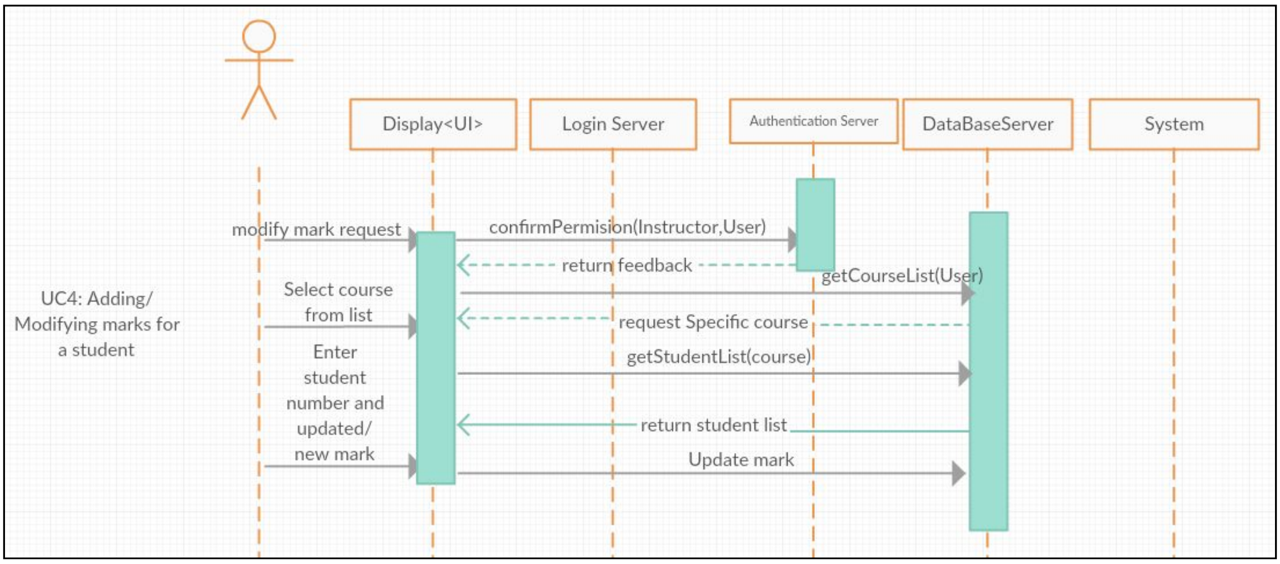


Fig. 19: Sequence Diagram for UC4(adding/modifying marks for a student)

