

**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING**

**OBJECT ORIENTED SOFTWARE
ENGINEERING
(SE-301)**



**CASE STUDY-
UNIVERSITY REGISTRATION
SYSTEM**

Submitted to:

Dr. Ruchika Malhotra

Submitted by:

Anushree Mahur- 2K18/SE/033

Anmolpreet Singh- 2K18/SE/027

TABLE OF CONTENT

S.NO	DESCRIPTION	PAGE NUMBER
1.	Introduction	3
2.	Problem Statement	4
3.	Initial Requirements Document	5
4.	Software Requirements Specification	6
5.	Use Case Diagram	41
6.	Class Diagram	42
7.	Sequence Diagrams	43
8.	Activity Diagrams	73
9.	Statechart Diagram	75
10.	Test Cases	76
11.	Implementation Details	81

INTRODUCTION

University Registration System(URS) is an user-friendly cross platform mobile application system compatible with both Android version 4.0+ and IOS version 12+ operating system. URS is an automated registration system , which is mainly concerned with easy and efficient registration of newly admitted students and faculty members of the University. The system allows the user to maintain an up to date profile and contact information and view relevant information, after logging into the system .

The system allows the Administrator of the University to maintain records of students currently enrolled in the University along with their contact information and courses enrolled by them and records of faculty currently teaching in the University. It also allows the administrator to maintain records of different degrees offered by the University and records of different courses offered in a particular semester.

The proposed system overcomes the drawbacks of the existing manual registration process and has the following benefits:

- Reduces manual efforts and overheads of University staff and administrators and is time saving and convenient to use.
- Increased security - URS guarantees the security of information stored in it by the use of passwords to restrict access of the information by unauthorised users. The system has a backup facility that highly minimise loss of information in case of any calamity
- Reduce cost of files and writing material to store manual records.
- Reduces data redundancy

URS can be a useful tool for any University to automate its registration process. Further , URS combined with the Course Registration System , Fee Management System, Attendance Management System and Result Management System can become a complete software system for students and faculty of any University to maintain and view all their relevant information at one place.

This Object Oriented Software Engineering project completed under the guidance of Dr. Ruchika Malhotra outlines the different phases of Object Oriented Life Cycle and documentation produced after each phase. It consists of different UML diagrams which are used to visualize and model the requirements specified in the Software Requirement Specification (SRS) document. On the basis of the documentation created, we have implemented the Version 1.0 of the University Registration System.

PROBLEM STATEMENT

A university is an institution that offers various undergraduate and postgraduate degrees. Each degree offers various courses according to a particular semester. In the manual registration system, a new student or faculty has to submit a hardcopy of the filled application form to the college/university, office staff enters all data into excel file and writes the same in manual register to maintain the records.

Drawbacks of Existing Manual Registration System:

- Takes more processing time
- Involves a lot of manual labour and effort
- Requires a large amount of paperwork and physical records
- Poor maintenance of the databases in the long run
- Difficult to search the data and finding records
- Prone more to human errors and mistakes

The **need for an automated registration system** arose because it is a known fact that University has to deal with a large number of students and faculty members and to handle it manually is a tough job. By computerizing it, we will be able to overcome the shortfalls for the Manual Registration System and make the registration process smoother and maintain the database of students and faculty efficiently for a long period of time.

The proposed system - University Registration System(URS) is mainly concerned with an easy and efficient registration process by providing a provision for adding the details of the students and faculty by themselves to register their details in the University database. So that the overhead of the University authorities becomes less.

The objective of URS is to allow the administrator of the University to maintain student and faculty details along with maintaining the details about degrees offered by the University and courses taught in different degrees. It allows the student and the faculty to keep an up to date contact information and profile.

