PROJECT FILE

OBJECT ORIENTED SOFTWARE ENGINEERING (SE-301)

University Registration Management System

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Submitted to:

Dr. Ruchika Malhotra

Submitted By:

~ Aaron Paul Jacob
2K18/SE/003

~ Ajay Kumar
2K18/SE/014

UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

PROBLEM STATEMENT:

Managing registration of students for the next year/semester manually, becomes a very tedious and time-consuming task, with lot of errors and redundancy in data while managing excel sheets and storing data of students separately.

Various other problems that are faced in manual registration process are:

- Redundancy and inconsistency in student data.
- Duplicate data being entered
- Difficulty in keeping track of various parameters such as Total students registered, and Date of registering etc.
- Difficulty in retrieving data of a particular student.
- Maintaining excel sheets and difficulty in locating and accessing them.

To overcome the above challenges, we have decided to develop a computerized University Registration Management System (URMS) which will solve the above stated problems and will have the following features:

- There will be three types of members that can access the system: Admin, Data entry operator, and student
- The administrator can login with authorized credentials and manages the whole system.
- The administrator will be able to see total number students, number of students in particular courses, details of specific student and view the database as a whole. He will also be able to view details of students by selecting a particular year, branch or course.
- The Data Entry Operator will be able to add, modify and remove courses after the successful authorization of credentials.
- The students will be able to login and enter their details such as email, phone number, address etc.
- A user- friendly portal will be given where students can access the list of courses offerings and information about each course such as course id, department, course description, course In-charge, course credits
- The system will not allow duplicated entries for same student id.
- The students will be able to register for maximum of six courses and a minimum of five courses.

INITIAL REQUIREMENT DOCUMENT

Title of the project	University Registration Management System
Stakeholders involved in capturing requirements	Students, Administrator
Techniques used for requirements capturing	Interviewing and brainstorming
Name of the person along with designations	Aaron Paul Jacob, Ajay Kumar
Date	August, 2020
Version	1.0

Consolidated list of Initial requirements:

1. There are three types of members that can access the system:
   i. Administrator
   ii. Student
   iii. DEO
2. A system is to be implemented where the administrator can login with authorized credentials and manage the whole system.
3. The administrator will be able to see total number students, number of students in particular courses, details of specific student and whole database.
4. The administrator will be able to add or remove courses from the list of courses and will decide maximum threshold number of students for a course.
5. The administrator will be able to edit details of any course and number of seats available in that course.
6. A user- friendly portal will be given where students can access the list of courses offerings for the semester and all information about each course such as
   i. Course in charge
   ii. Department
   iii. Course description
   iv. Pre-requisite
   v. Seats availability
7. The students can login with their respective ids.
8. The system will not allow duplicated entries for same student id.
9. The students will be able to register for maximum of six courses and a minimum of five courses.
10. The system will ensure that a course does not have more than a threshold number of students. Once a course is full, subsequent students will not be able to select that course.

Software Requirements Specification

Version 1.0

October 22nd, 2020

University Registration Management System

Aaron Paul Jacob (2K18/SE/003)
Ajay Kumar (2K18/SE/014)
B.Tech. Software Engineering

5th Semester

Table of Contents

Revision History

Name	Date	Version
University Registration Management System	28th October 2020	1.0.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of a University Registration Management System. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

University administrations who want to use the university registration management system for handling all the semester registration related matters.

1.4 Product Scope

University Registration Management System is a system that universities can use to manage their registration for every semester/year, manage course offerings details, updating student details, get an overview of all the students in a course.

The system also aimed at allowing students to register online for different courses from the list of course offerings

1.5 References

- Object-Oriented Software Engineering by Yogesh Singh & Ruchika Malhotra, PHI Learning Pvt. Ltd., 2012
- IEEE Recommended Practice for Software Requirements Specifications – IEEE Std. 830-1998.

1.6 Overview

- The SRS contains an analysis of the requirements necessary to help easy design.
- The overall description provides interface requirements for the Airline Reservation system, product perspective, hardware interfaces software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.
- Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Airline

Reservation system in any fashion.

2. Overall Description

2.1 Product Perspective

The University Registration Management system is designed for all universities and colleges who wishes to easily manage their semester registration process. This software provides options for viewing different courses and their details and provides students with the facility to add a course, remove a course and view the registered courses.

2.2 Product Functions

- A system is to be implemented which can run on the web server.
- The system shall be able to generate Login Id and password to the system operator.
- There are 3 types of users in the University Registration Management System: Student Admin, and DEO
- The administrator shall be able to maintain details of all the courses.
- The administrator shall be able to maintain details of all the students.
- The system shall be able to register for different course.
- The system should give the student the option to add or drop courses.
- A student can only view the details of courses and cannot modify the details.
- The system shall be able to fetch all the student’s details.
- The student shall be able to view all the registered courses.
- The system shall be able to update the profile of the students.

2.3 Operating Environment

The URMS is designed to be accessed through most of the versions windows operating systems such as:

- Windows 7
- Windows 8
- Windows 10

And also, through MacOS and Linux

2.4 Design and Implementation Constraints

At the time of registration, each student is provided a student ID and password that must be used for logging the system for registration. Hence the student is required to remember or store these login credentials carefully.

2.5 User Characteristics

1. The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus, the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user
2. The Product is absolutely user friendly, so the intended users can be naïve users.
3. The product does not expect the user to possess any technical background. Any person who knows how to use the mouse and the keyboard can successfully use this product.

2.6 Assumptions and Dependencies

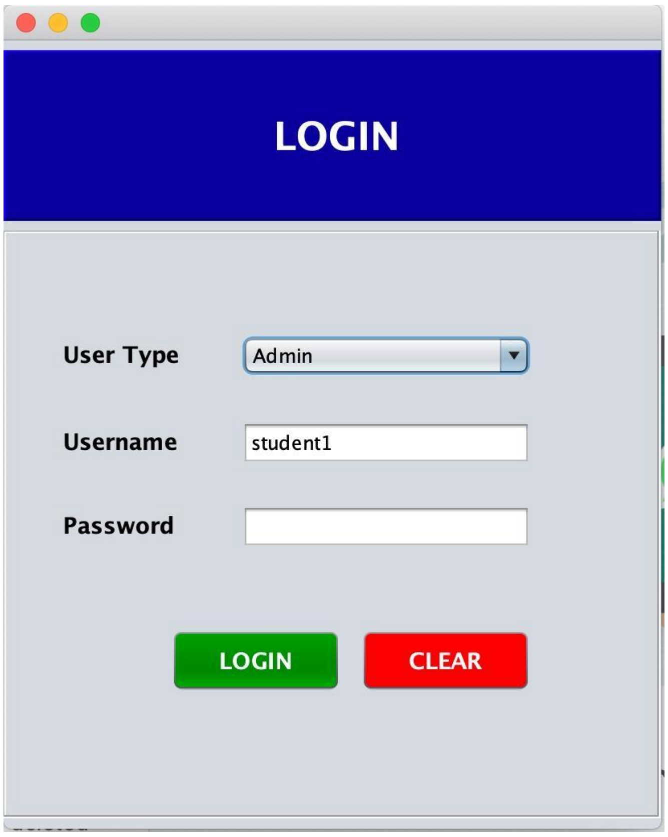
The requirements stated in the SRS could be affected by the following factors:

- One major dependency that the project might face is the changes that need to be incorporated with the changes in the University policies regarding registrations. As the policies changes the system needs to be updated with the same immediately. So, this should be changed as and when required by the developer.
- At this stage no quantitative measures are imposed on the software in terms of speed and memory although it is implied that all functions will be optimized with respect to speed and memory.

It is furthermore assumed that the scope of the package will increase considerably in the future.

3. External Interface Requirements

3.1 Login interface



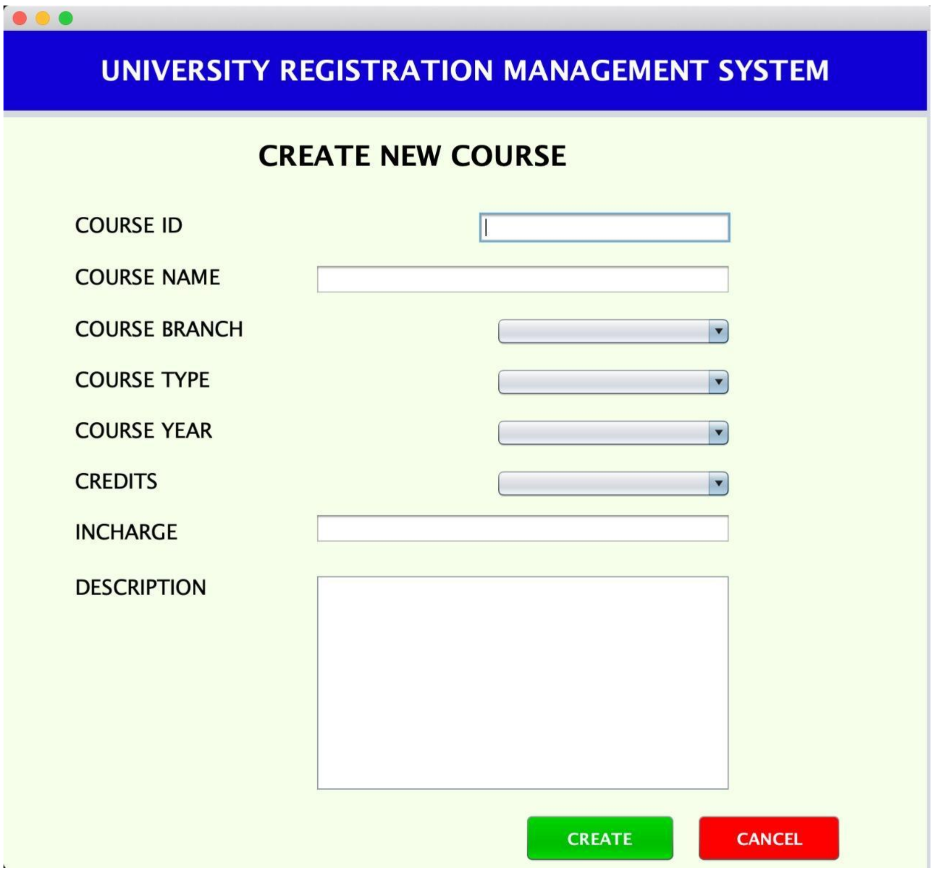
LOGIN

User Type: Admin

Username: student1

Password:

3.2 Create new course



UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

CREATE NEW COURSE

COURSE ID

COURSE NAME

COURSE BRANCH

COURSE TYPE

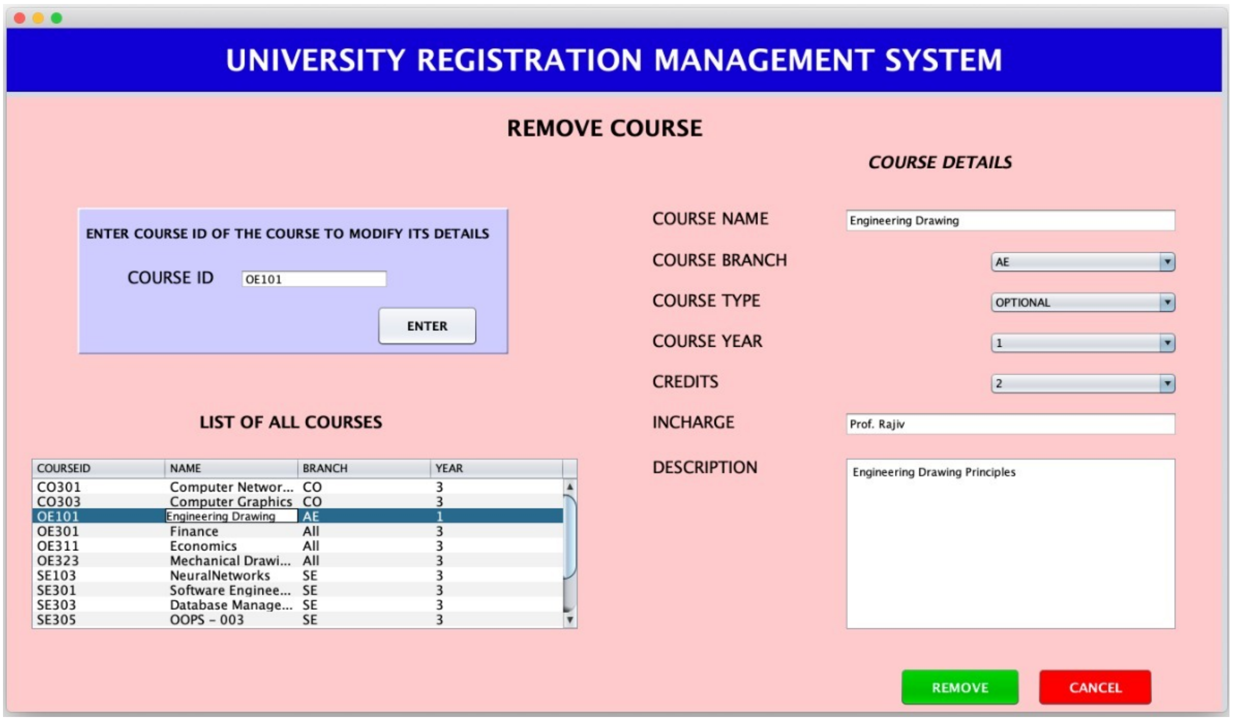
COURSE YEAR

CREDITS

INCHARGE

DESCRIPTION

3.3 Remove course



UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

REMOVE COURSE

ENTER COURSE ID OF THE COURSE TO MODIFY ITS DETAILS

COURSE ID: OE101

ENTER

LIST OF ALL COURSES

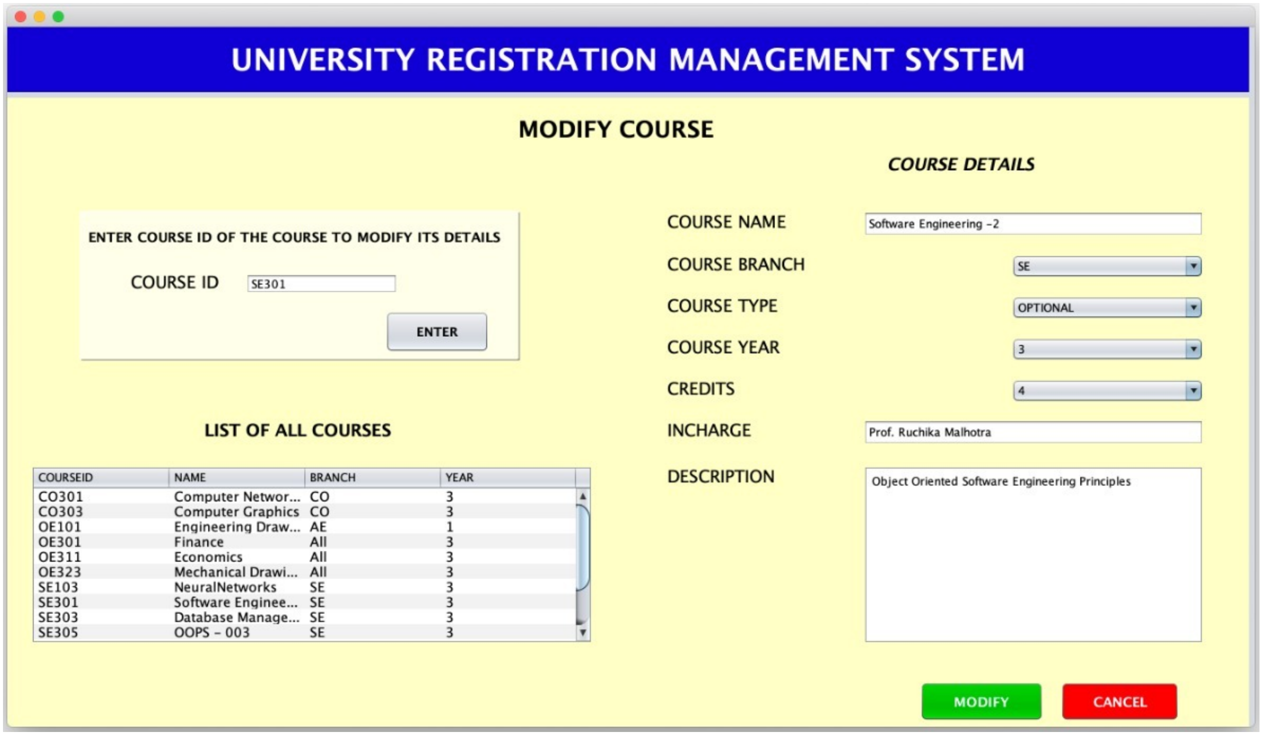
COURSEID	NAME	BRANCH	YEAR
CO301	Computer Networ...	CO	3
CO303	Computer Graphics	CO	3
OE101	Engineering Drawing	AE	1
OE301	Finance	All	3
OE311	Economics	All	3
OE323	Mechanical Drawi...	All	3
SE103	NeuralNetworks	SE	3
SE301	Software Enginee...	SE	3
SE303	Database Manage...	SE	3
SE305	OOPS – 003	SE	3

COURSE DETAILS

COURSE NAME	Engineering Drawing
COURSE BRANCH	AE
COURSE TYPE	OPTIONAL
COURSE YEAR	1
CREDITS	2
INCHARGE	Prof. Rajiv
DESCRIPTION	Engineering Drawing Principles

REMOVE CANCEL

3.4 Modify course



UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

MODIFY COURSE

ENTER COURSE ID OF THE COURSE TO MODIFY ITS DETAILS

COURSE ID: SE301

ENTER

LIST OF ALL COURSES

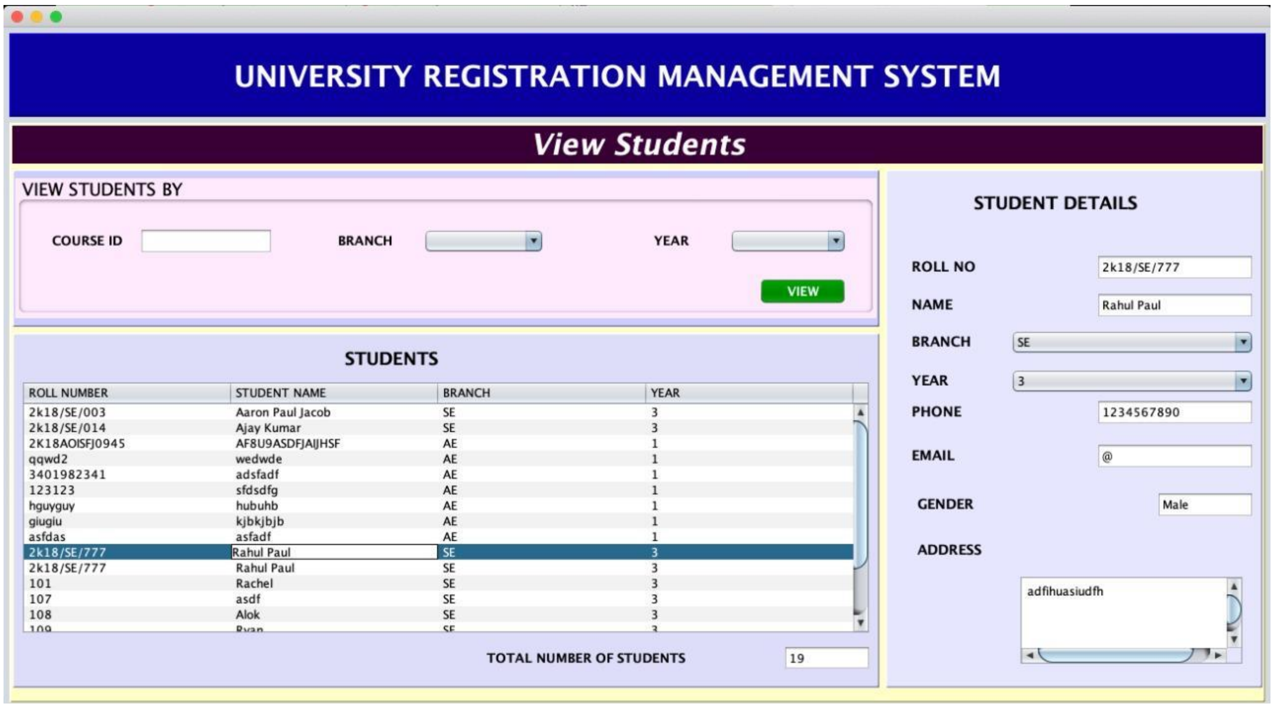
COURSEID	NAME	BRANCH	YEAR
CO301	Computer Networ...	CO	3
CO303	Computer Graphics	CO	3
OE101	Engineering Draw...	AE	1
OE301	Finance	All	3
OE311	Economics	All	3
OE323	Mechanical Drawi...	All	3
SE103	NeuralNetworks	SE	3
SE301	Software Enginee...	SE	3
SE303	Database Manage...	SE	3
SE305	OOPS – 003	SE	3

COURSE DETAILS

COURSE NAME	Software Engineering -2
COURSE BRANCH	SE
COURSE TYPE	OPTIONAL
COURSE YEAR	3
CREDITS	4
INCHARGE	Prof. Ruchika Malhotra
DESCRIPTION	Object Oriented Software Engineering Principles

MODIFY CANCEL

3.5 View all student



UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

View Students

VIEW STUDENTS BY

COURSE ID

BRANCH

YEAR

VIEW

STUDENTS

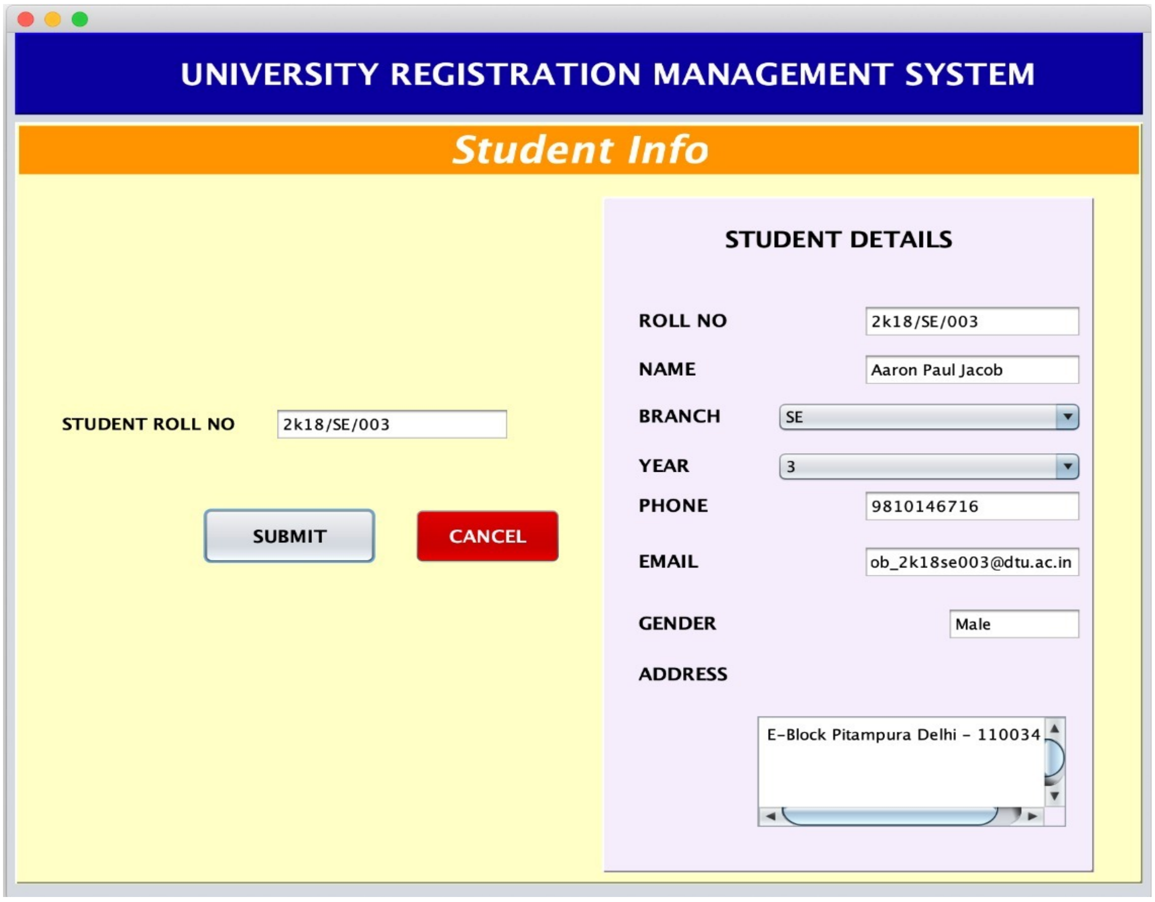
ROLL NUMBER	STUDENT NAME	BRANCH	YEAR
2k18/SE/003	Aaron Paul Jacob	SE	3
2k18/SE/014	Ajay Kumar	SE	3
2K18AOISFJ0945	AF8U9ASDFJAIJHSF	AE	1
qqwd2	wedwde	AE	1
3401982341	adsfadf	AE	1
123123	sfdsdfg	AE	1
hguyguy	hubuhb	AE	1
giugiu	kjbkjbjb	AE	1
asfdas	asfadf	AE	1
2k18/SE/777	Rahul Paul	SE	3
2k18/SE/777	Rahul Paul	SE	3
101	Rachel	SE	3
107	asdf	SE	3
108	Alok	SE	3
109	Ryan	SE	3

TOTAL NUMBER OF STUDENTS 19

STUDENT DETAILS

ROLL NO	2k18/SE/777
NAME	Rahul Paul
BRANCH	SE
YEAR	3
PHONE	1234567890
EMAIL	@
GENDER	Male
ADDRESS	adfihuasiudfh

3.6 View student



UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

Student Info

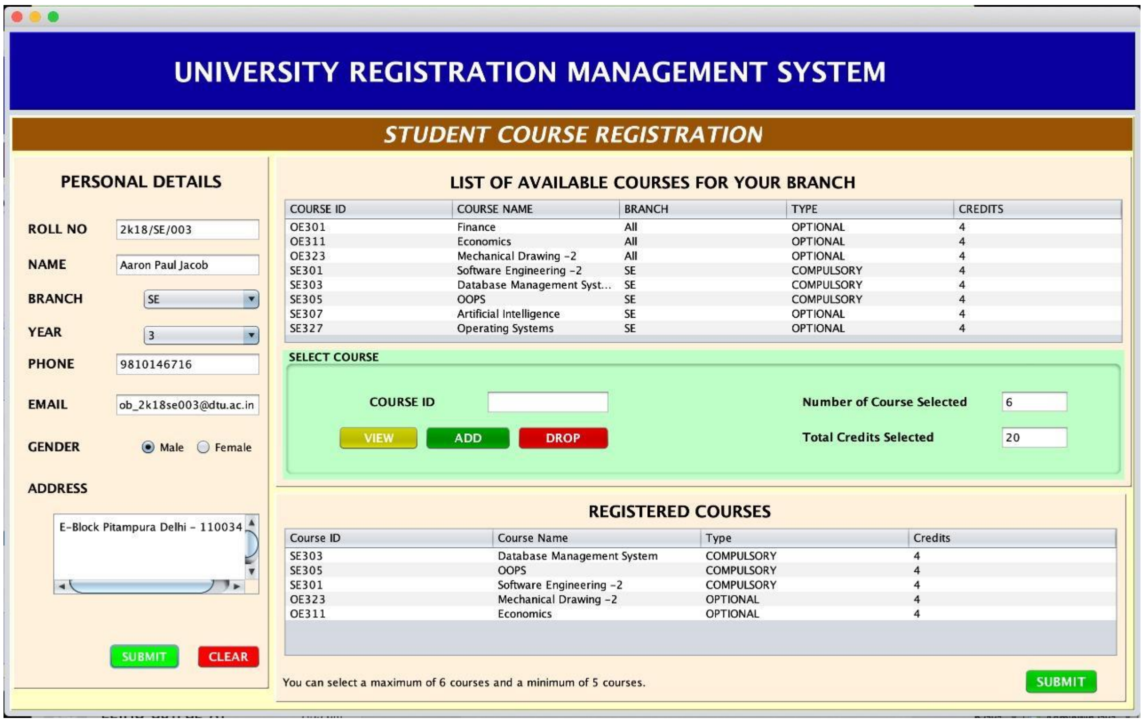
STUDENT ROLL NO 2k18/SE/003

SUBMIT CANCEL

STUDENT DETAILS

ROLL NO	2k18/SE/003
NAME	Aaron Paul Jacob
BRANCH	SE
YEAR	3
PHONE	9810146716
EMAIL	ob_2k18se003@dtu.ac.in
GENDER	Male
ADDRESS	E-Block Pitampura Delhi - 110034

3.7 Student course registration


UNIVERSITY REGISTRATION MANAGEMENT SYSTEM

STUDENT COURSE REGISTRATION

PERSONAL DETAILS

ROLL NO	2k18/SE/003
NAME	Aaron Paul Jacob
BRANCH	SE
YEAR	3
PHONE	9810146716
EMAIL	ob_2k18se003@dtu.ac.in
GENDER	[✓ Male] Female
ADDRESS	E-Block Pitampura Delhi - 110034

LIST OF AVAILABLE COURSES FOR YOUR BRANCH

COURSE ID	COURSE NAME	BRANCH	TYPE	CREDITS
OE301	Finance	All	OPTIONAL	4
OE311	Economics	All	OPTIONAL	4
OE323	Mechanical Drawing -2	All	OPTIONAL	4
SE301	Software Engineering -2	SE	COMPULSORY	4
SE303	Database Management Syst...	SE	COMPULSORY	4
SE305	OOPS	SE	COMPULSORY	4
SE307	Artificial Intelligence	SE	OPTIONAL	4
SE327	Operating Systems	SE	OPTIONAL	4

SELECT COURSE

COURSE ID	
Number of Course Selected	6
Total Credits Selected	20

REGISTERED COURSES

Course ID	Course Name	Type	Credits
SE303	Database Management System	COMPULSORY	4
SE305	OOPS	COMPULSORY	4
SE301	Software Engineering -2	COMPULSORY	4
OE323	Mechanical Drawing -2	OPTIONAL	4
OE311	Economics	OPTIONAL	4

You can select a maximum of 6 courses and a minimum of 5 courses.