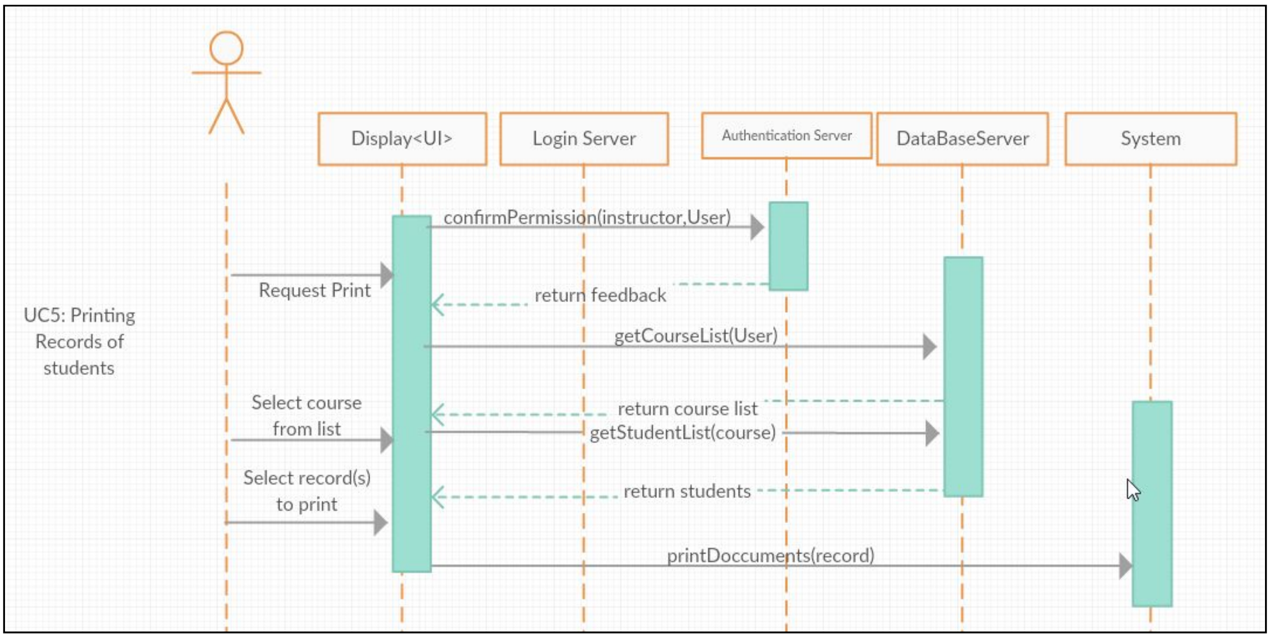


Fig. 20: Sequence Diagram for UC5(Printing records of students)

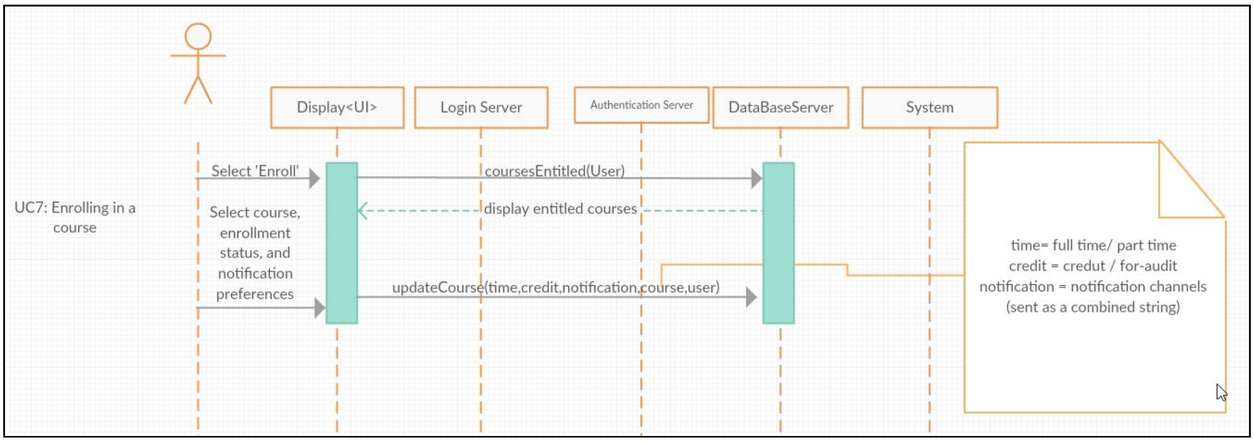

Display<UI>
Login Server
Authentication Server
DataBaseServer
System
UC7: Enrolling in a course
Select 'Enroll'
coursesEntitled(User)
display entitled courses
Select course, enrollment status, and notification preferences
updateCourse(time,credit,notification,course,user)
time= full time/ part time
credit = credut / for-audit
notification = notification channels
(sent as a combined string)

Fig. 21: Sequence Diagram for UC7(Enrolling in a course)