Basic functions that should be handled by URS are -

- Initial University Registration - This includes the registration of the student in a particular degree and faculty in the University, by the student and faculty respectively and are validated by the system, then the record of student/faculty details are added in the University Database. The system issues Registration Confirmation Slip via email. Once the student/faculty is registered in the University database, his/her account is created on the URS and proper login credentials are provided by the system to the student.
- Maintains the personal details of students.
- Maintains the list of students registered in a particular degree.
- Maintains the list of students registered for a particular course in a semester.
- Maintains the details of faculty and courses taught by them.
- Maintains a database of degrees offered by university and courses offered in each degree in a particular semester.
- The system allows the student to view and update his personal details and the information about the courses he/she has enrolled in that semester.
- The faculty can also view and update his details and the list of students that are enrolled in the course taught by him/her.

INITIAL REQUIREMENTS DOCUMENT

Title of the project	University Registration System
Stakeholders involved in capturing requirements	Admin, Employees, Students, Faculty, Project Leader
Techniques used for requirement capturing	Interviewing and Brainstorming
Name of the persons along with designations	Anushree Mahur , Anmolpreet Singh
Date	August,2020
Version	1.0

Consolidated list of initial requirements:

1. The system shall be implemented on cross platform integrated applications.
2. The system will be used by administrators, faculty and students of the university.
3. The student and the faculty shall be able to register themselves by entering their details.
4. The system must validate the details entered by the student or the faculty and create a user account if all the details are correct and validated.
5. The system shall be able to generate login ID and password for student and faculty after successful registration and validation .
6. The administrator shall be able to update, delete and view student and faculty records into the University Database.
7. The administrator shall be able to add, update, delete and view degree and course details.
8. The student should be able to view their personal details and details of courses they are enrolled in a particular semester.
9. The faculty should be able to view their personal details and list of students enrolled in the course taught by them.
10. The student, faculty and the administrator shall be able to manage their profiles i.e. update information , change password and logout of the system.
11. In case of complaints/queries, students/faculty/administrators can contact University through the system.
12. The system should be able to generate Initial University Registration Confirmation slip .
13. The system should also be able to generate reports like:
 - I. List of Students enrolled in the University
 - II. List of Faculty teaching the University
 - III. List of Degrees offered by the University
 - IV. List of Courses offered by the University in a particular semester

SOFTWARE REQUIREMENTS SPECIFICATION

For

UNIVERSITY REGISTRATION SYSTEM

Version 1.0

Prepared by Anushree Mahur & Anmolpreet Singh

Under the Guidance of Dr. Ruchika Malhotra

September 21,2020

TABLE OF CONTENTS:

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.4 References
- 1.5 Overview

2. Overall Description

- 2.1 Product Perspective
 - 2.1.1 System Interfaces
 - 2.1.2 User Interfaces
 - 2.1.3 Hardware Interfaces
 - 2.1.4 Software Interfaces
 - 2.1.5 Communication Interfaces
 - 2.1.6 Memory Constraints
 - 2.1.7 Operations
 - 2.1.8 Site Adaptation Requirements
- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions and Dependencies
- 2.6 Apportioning of Requirements

3. Specific Requirements

- 3.1 External Interface Requirements
 - 3.1.1 User Interfaces
 - 3.1.2 Hardware Interfaces
 - 3.1.3 Software Interfaces
 - 3.1.4 Communication Interfaces
- 3.2 Functional Requirements
 - 3.2.1 Login
 - 3.2.2 Initial Registration
 - 3.2.3 Profile Management
 - 3.2.4 Maintain Degree Details
 - 3.2.5 Maintain Course Details
 - 3.2.6 Maintain Student Details
 - 3.2.7 Maintain Faculty Details
 - 3.2.8 View Student-Course Details
 - 3.2.9 Contact University
 - 3.2.10 Report Generation
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software System Attributes
- 3.6 Logical Database Requirements
- 3.7 Other Requirements

1. INTRODUCTION:

This document namely Software Requirement Specification(SRS) gives a brief description about the purpose of the proposed system ,scope ,acronyms and abbreviations used for the system. It aims at defining the overall software requirements for “University Registration System”. The document provides clear and brief description about the requirements and stakeholders involved in the system. It also concentrates on the capabilities required by the stakeholders while defining the system.