3.8 Hardware Interfaces

The minimum hardware requirements of Air Ticket Reservation System are a 1 Gigahertz CPU and 2 Gigabytes of RAM. The system must interface with the standard output device, keyboard and mouse to interact with this software.

3.9 Software Interfaces

URMS is a desktop application and designed to be accessed by most of the versions of windows operating systems.

URMS can relate to a PostgreSQL or MySQL database but the current development is on MySQL

3.10 Communications Interfaces

URMS might require an internet connection to install new updates

4. System Features

This section demonstrates University Registration Management system’s most prominent features and explains how they can be used and the results they will give back to the user.

4.1 Login

The system will control user login based on the mode of login selected by the user.

Introduction: This use case allows student/administrator to access the functionalities of the system
Actors: Student, Admin, DEO
Pre-Condition: The user must be having login credentials and the system must be connect to network.
Post Condition: If the use case is successful, the user is logged in to the system
Event flow Basic Flow: i. System displays login page ii. The user enters his/her login credentials iii. Authentication is performed iv. If the credentials are correct the system redirect to the dashboard
Alternative Flow 1: Invalid login credential if the login credentials are not correct an error message is displayed
Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.

4.2 Create New Course:

Introduction: This use case allows user to create new courses
Actors: Admin, DEO
Pre-Condition: The user must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, a new course is added to course offerings
Event flow Basic Flow: i. The system displays a course creation page. ii. Name of the course to be created is entered. iii. Course description and pre-requisite are filled. iv. Faculty in charge along with maximum number of seats is entered. v. After entering all details of course, the course is saved to database.
Alternative Flow 1: Course already exist If the course with the same name is already exists, then system will ask user to either changed the name of new course to be added or first removed existing one from the list.

Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: None
Associated use case: Login

4.3 Remove Course

Introduction: This use case allows user to remove courses from the list of course offerings
Actors: Admin, DEO
Pre-Condition: The user must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the selected course is removed from the course offerings
Event flow Basic Flow: i. The system displays list of all courses. ii. Admin select the course to be removed iii. The updated list of courses is saved to database.
Alternative Flow 2: Invalid course ID If the course ID entered is invalid, the system will display message that course doesn’t exist.
Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: None
Associated use case: Login

4.4 Modify Course

Introduction: This use case allows user to modify/change existing courses details
Actors: Admin, DEO
Pre-Condition: The admin must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the details of the course are updated
Event flow Basic Flow: i. The system displays list of all courses. ii. Admin select the course to be modified. iii. Admin can change the course description, pre-requisite, in charge and seats count as per needs. iv. Updated course is saved to database.
Alternative Flow 1: Similar course exist If the modified course has name that already exist, the system display message that same course already exists

Alternative Flow 2: Invalid Course ID If an invalid course ID is entered, the system display message that course doesn’t exists.
Alternative Flow 3: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: None
Associated use case: Login, Display courses

4.5 View Student

Introduction: This use case allow admin to view the details of any student.
Actors: Admin
Pre-Condition: The admin must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the admin will be able to view all information of the desired student
Event flow Basic Flow: i. The system displays view student detail page. ii. The admin enters the student’s id of that student whose details to be viewed. iii. The system displays all the details of student along with registered courses.
Alternative Flow 1: Invalid student ID If the entered student id is invalid, then system will display message stating enter valid student ID.
Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: None
Associated use case: Login

4.6 View all student

Introduction: This use case allow admin to view all student details collectively.
Actors: Admin
Pre-Condition: The admin must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the admin will be able to view collective information of all students
Event flow Basic Flow: i. The system displays all student detail page. ii. The admin selects whether the number of students in a course to viewed or the total number registered viewed. iii. The system displays all the desired details of students.

Alternative Flow 1: Invalid Course ID If the entered course ID is invalid, the system will display message to enter valid course ID
Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: None
Associated use case: Login

4.7 Add Course

Introduction: This use case allows Student to add courses from the list of course offerings.
Actors: Student, Admin
Pre-Condition: The student must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the selected course is added to enrolled courses list
Event flow Basic Flow: i. The system displays list of available course offerings for the semester ii. The Student selects desired course from the list of available course offerings. iii. The course is added to enrolled courses list iv. The student information along with enrolled courses is saved to database.
Alternative Flow 1: Course is full If the course to be added has no seats available, then student cannot add that course
Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: none
Associated use case: Login, Display courses

4.8 Drop Course

Introduction: This use case allows Student to drop/remove courses from the selected course offerings.
Actors: Student, Admin
Pre-Condition: The user must be logged onto the system before the use case begins.
Post Condition: If the use case is successful, the selected course is removed from the enrolled courses list
Event flow Basic Flow: i. The system displays list of selected courses ii. The Student selects the course to be removed from the list of registered courses. iii. The student information along with updated enrolled courses list is saved to database.
Alternative Flow 1: User exits This allows the user to exit at any time during the use case. The use case ends.

Special requirement: None
Associated use case: Login, Display courses

4.9 View Registered course

Introduction: This use case allows Student to view registered courses.
Actors: Student, Admin
Pre-Condition: The user must be logged onto the system before the use case begins.
Post Condition: None
Event flow Basic Flow: i. The system displays list of enrolled course offerings for the semester
Alternative Flow 1: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: none
Associated use case: Login

4.10 Display Course

Introduction: This use case allows to view the full list of course offerings for the semester
Actors: Student, Admin
Pre-Condition: The user must be logged onto the system before the use case begins.
Post Condition: None
Event flow Basic Flow: i. The system displays list of all course offerings for the semester
Alternative Flow 1: User exits This allows the user to exit at any time during the use case. The use case ends.
Special requirement: none
Associated use case: Login

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The minimum hardware requirements of University Registration Management System are a 1 Gigahertz CPU and 2 Gigabytes of RAM. The system must interface with the standard output device, keyboard and mouse to interact with this software.

5.2 Safety Requirements

To ensure that no one of URMS's users loses any data while using URMS (due to a crash or a bug of some kind) the developer team updates URMS regularly.

5.3 Security Requirements

URMS has two levels of users each with different privileges i.e.

- Administrator who is the super user and can perform most of the functions in the system such as creating course, remove course, modify course details, view students list, view student details etc.
- Student who can Register for different courses, view course details etc.

5.4 Software Quality Attributes

URMS provides the users with very simple features. Due to its well designed and easy to use interface it can be used by any users. However, users must already have a basic knowledge of using a computer.