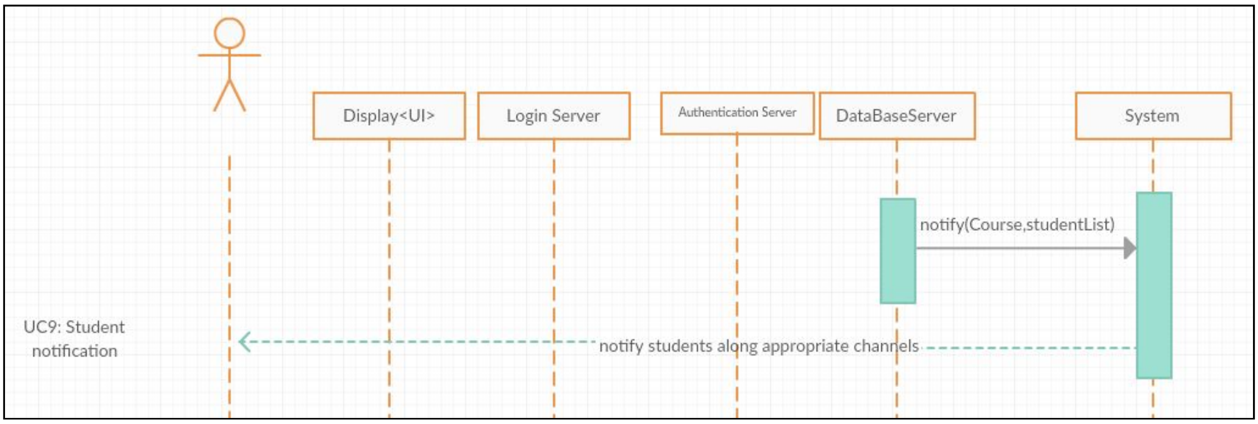

Display<UI>
Login Server
Authentication Server
DataBaseServer
System
notify(Course,studentList)
UC9: Student notification
notify students along appropriate channels

Fig. 22: Sequence Diagram for UC9(Enrolling in a course)