1.1 PURPOSE

The University Registration System (URS) provides an easy and efficient way for students and faculty to register themselves on the system and view necessary details related to their profile and curriculum. The administrator through URS maintains the records of all students currently enrolled in the university and faculty currently teaching in the university. It allows the student and the faculty to keep an up to date contact information and profile in the University database. Also the administrator maintains the records of different degrees and courses offered by the University.

1.2 SCOPE

Name of the software is University Registration System (URS).The system will be referred to as URS in the rest of the SRS. The proposed URS must be able to perform the following functions.

DO's:

1. Maintains the personal details of students.
2. Maintains the list of students registered in a particular degree.
3. Maintains the list of students registered for a particular course in a semester.
4. Maintains the details of faculty and courses taught by them.
5. Maintains a database of degrees offered by university and courses offered in each degree in a semester.
6. Students/Faculty can register themselves in the University Database.
7. Generates and issues login credentials for registered students and faculty
8. The system allows the student to view and update his personal details and the information about the courses he/she has enrolled in that semester.
9. The faculty can also view and update his details and the list of students that are enrolled in the course taught by him/her.
10. Contact University via email in case of any complaints/queries/feedback.
11. Generates the following reports :
 - ❖ List of students currently enrolled in the University
 - ❖ List of degrees offered by the University
 - ❖ List of courses offered by the University
 - ❖ List of faculty currently teaching in the University

DON'Ts:

1. Fee payment feature is not provided in the system.
2. Attendance management is not provided in the system.
3. Course/Semester Registration is not provided by the system.

BENEFITS:

The URS provides the following benefits:

- Easy and efficient mechanism for registration of the students and faculty.
- Helps to maintain up to date contact information of students and faculty in the University Database.
- Feature to contact the University incase of complaint/query or provide feedback.
- Generates important reports.

1.1 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

The definitions and acronyms used in URS are given as :

SRS: Software Requirement Specification

URS: University Registration System

RAM: Random Access Memory

IEEE : Institute of Electrical and Electronics Engineers

Administrator: University Official having all the privileges to operate the URS.

Student: Any candidate admitted in a degree offered by the university .

Faculty: Teaching staff of the university-Professor, Associate Professor and Assistant Professor.

1.2 REFERENCES

The referenced material used in URS is given as :

- a.Object Oriented SOftware Engineering by Prof. Yogesh Singh & Dr. Ruchika Malhotra.
- b.IEEE Recommended Practice for Software Requirements Specifications-IEEE Std. 830-1998.

1.3 OVERVIEW

The rest of the SRS document describes various system requirements, interfaces, features and functionalities.

2. OVERALL DESCRIPTION

The URS provides an easy and efficient mechanism for registration of newly admitted students and faculty in the University database. URS maintains the records of students and faculty in the University. It also maintains the records of different degrees and courses offered by the University.

It is assumed that the student has a valid Student ID corresponding to a successful admission in a specific degree in the university. It is also assumed that the faculty has a valid Faculty ID.

2.1 PRODUCT PERSPECTIVE

The URS shall be developed using client/server architecture and be compatible with cross platform both Android and IOS Operating System. The front-end of the system will be developed using Flutter SDK and the back-end will be developed using Firebase.

2.1.1 SYSTEM INTERFACES

None

2.1.2 USER INTERFACES

The URS will have the following user-friendly and menu-driven interfaces:

1. **Login:** To allow the entry of only authorized users through valid login ID and password.
2. **Student Details:** To maintain students details.
3. **Faculty Details:** To maintain faculty details.
4. **Initial Registration:** To allow new students and faculty to register themselves in the University database and on successful validation of registration details create user account and provide login credentials
5. **Degree and Course Details:** To maintain details of different degrees offered in the university and different courses offered in a particular semester.
6. **View Student Course details:** To view the courses enrolled by a student in a particular semester and to view the list of students enrolled in the course taught by a faculty.
7. The software should generate the following information:
 - List of students currently enrolled in the University
 - List of degrees offered by the University
 - List of courses offered by the University
 - List of faculty currently teaching in the University

2.1.3 HARDWARE INTERFACES

1. Screen resolution of at least 640 x 480 or above.
2. Mobile systems will be in the networked environment as it is a multi-user system.