Appendix A: Glossary

SRS – Software Requirements Specification

URMS – University Registration Management System

IEEE – Institute of Electrical and Electronics Engineers

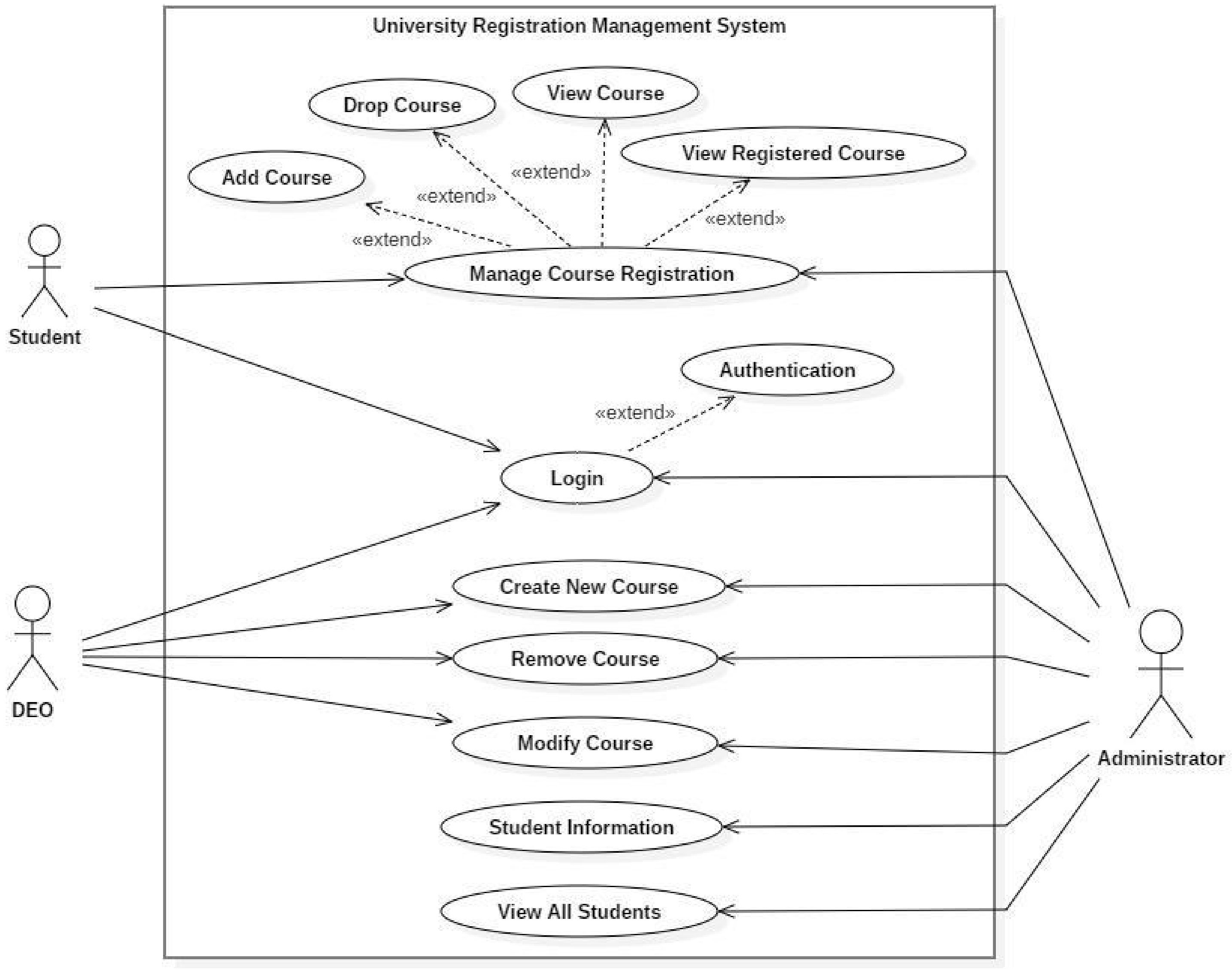
CPU – Central Processing Unit

RAM – Random Access Memory

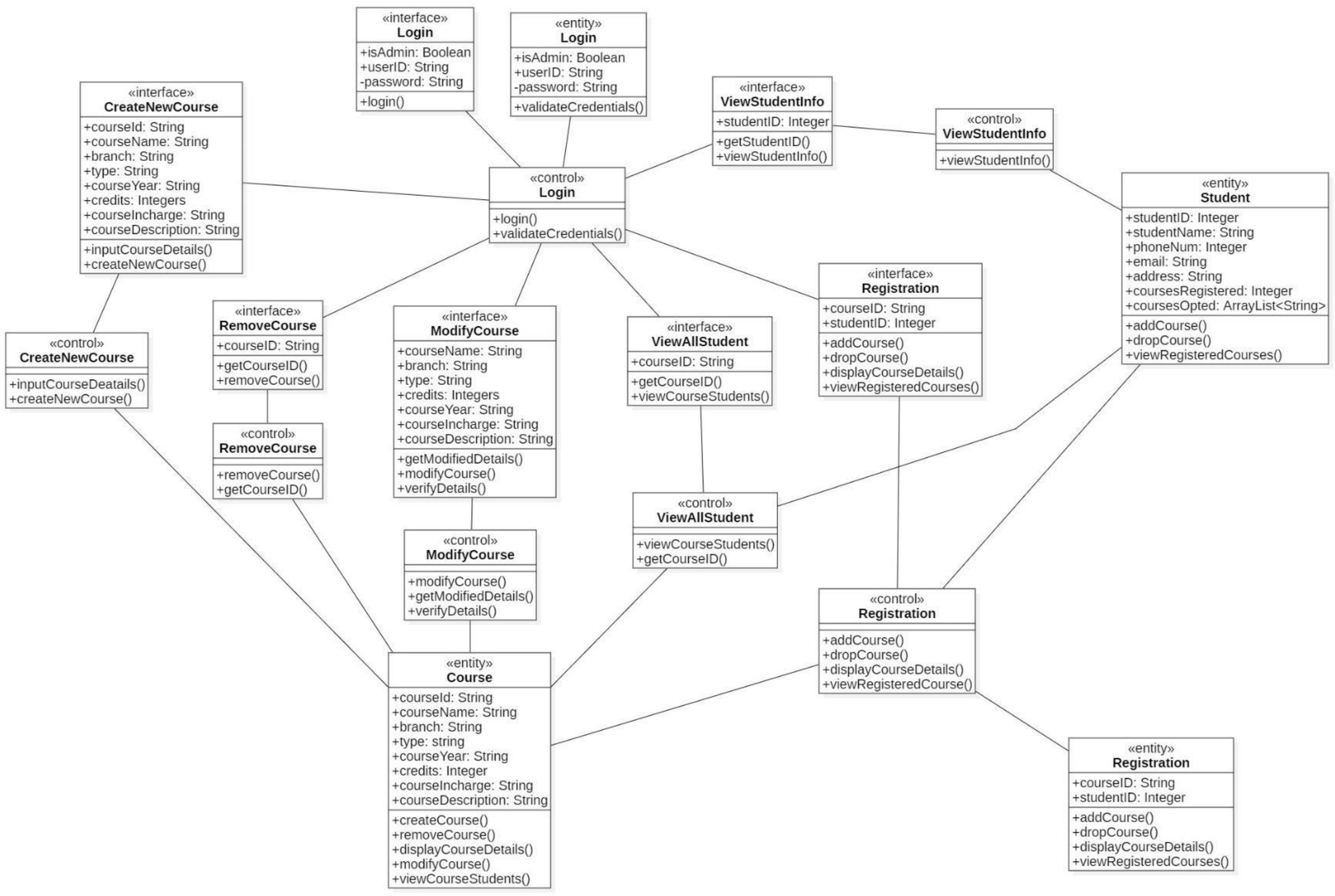
Appendix B: Analysis

Models

Use case Diagram

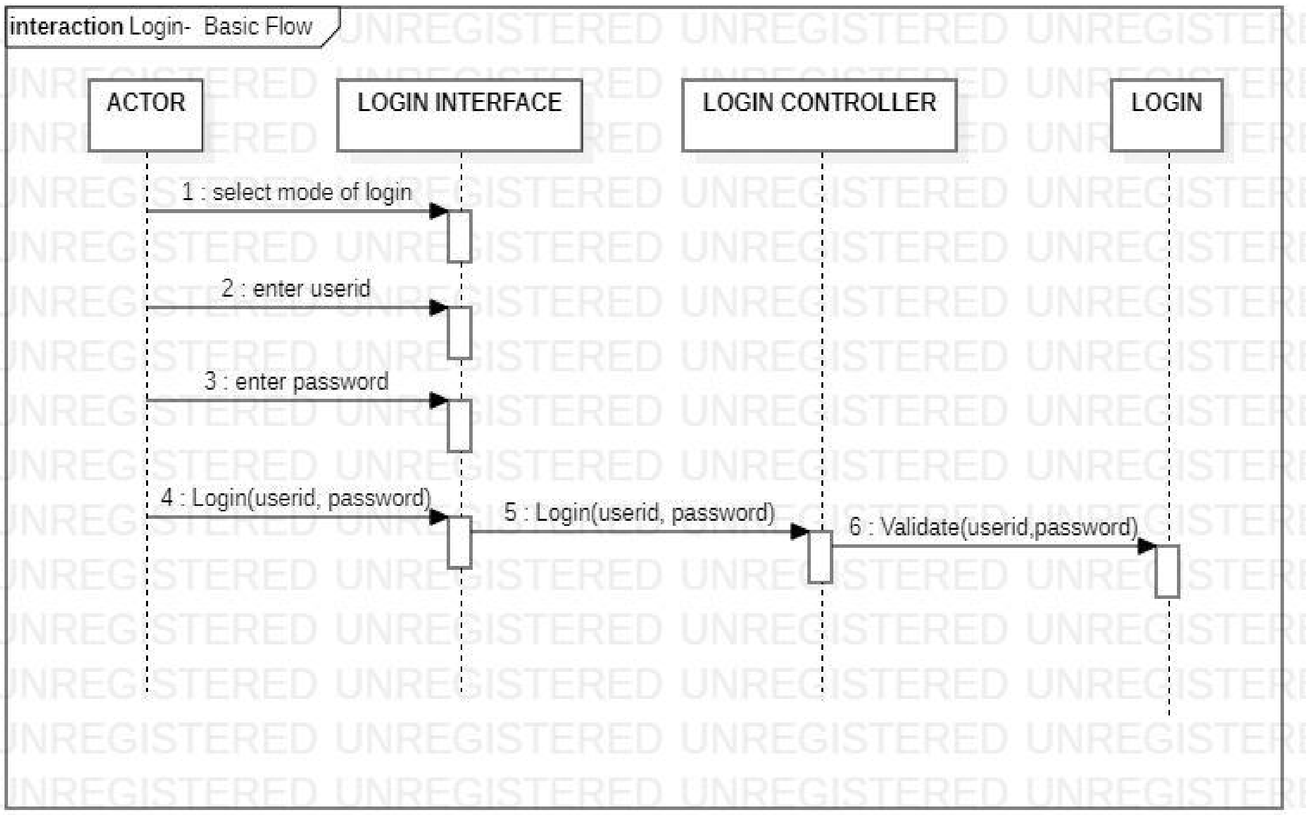


Class Diagram



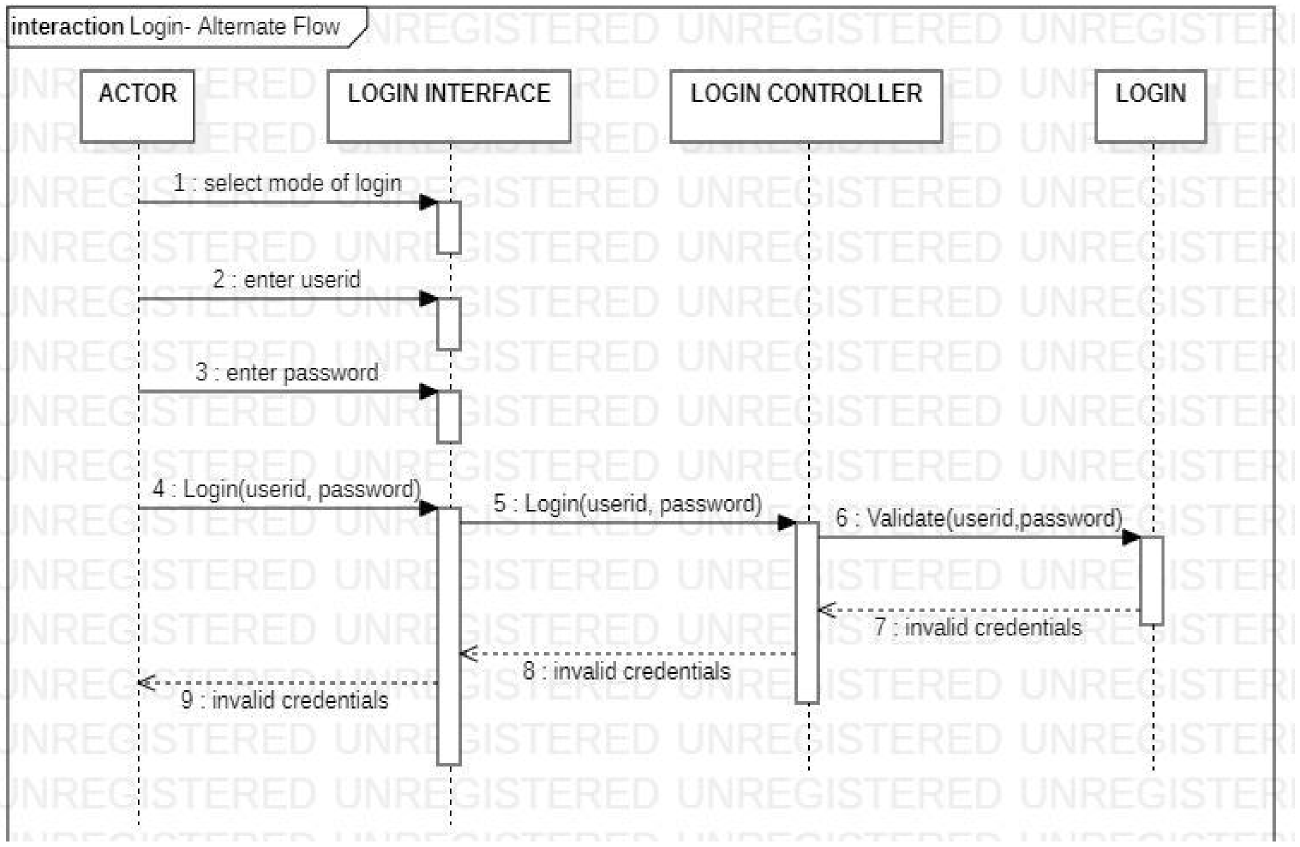
Sequence Diagrams

1. **Login – basic flow:**

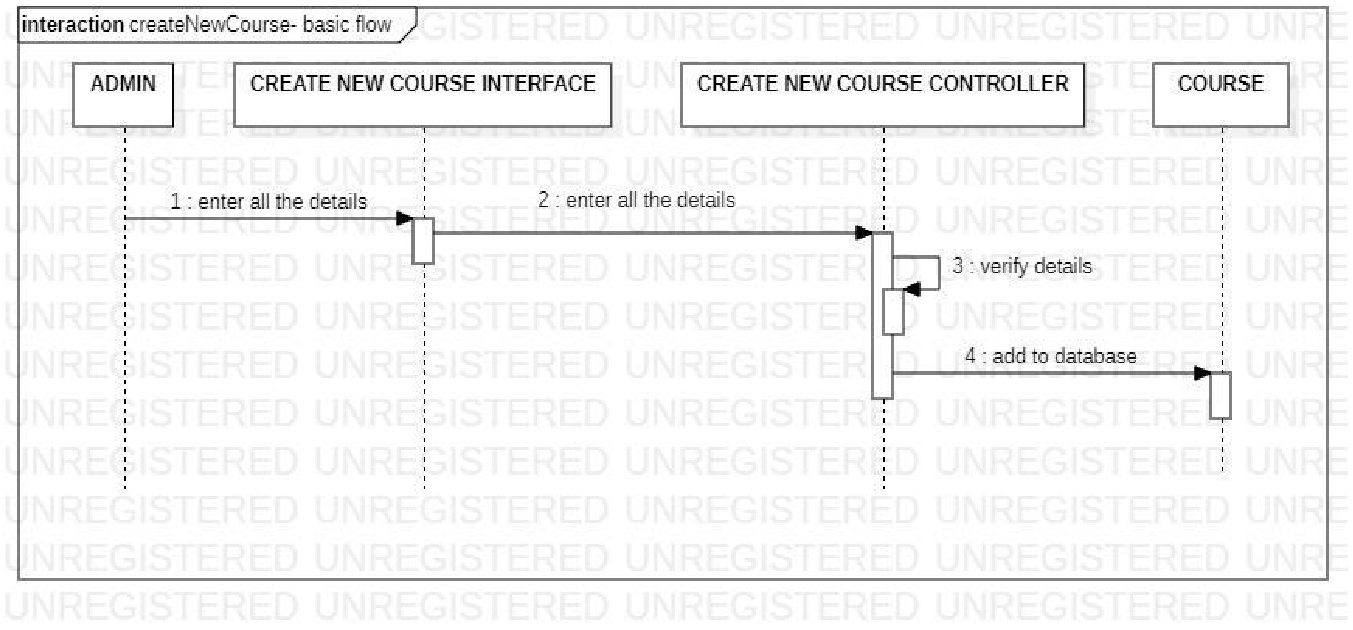


interaction Login- Basic Flow
ACTOR
LOGIN INTERFACE
LOGIN CONTROLLER
LOGIN
1 : select mode of login
2 : enter userid
3 : enter password
4 : Login(userid, password)
5 : Login(userid, password)
6 : Validate(userid,password)