4.2 Collaboration Diagrams

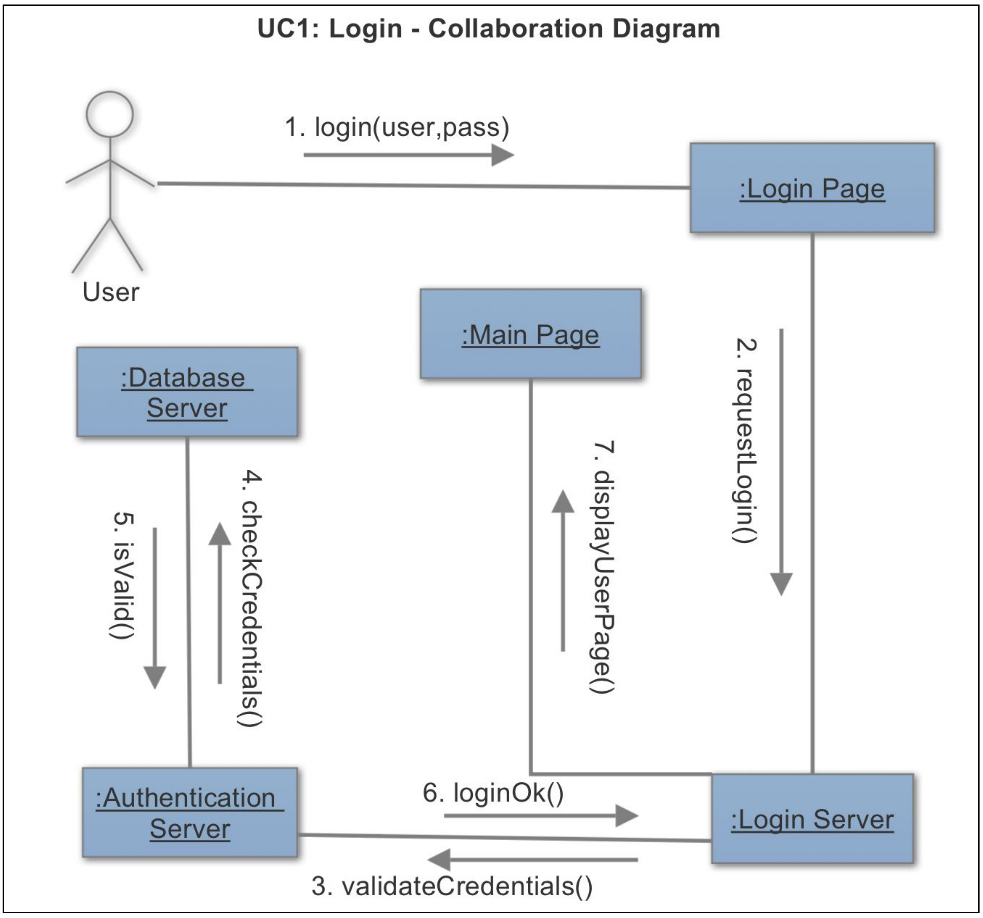


Fig. 23: Collaboration Diagram for Scenario 2.2.1

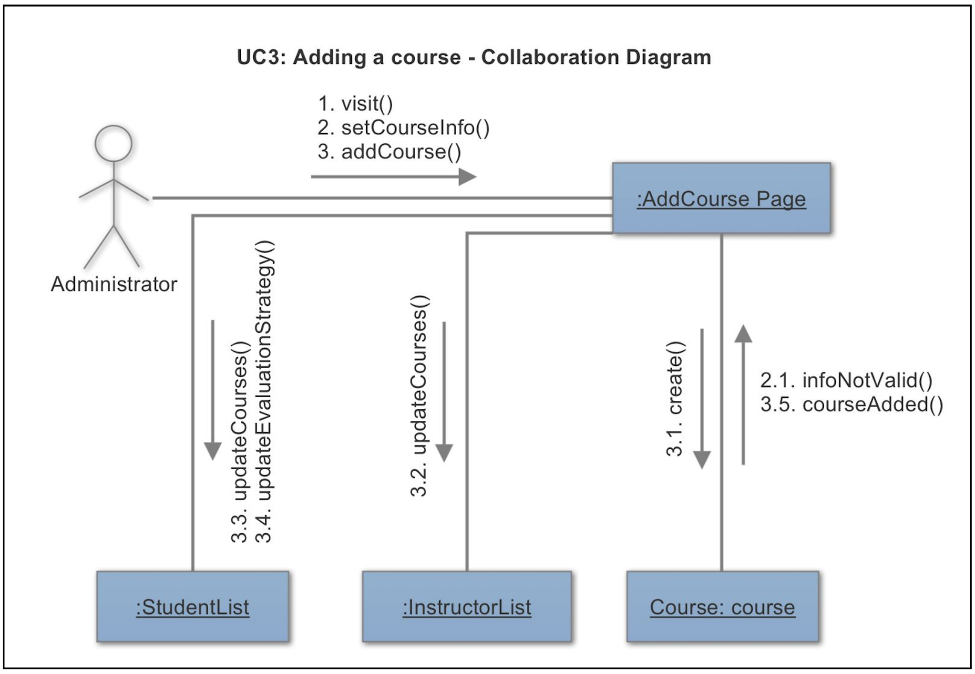


Fig. 24: Collaboration Diagram for Scenario 2.2.3

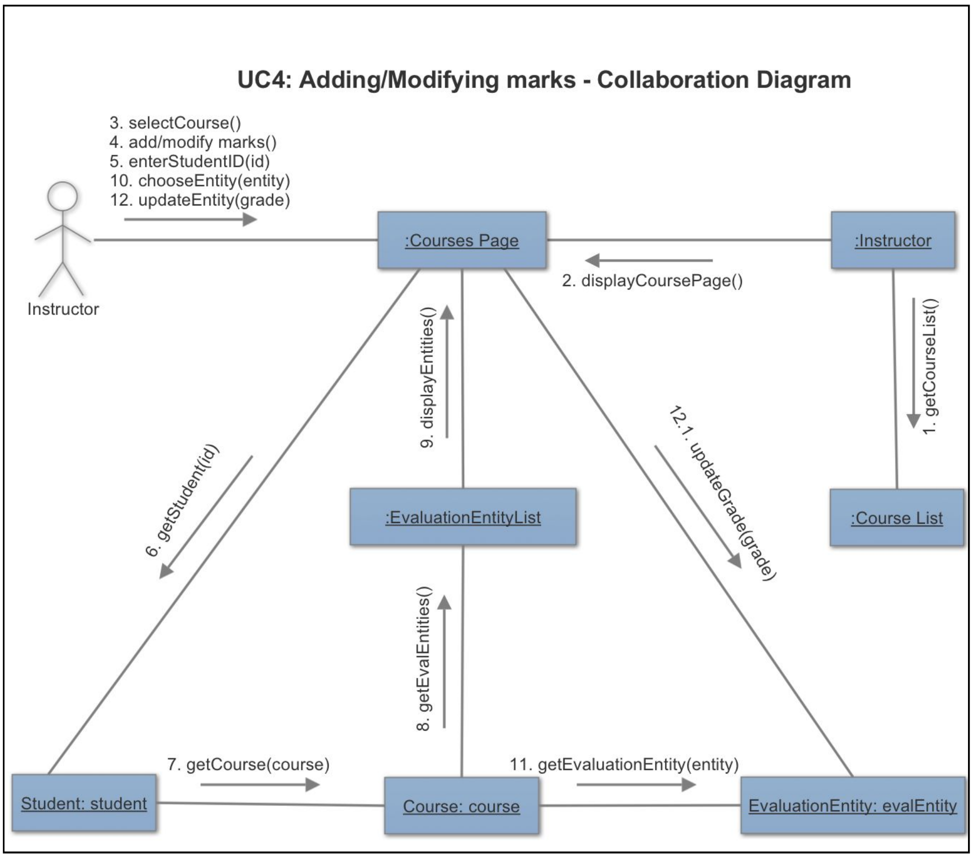


Fig. 25: Collaboration Diagram for Scenario 2.2.4

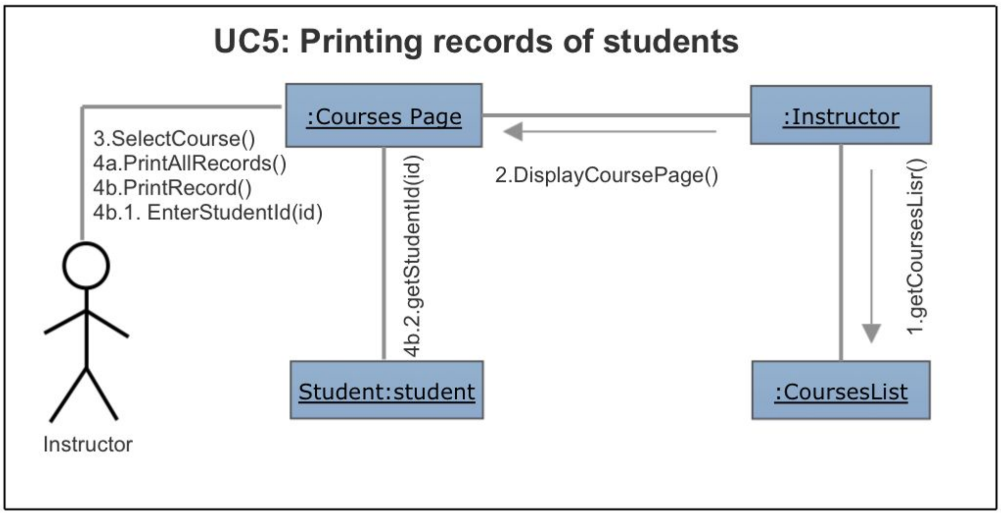


Fig. 26: Collaboration Diagram for Scenario 2.2.5

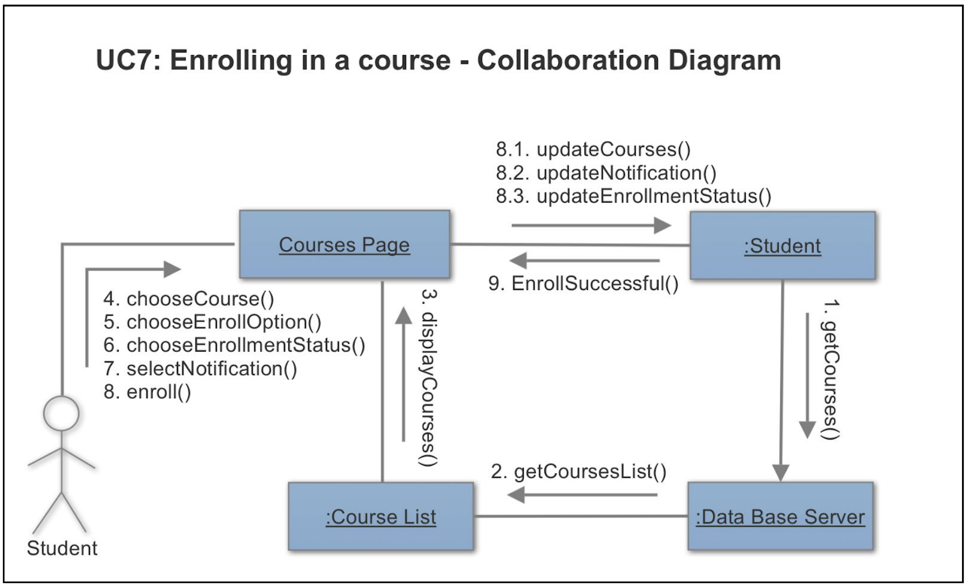


Fig. 27: Collaboration Diagram for Scenario 2.2.7

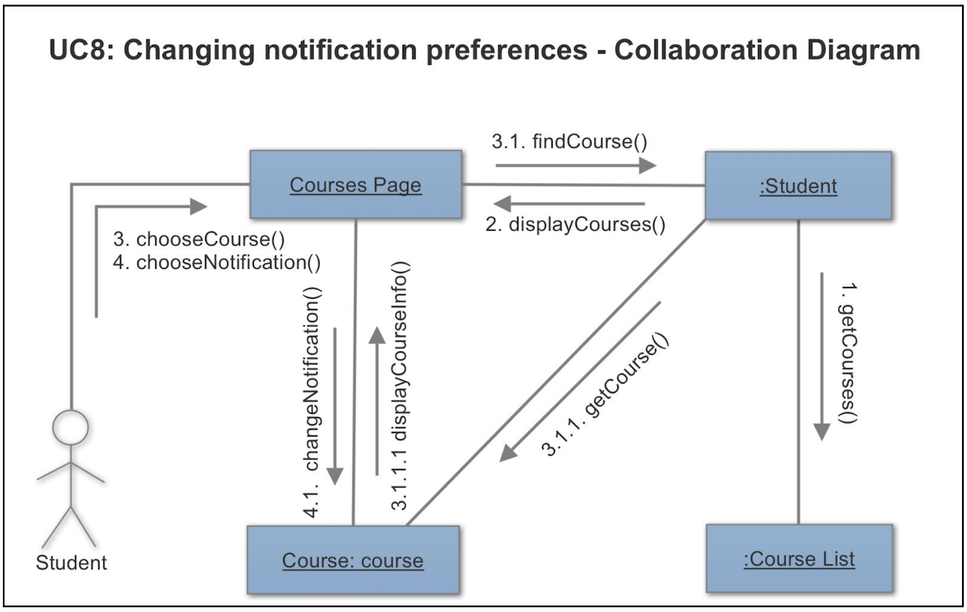


Fig. 28: Collaboration Diagram for Scenario 2.2.8

5 Non-Functional Requirements Specification

The non-functional requirements of the system comprises of utilities, environments and other specifications that are necessary for the smooth operation of the system as a whole. This includes interfaces, development environment, capacity specifications, network and operational parameters.

5.1 Overview

NFR1: The system has to be able to handle at least 100 courses but there should be no fixed limits on the courses to be added
NFR2: The system has to be able to handle at least 600 students per course there should be no fixed limits on the size of a course
NFR3: The roles of the users (Administrator, Instructor, Student) have to be distinct. No user in one role may have permissions that do not belong to his/her role
NFR4: The system cannot allow access unless a user is authenticated
NFR5: The system has to respond in all commands in less than 3 secs. Irrespective of the number of courses and size of classes.

5.2 Enabling Technologies

5.2.1 Target Hardware & Hardware Interfaces

For the system to be accessed and benefitted from, full screens should be used when installed on the operating system that is used to develop the software, and on any other operating system where the system is installed.

5.2.2 Target Development Environment

The system should be developed in a Mac environment and using java. Eclipse Neon.2 Release (4.6.2) will be the Integrated Development Environment (IDE) used for coding purposes.

5.2.3 System Interfaces

At the time being system interfaces that may be required are not identified.