2.1.4 SOFTWARE INTERFACES

1. Android/IOS Operating System
2. Google Flutter for designing cross-platform front-end
3. Google Firebase for back-end

2.1.5 COMMUNICATION INTERFACES

Communication is via a wide area network (WAN).

2.1.6 MEMORY CONSTRAINTS

At least 512 MB RAM and 500 MB space of memory will be required to run the software.

2.1.7 OPERATIONS

None

2.1.8 SITE ADAPTATION REQUIREMENTS

The terminal at the client site will have to support the hardware and software interfaces specified in sections 2.1.3 and 2.1.4, respectively.

2.2 PRODUCT FUNCTIONS

The URS will allow access only to authorized users with specific roles (Administrator, Student and Faculty). Depending upon the user's role, he/she will be able to access only specific modules of the system. A summary of major functions that the URS will perform is given as follows:

- A login facility for enabling only authorized access to the system.
- The administrator will be able to add, modify, delete or view degree and course information.
- The administrator will be able to modify, delete or view student and faculty information.

- The student/faculty shall be able to register themselves in the University database.
- On successful validation of registration details filled by student/faculty, a user account is created on the system and login credentials are provided to student/faculty.
- The student/faculty shall be able to view relevant information and manage their profiles.
- The administrator/student/faculty shall be able to contact the University incase of complaint/query or submit a feedback.
- The administrator will be able to generate various reports from the URS.

2.3 USER CHARACTERISTICS

- Qualification: At least matriculation and comfortable with English.
- Experience: Should be well versed/informed about the registration process.
- Technical Experience: Elementary knowledge of computers and smartphones.

2.4 CONSTRAINTS

- There will be only one administrator.
- The delete operation is available to the administrator. To reduce the complexity of the system, there is no check on delete operation. Hence, the administrator should be very careful before deletion of any record and he/she will be responsible for data consistency.
- Users will not be allowed to update the primary key.

2.5 ASSUMPTIONS AND DEPENDENCIES

- The login credentials are created by the user itself i.e. User ID is the email ID and password entered in the initial registration form.
- The login credentials for the administrator are provided by the University initially. Later the Admin can modify his/her email ID and password.
- The administrator cannot add a student or faculty. The student/faculty need to register themselves by entering their details.
- The academic section of the University will provide the list of degrees and Student IDs enrolled in the degree.
- The academic section of the University will provide the list of courses and Student IDs enrolled in the course.
- The student has a valid Student ID corresponding to a successful admission in a specific degree in the university and the faculty has a valid Faculty ID issued by the University.
- One course is taught only by one faculty and vice-versa.

2.6 APPORTIONING OF REQUIREMENTS

Not required

3. SPECIFIC REQUIREMENTS

This section contains the software requirements in detail along with the various forms to be developed.

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES:

The following user interfaces(or forms)will be provided by the system.

I. INITIAL REGISTRATION FORM

This form will be accessible only to the student/faculty. It will allow him/her to register in the University database.

i)For Student:

Various fields available on this form will be:

- **Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **Phone No. :** Numeric and must have 10 digits.
- **Email ID:** Alphanumeric and can have length up to 50 characters. Email must have at least one '@' and '.' symbol.
- **Password:** Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed.
- **Verify Password:** Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed. The content of this field must match with the contents of the new password field.
- **Student ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 4XX.
- **Select Degree:** Degree options to be selected from the drop down list.