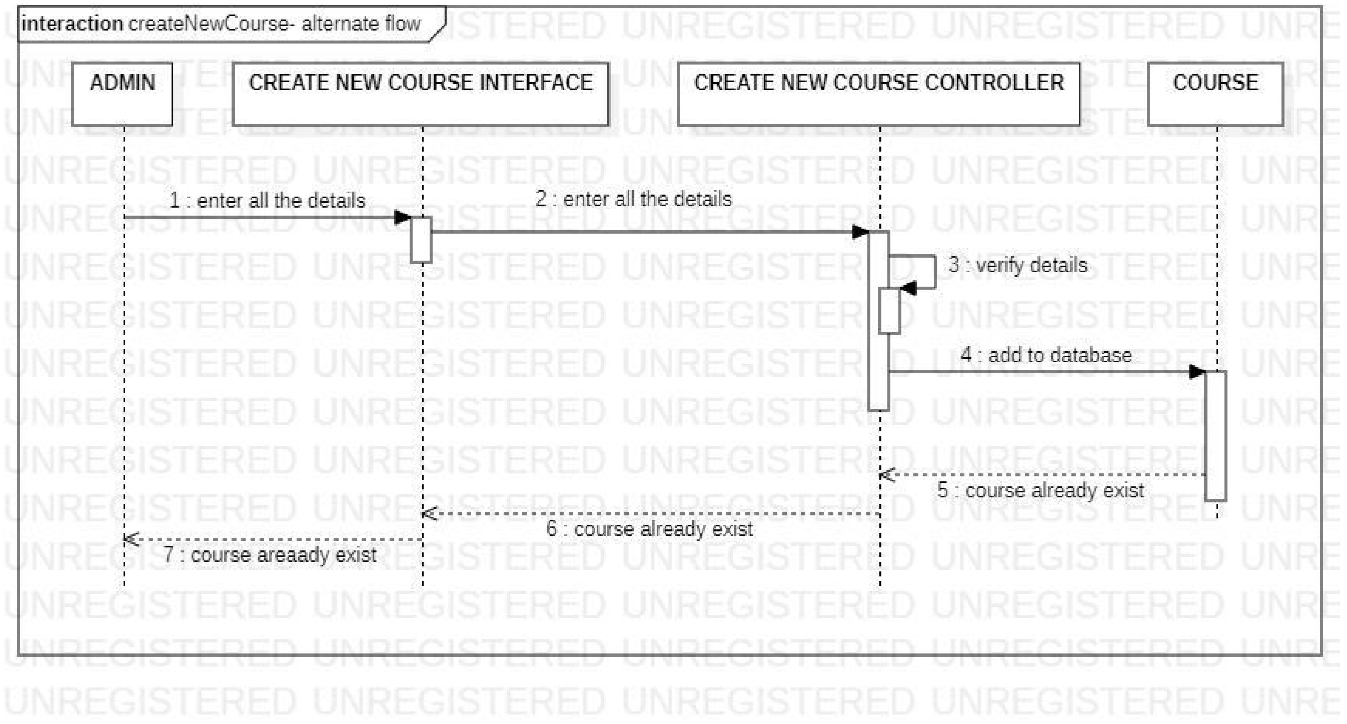
2. **Login – alternate flow:**



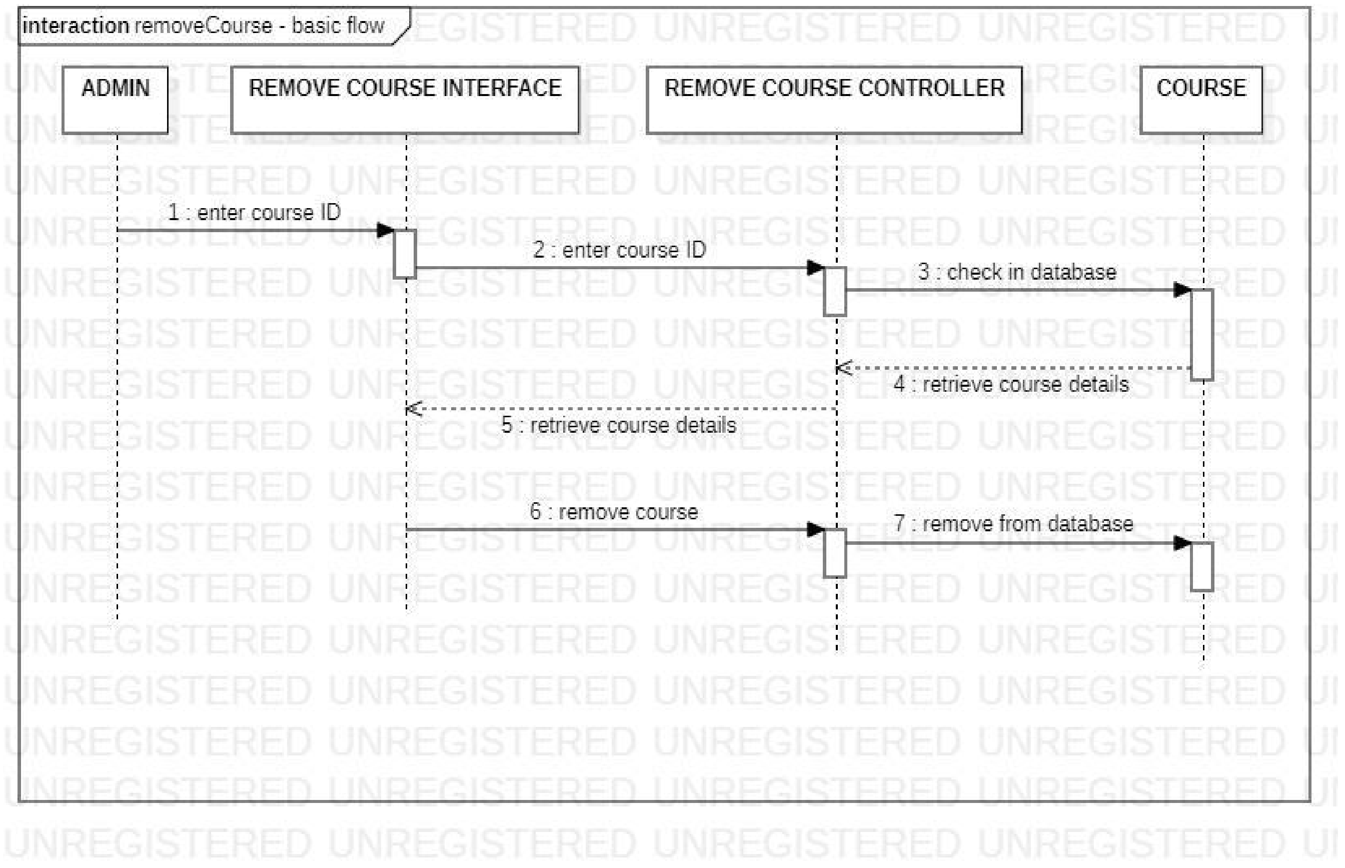
3. **Create new course – basic flow:**


interaction createNewCourse- basic flow
ADMIN
CREATE NEW COURSE INTERFACE
CREATE NEW COURSE CONTROLLER
COURSE
1 : enter all the details
2 : enter all the details
3 : verify details
4 : add to database

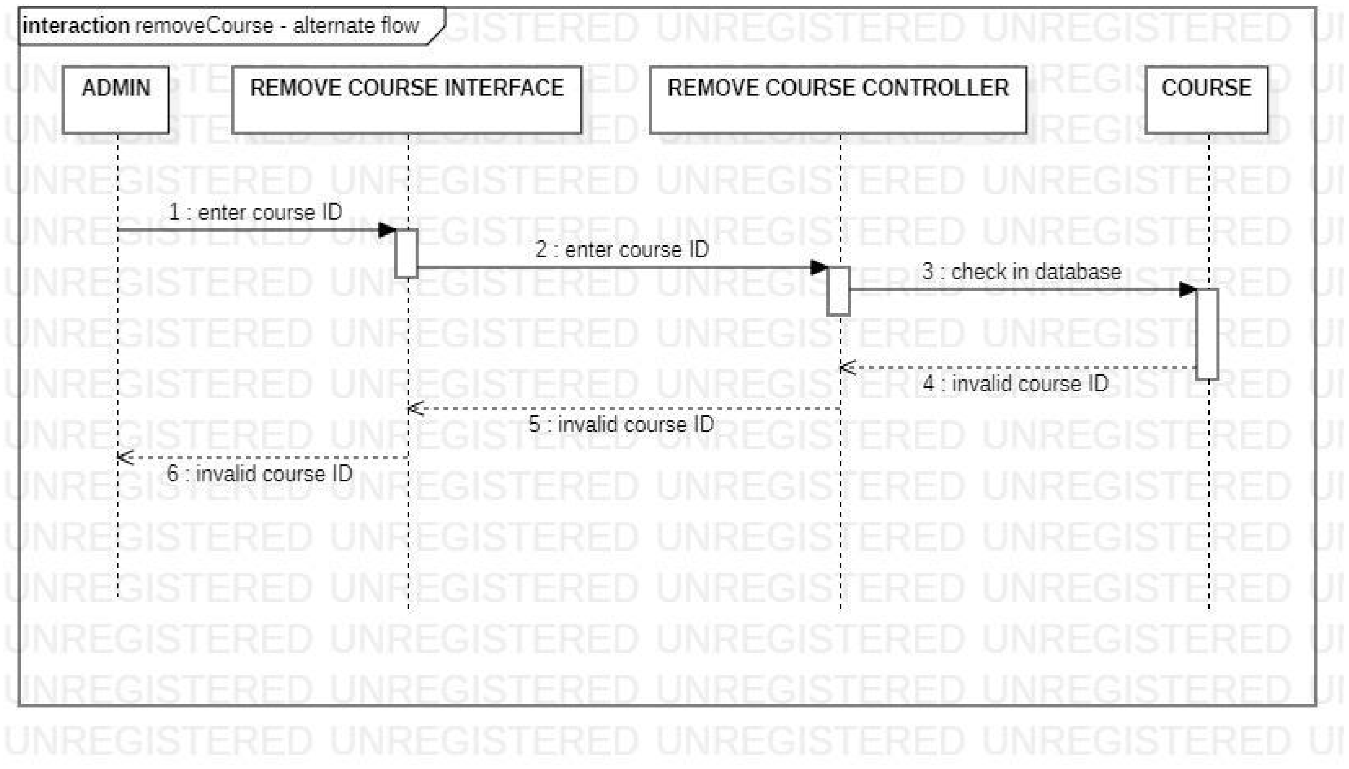
4. **Create new course – alternate flow:**


interaction createNewCourse- alternate flow
ADMIN
CREATE NEW COURSE INTERFACE
CREATE NEW COURSE CONTROLLER
COURSE
1 : enter all the details
2 : enter all the details
3 : verify details
4 : add to database
5 : course already exist
6 : course already exist
7 : course areaady exist