5.3 Capacity Planning

5.3.1 Permanent Storage

For system installation and activity logging, the disk size used will be 10 GB. Most of the space will be occupied by information about users, such as number of students on the system and the number of courses available.

5.4 Network

The system should have connectivity to the Internet with bandwidth sufficient enough to carry digital and written data. Furthermore, the network should possess efficient packet switching protocols that will update data on the system regularly without skips or delays.

5.5 Workstations

The minimum system requirements and configurations for the computers used for development, deployment and execution of the system are:

A hard disk space of 3GB to install Windows, the course management system, and a Java Virtual Machine. A processor speed of 300 MHz and memory of 64 MG is sufficient. A display setting 2560 x 1600 resolution and a 17-bit color palette should be used. The computer should have connectivity to a LAN with bandwidth high enough to carry written data.

5.6 Operational Parameters

5.6.1 Usability

The system should be learnable and usable by any of the three general type users. In other words, administrators, instructors, and students should find this system easy to navigate. The GUI will be designed in such a way that the buttons have meaningful names making it easy to access all the functions of the system.

5.6.2 Reliability

To discuss reliability, recoverability requirements for the new system will be considered.

Recoverability & Backup

The recoverability and backup of the system will be done by maintaining an internal log of transaction sessions to facilitate resolving ambiguities arising from a hardware failure in the middle of a transaction. Entries will be made in the log when the administrator of the system starts up and shuts down the system, for each operation done by any of the users of the system.

Restart

Restarting the system should not be done regularly, and in fact should only be done by the system administrator when the system is turned off.

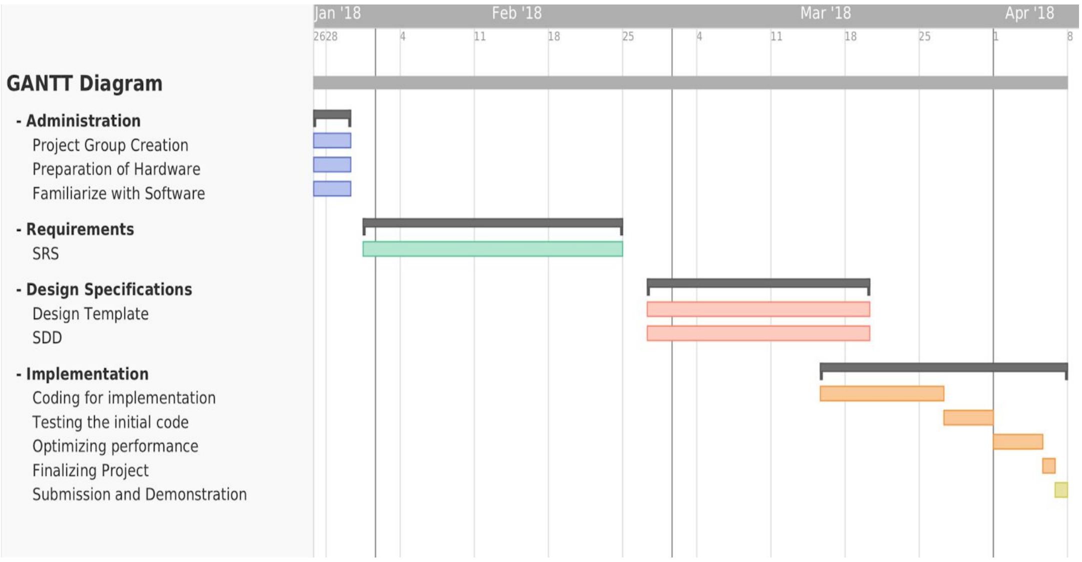
5.6.3 Maintainability

This simple course management system should be easy to analyze, change, and test. In fact, to measure the stability of the system, adding 100 courses to the system and 600 students to a course will be done to show that the limits can be reached with no damage to the system.

5.6.4 Portability

The system should be usable in other operating systems other than the one in which it was created. It should be able to adapt and perform all of its intended functions in other operating systems. Moreover, it should also be easy to install on other operating systems.

6 Activities Plan



7 Domain Dictionary

All terminology used in this document has been described either in this document or in a referred document. Please consult reference documentation for additional details (see Section 1.3).