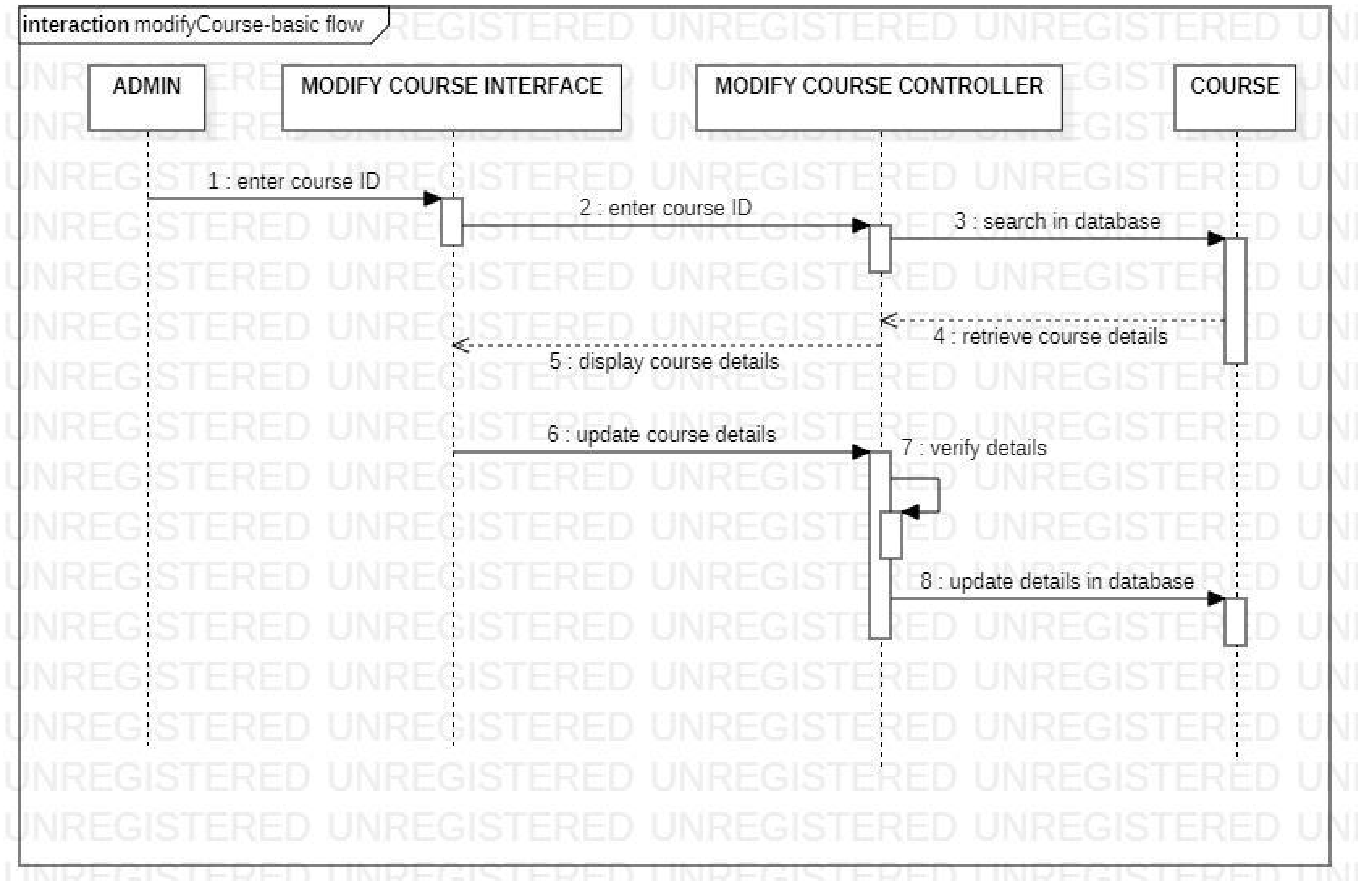
5. **Remove course – Basic flow:**



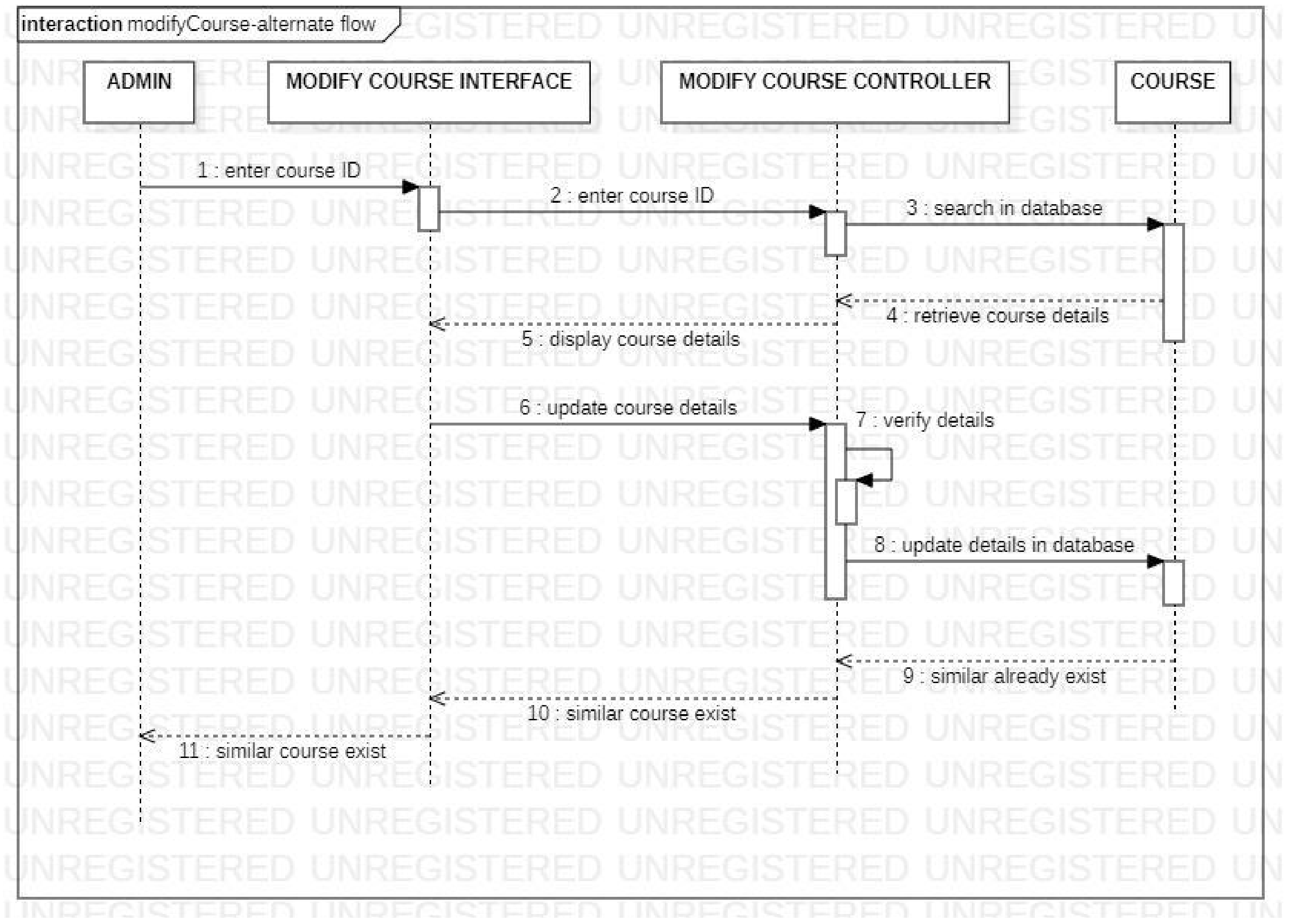
6. **Remove course – Alternate flow:**



7. **Modify course – Basic flow:**

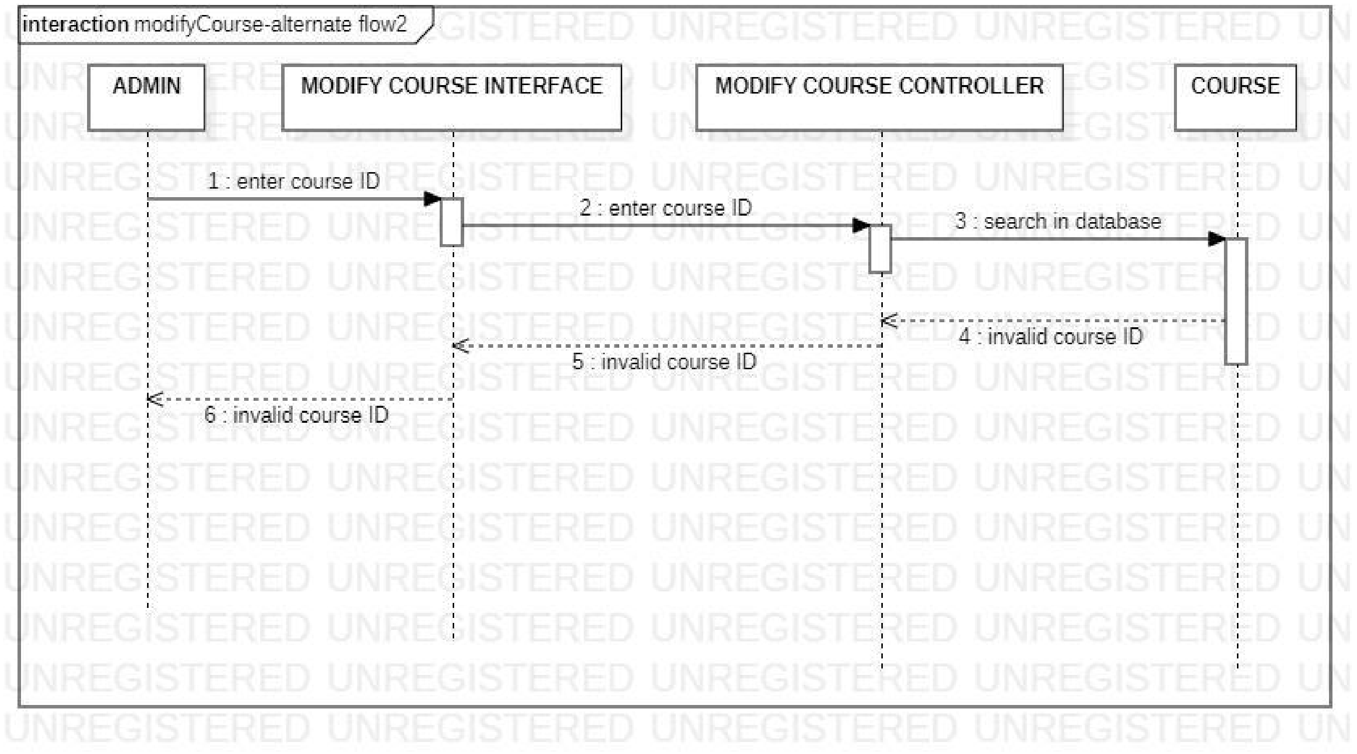

interaction modifyCourse-basic flow
ADMIN
MODIFY COURSE INTERFACE
MODIFY COURSE CONTROLLER
COURSE
1 : enter course ID
2 : enter course ID
3 : search in database
4 : retrieve course details
5 : display course details
6 : update course details
7 : verify details
8 : update details in database

8. **Modify course – Alternate flow1:**

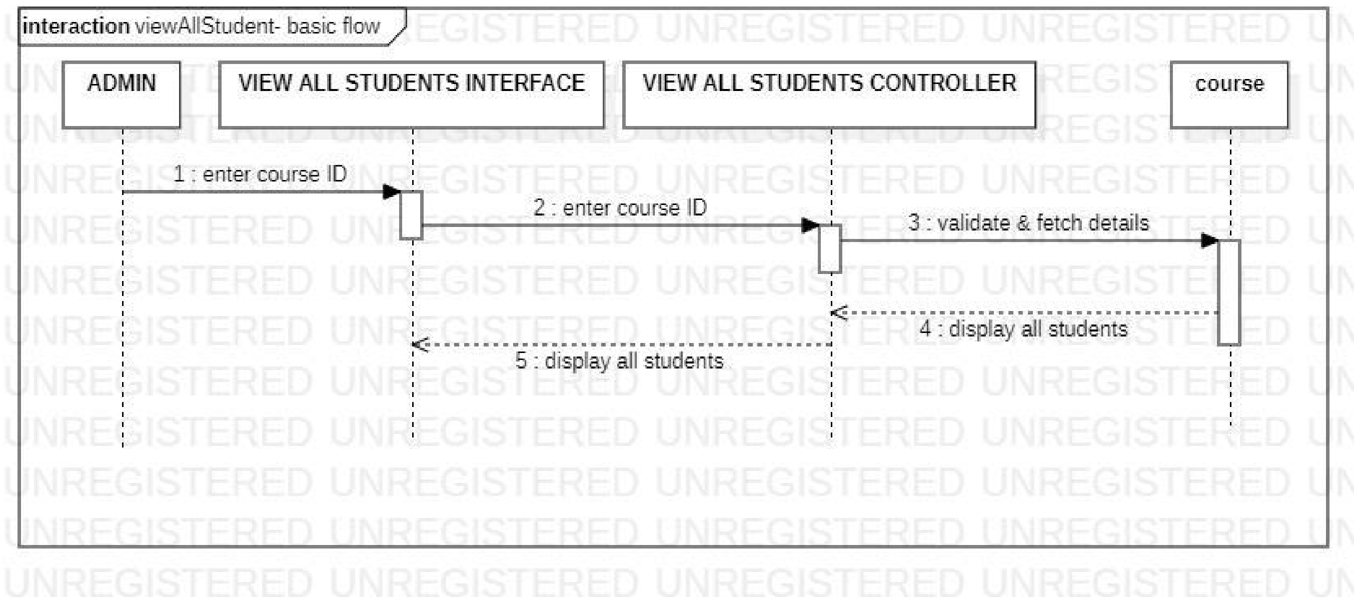


interaction modifyCourse-alternate flow
ADMIN
MODIFY COURSE INTERFACE
MODIFY COURSE CONTROLLER
COURSE
1 : enter course ID
2 : enter course ID
3 : search in database
4 : retrieve course details
5 : display course details
6 : update course details
7 : verify details
8 : update details in database
9 : similar already exist
10 : similar course exist
11 : similar course exist

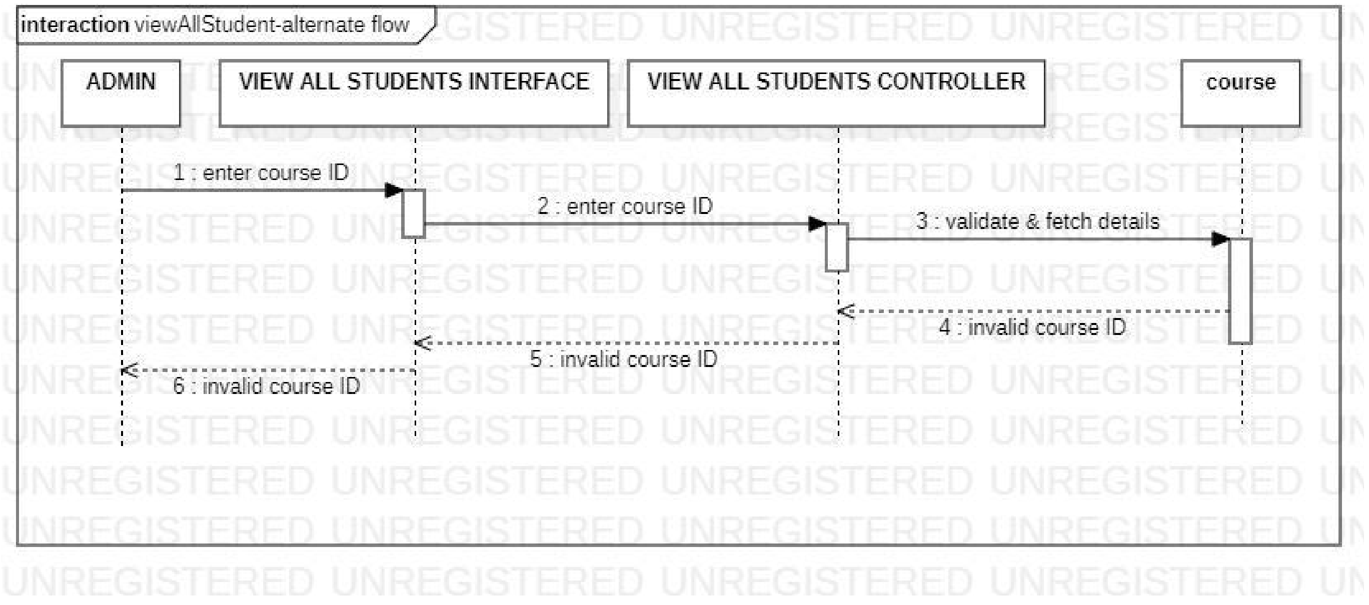
9. **Modify course – Alternate flow2:**



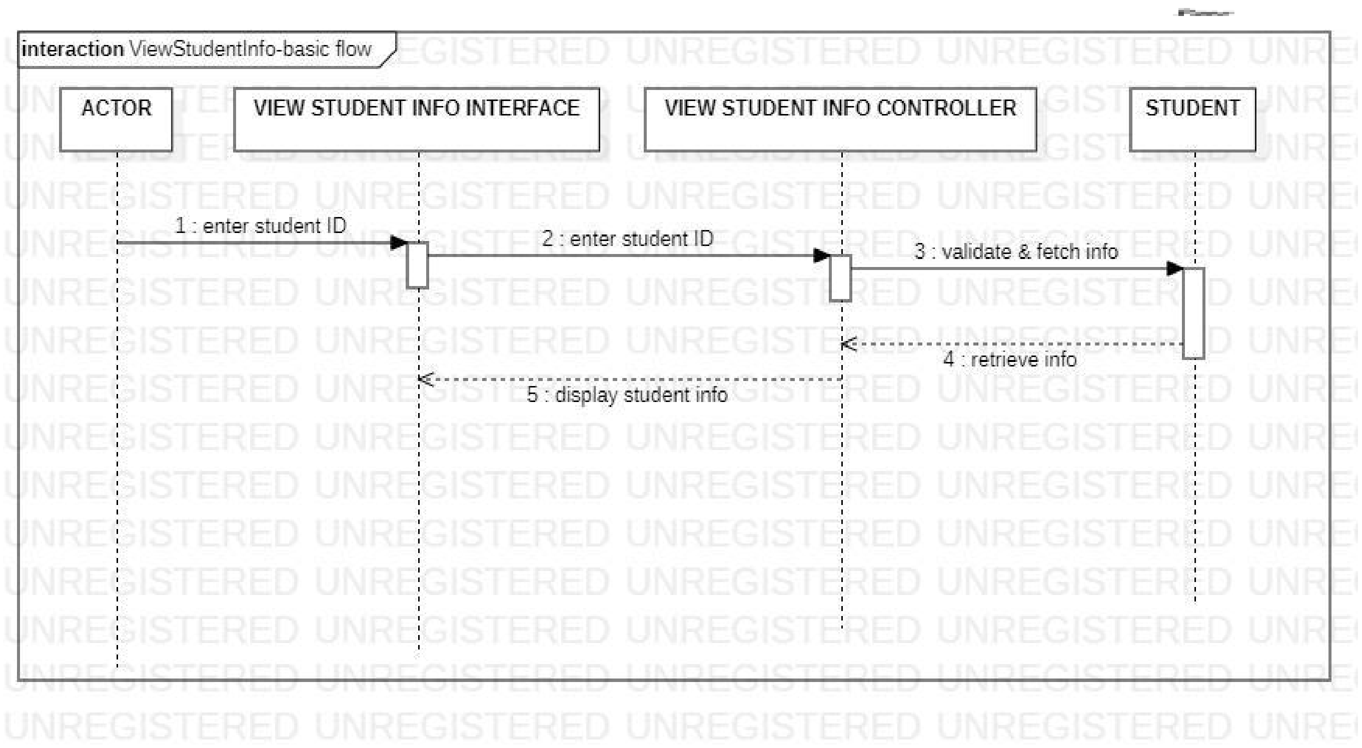
10. **View all student – Basic flow:**


interaction viewAllStudent- basic flow
ADMIN
VIEW ALL STUDENTS INTERFACE
VIEW ALL STUDENTS CONTROLLER
course
1 : enter course ID
2 : enter course ID
3 : validate & fetch details
4 : display all students
5 : display all students

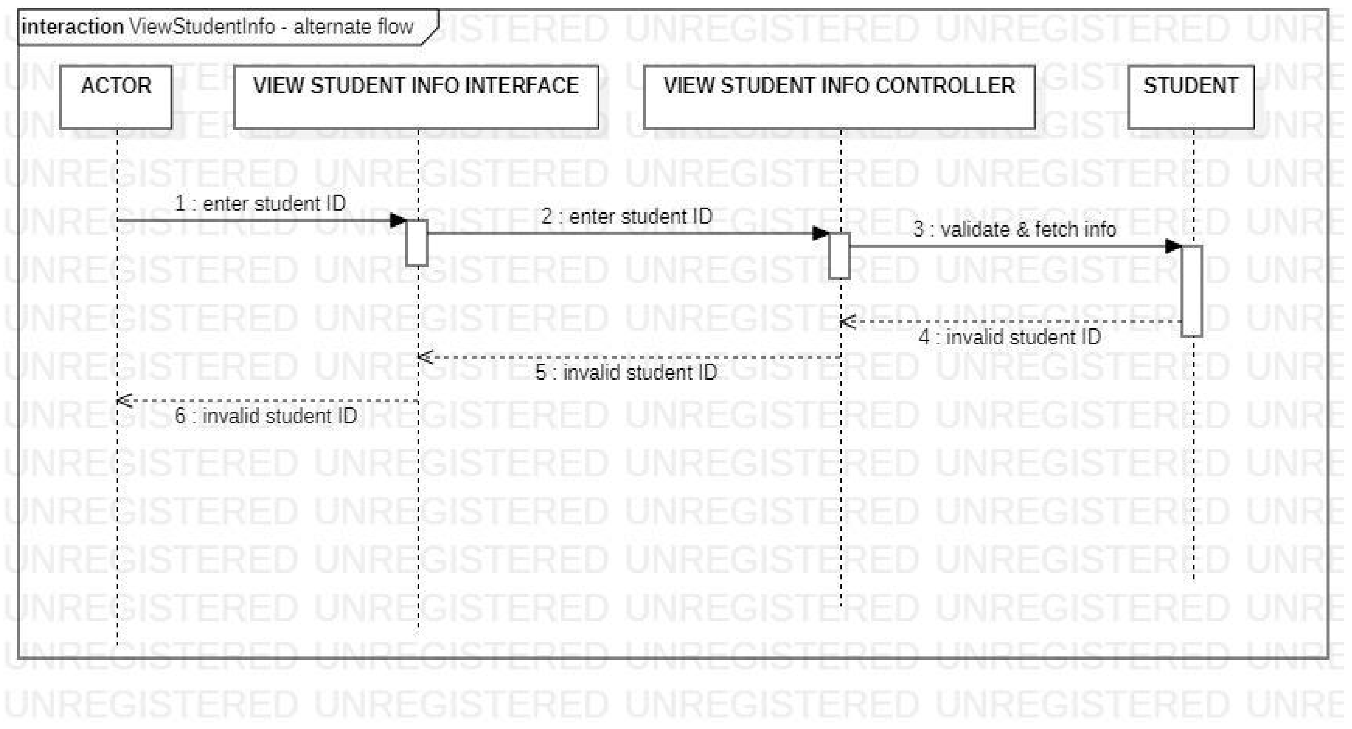
11. **View all student – alternate flow:**


interaction viewAllStudent-alternate flow
ADMIN
VIEW ALL STUDENTS INTERFACE
VIEW ALL STUDENTS CONTROLLER
course
1 : enter course ID
2 : enter course ID
3 : validate & fetch details
4 : invalid course ID
5 : invalid course ID
6 : invalid course ID

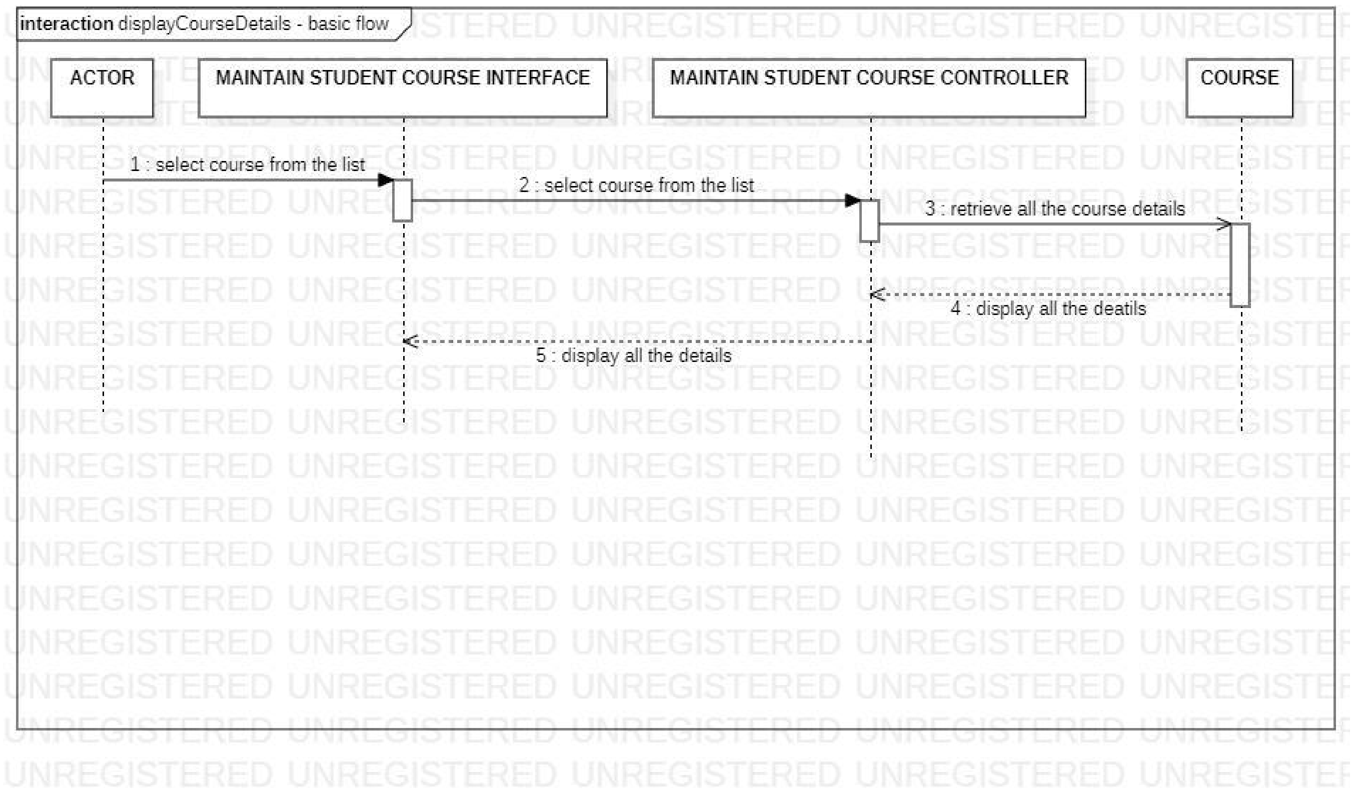
12. **View student info – Basic flow:**


interaction ViewStudentInfo-basic flow
ACTOR
VIEW STUDENT INFO INTERFACE
VIEW STUDENT INFO CONTROLLER
STUDENT
1 : enter student ID
2 : enter student ID
3 : validate & fetch info
4 : retrieve info
5 : display student info

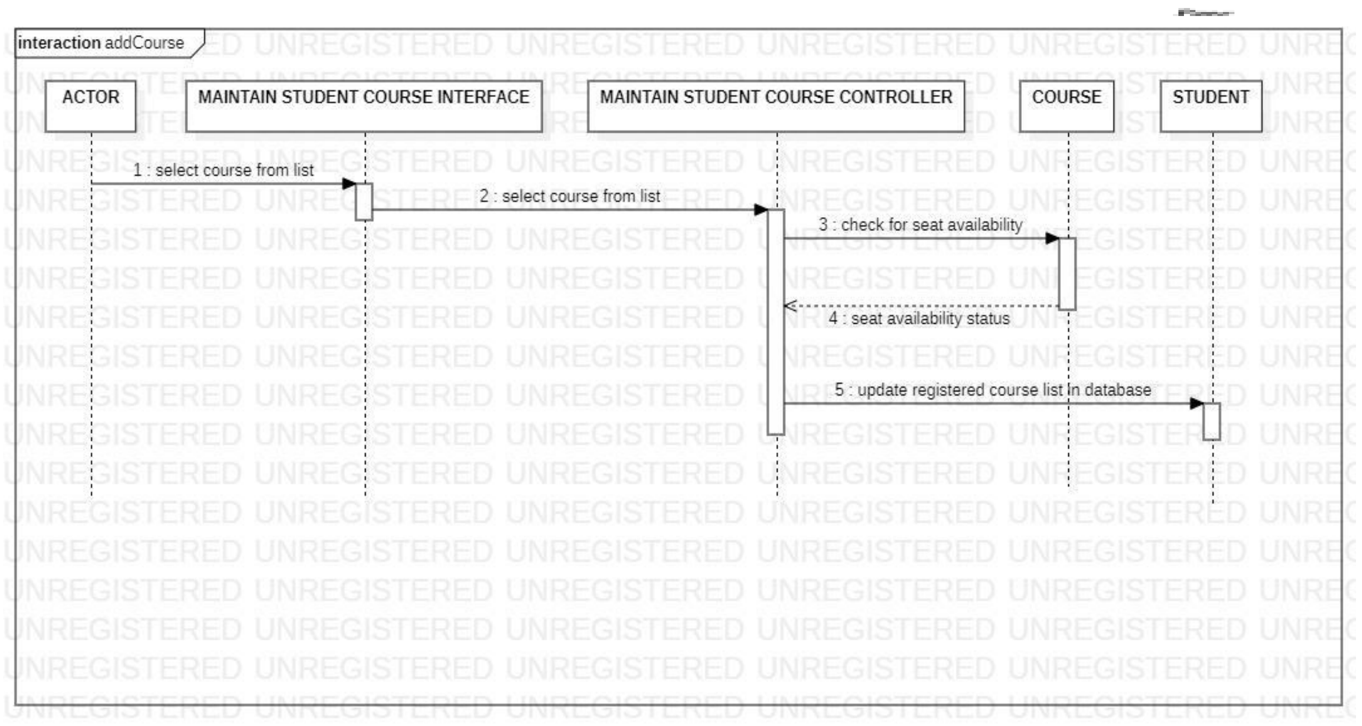
13. **View student info – Alternate flow:**


interaction ViewStudentInfo - alternate flow
ACTOR
VIEW STUDENT INFO INTERFACE
VIEW STUDENT INFO CONTROLLER
STUDENT
1 : enter student ID
2 : enter student ID
3 : validate & fetch info
4 : invalid student ID
5 : invalid student ID
6 : invalid student ID

14. **Display Course Details- basic flow:**

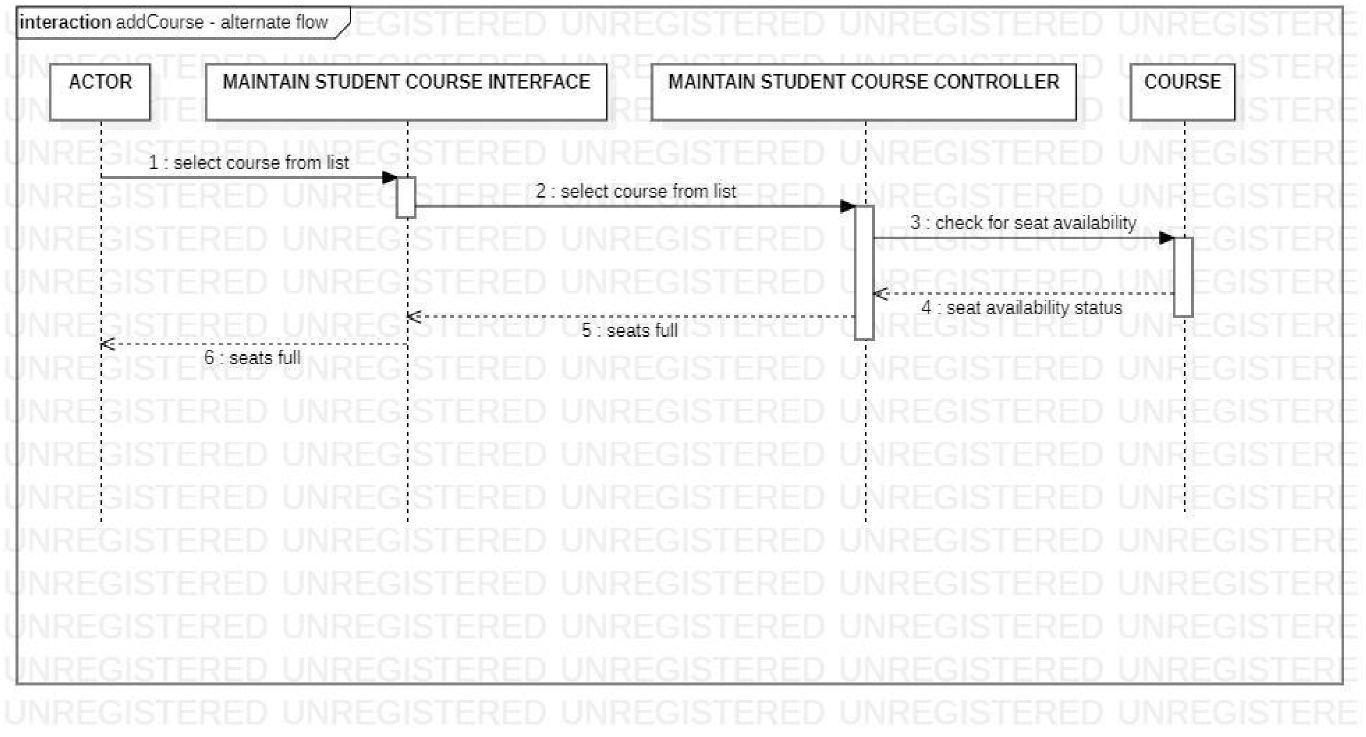

interaction displayCourseDetails - basic flow
ACTOR
MAINTAIN STUDENT COURSE INTERFACE
MAINTAIN STUDENT COURSE CONTROLLER
COURSE
1 : select course from the list
2 : select course from the list
3 : retrieve all the course details
4 : display all the deatils
5 : display all the details

15. **Add Course – Basic flow:**



interaction addCourse
ACTOR
MAINTAIN STUDENT COURSE INTERFACE
MAINTAIN STUDENT COURSE CONTROLLER
COURSE
STUDENT
1 : select course from list
2 : select course from list
3 : check for seat availability
4 : seat availability status
5 : update registered course list in database

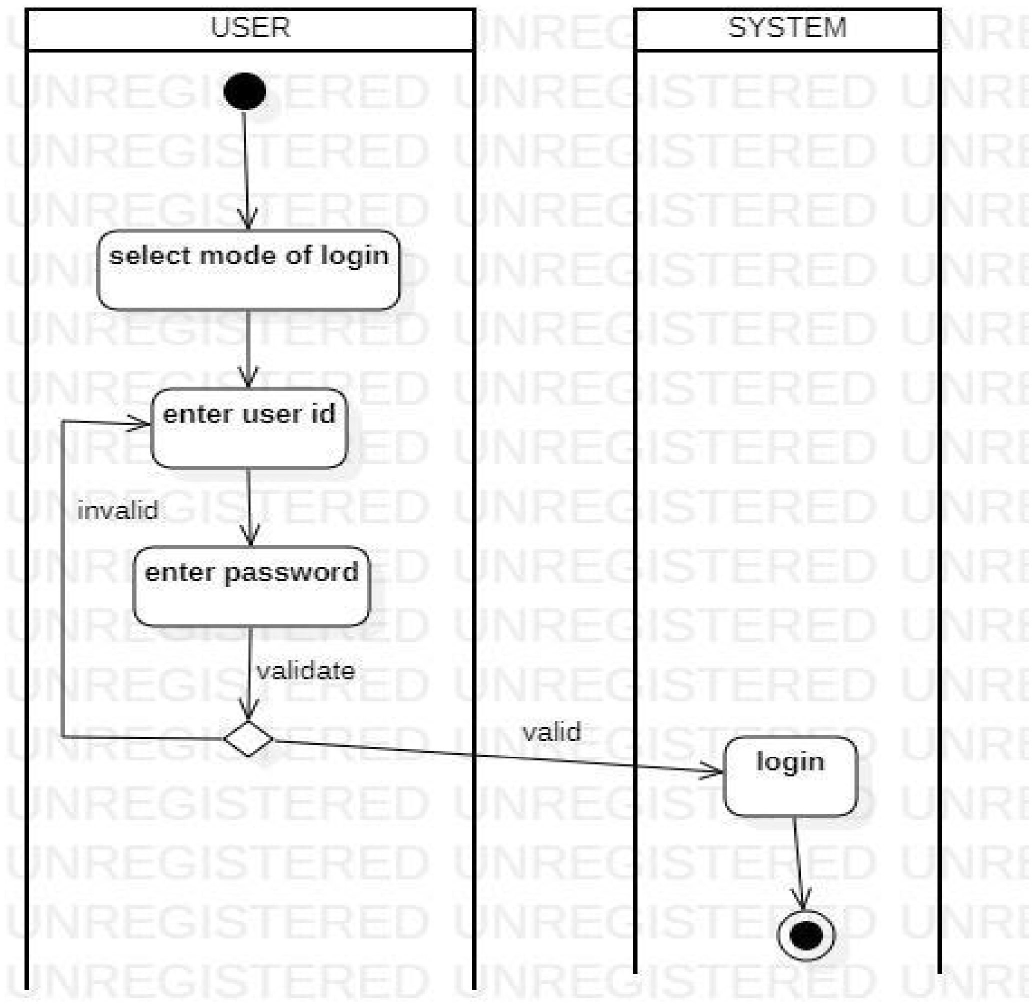
16. **Add Course – Alternate flow:**



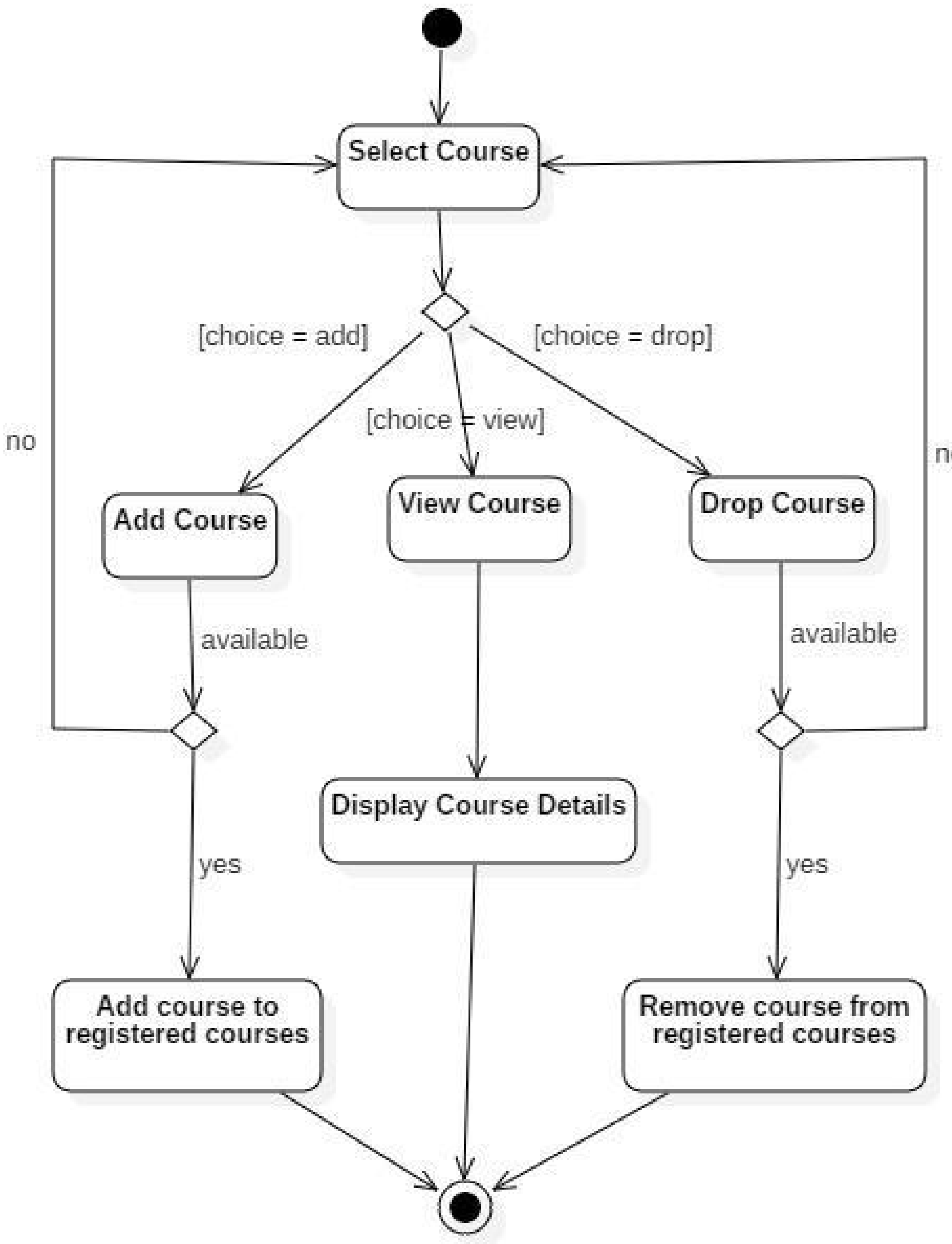
interaction addCourse - alternate flow
ACTOR
MAINTAIN STUDENT COURSE INTERFACE
MAINTAIN STUDENT COURSE CONTROLLER
COURSE
1 : select course from list
2 : select course from list
3 : check for seat availability
4 : seat availability status
5 : seats full
6 : seats full

Activity Diagram

1. Login

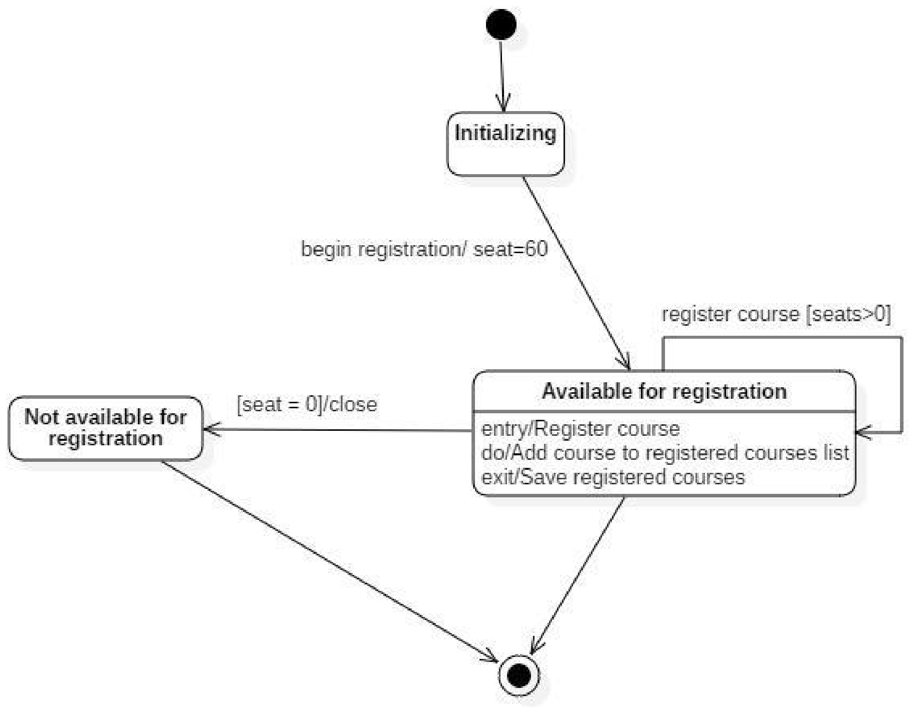


2. Course registration



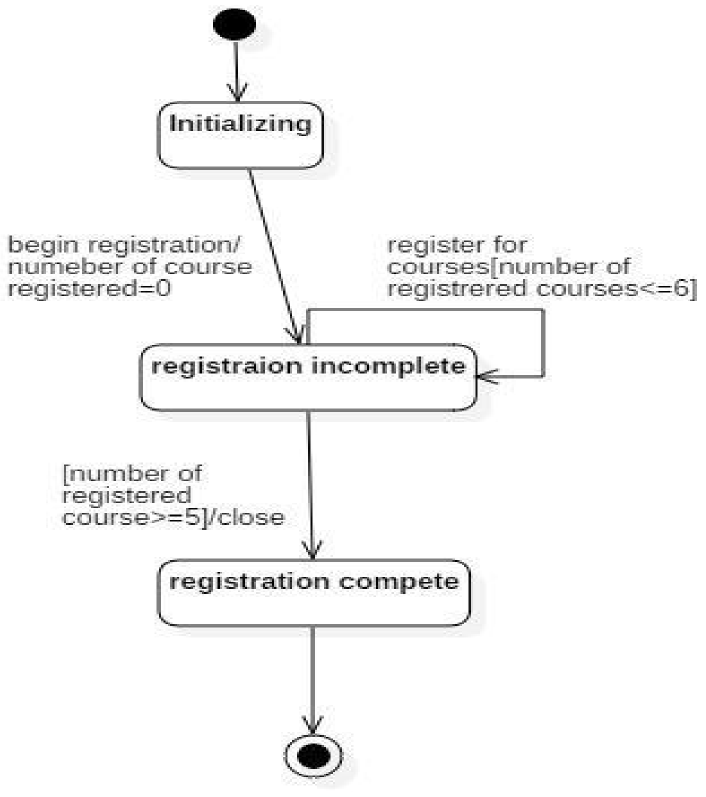
State Chart Diagram

1. Course



Initializing
begin registration/ seat=60
register course [seats>0]
Available for registration
entry/Register course
do/Add course to registered courses list
exit/Save registered courses
[seat = 0]/close
Not available for registration

2. Registration


Initializing
begin registration/ numeber of course registered=0
register for courses[number of registrered courses<=6]
registraion incomplete
[number of registered course>=5]/close
registration compete

Test case matrices

1. Create new course

Test Case ID	Scenario name and description	Input 1 Course ID	Input 2 Course name	Input 3 Branch	Input 4 Course type	Input 5 year	Input 6 credits	Excepted output	Remarks
TC_1	Scenario 1 – create new course: basic flow	OE311	Economics	All	Optional	3	4	Course added	A new course is added to database
TC_2	Scenario 2 – create new course alternate flow: invalid entry	-	Economics	All	Optional	3	4	Enter course Id	Course Id is not in the specified format
TC_3	Scenario 2 – create new course alternate flow: invalid entry	OE311	-	All	Optional	3	4	Enter course name	Course name is not in the specified format
TC_4	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	-	Optional	3	4	Branch not selected	Branch is not in the specified format
TC_5	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	-	3	4	course type not selected	Course type is not in the specified format
TC_6	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	Optional	-	4	Year not selected	Year is not in the specified format
TC_7	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	Optional	3	-	Credits not selected	Credits are not in the specified format
TC_8	Scenario 3 – create new course alternate flow : Course already exists	CO301	Economics	All	Optional	3	4	Course already exist	Similar course is already in database

2. Remove course

Test Case ID	Scenario name and description	Input 1 Course ID	Excepted output	Remarks
TC_1	Scenario 1 – remove course: basic flow	OE311	Course removed	The course removed from database
TC_2	Scenario 2 – remove course-alternate flow : invalid course id	-	Enter course Id	Course Id is not in the specified format
TC_3	Scenario 3 – remove course-alternate flow: wrong course id	OE310	Wrong course ID	No course with this id exists in database

3. Modify course

Test Case ID	Scenario name and description	Input 1 Course ID	Input 2 Course name	Input 3 Branch	Input 4 Course type	Input 5 year	Input 6 credits	Excepted output	Remarks
TC_1	Scenario 1 – Modify course: basic flow	OE311	Economics	All	Optional	2	4	Course modified	The modified course id added to database
TC_2	Scenario 2 – modify course alternate flow: invalid entry	-	Economics	All	Optional	3	4	Enter course Id	Course Id is not in the specified format
TC_3	Scenario 2 – create new course alternate flow: invalid entry	OE311	-	All	Optional	3	4	Enter course name	Course name is not in the specified format
TC_4	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	-	Optional	3	4	Branch not selected	Branch is not in the specified format
TC_5	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	-	3	4	course type not selected	Course type is not in the specified format
TC_6	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	Optional	-	4	Year not selected	Year is not in the specified format
TC_7	Scenario 2 – create new course alternate flow: invalid entry	OE311	Economics	All	Optional	3	-	Credits not selected	Credits are not in the specified format
TC_8	Scenario 3 – create new course alternate flow : Course already exists	CO301	Economics	All	Optional	3	4	Course already exist	Similar course is already in database

4. Student personal details

Test Case ID	Scenario name and description	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Excepted output	Remarks
		Name	**Branch**	Year	Phone	email	gender	Address		
TC$_1$	Scenario 1 – all the details are valid	Aaron	SE	3	9968337733	aaron@gmail.com	male	Lajpat nagar	Details saved successfully	The details are save and course are shown
TC$_2$	Scenario 2 – any field invalid	-	SE	3	9968337733	aaron@gmail.com	Male	Lajpat nagar	Enter name	Name is in valid
TC$_3$	Scenario 2 – any field invalid	Aaron	-	3	9968337733	aaron@gmail.com	Male	Lajpat nagar	Select branch	Branch not selected
TC$_4$	Scenario 2 – any field invalid	Aaron	SE	-	9968337733	aaron@gmail.com	Male	Lajpat nagar	Select year	Year is invalid
TC$_5$	Scenario 2 – any field invalid	Aaron	SE	3	-	aaron@gmail.com	Male	Lajpat nagar	Phone number in valid	Phone number is not in correct format
TC$_6$	Scenario 2 – any field invalid	Aaron	SE	3	9968337733	-	Male	Lajpat nagar	Mail is not correct	Email in wrong format
TC$_7$	Scenario 2 – any field invalid	Aaron	SE	3	9968337733	aaron@gmail.com	-	Lajpat nagar	Select gender	Gender is not selected
TC$_8$	Scenario 2 – any field invalid	Aaron	SE	3	9968337733	aaron@gmail.com	male	-	Enter address	Address is invalid format