The screenshot shows a mobile application titled "Student Registration". At the top, there is a blue header bar with the title "Student Registration" and a back arrow icon. Below the header are six input fields arranged vertically: "Name", "Phone", "Email", "Password" (with a lock icon), "Verify Password" (with a lock icon), and "Student Id". After these fields is a dropdown menu labeled "Please select a Degree" with a downward arrow. At the bottom is a large blue "Register" button.

ii) For Faculty:

Various fields available on this form will be:

- **Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **Phone No. :** Numeric and must have 10 digits.
- **Email ID:** Alphanumeric and can have length up to 50 characters. Email must have at least one '@' and '.' symbol.
- **Password:** Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed.
- **Verify Password:** Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed. The content of this field must match with the contents of the new password field.
- **Faculty ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 3XX.
- **Select Course:** Course option to be selected from the drop down list.

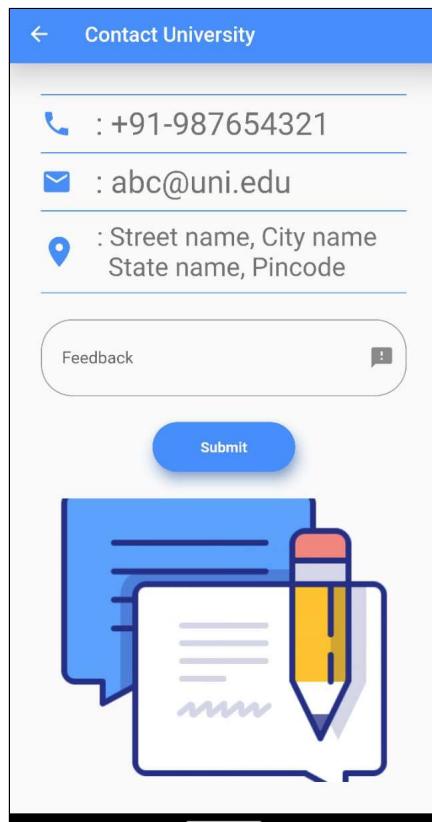
The screenshot shows a mobile application titled "Faculty Registration". At the top, there is a blue header bar with the title "Faculty Registration" and a back arrow icon. Below the header are six input fields arranged vertically: "Name", "Phone", "Email", "Password" (with a lock icon), "Verify Password" (with a lock icon), and "Faculty Id". After these fields is a dropdown menu labeled "Please select a Course" with a downward arrow. At the bottom is a large blue "Register" button.

II. CONTACT UNIVERSITY FORM

This form will be accessible only to the student/faculty/system administrator. It will allow him/her to contact the university via email incase of a complaint/query or to submit a feedback.

The field available on this form will be :

Submit a feedback: Text type

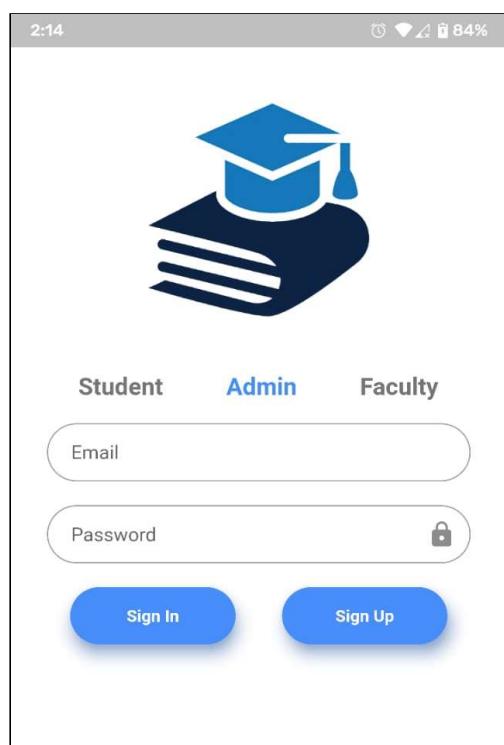


III. LOGIN FORM

This form will allow the user to access the different forms based on his/her role.

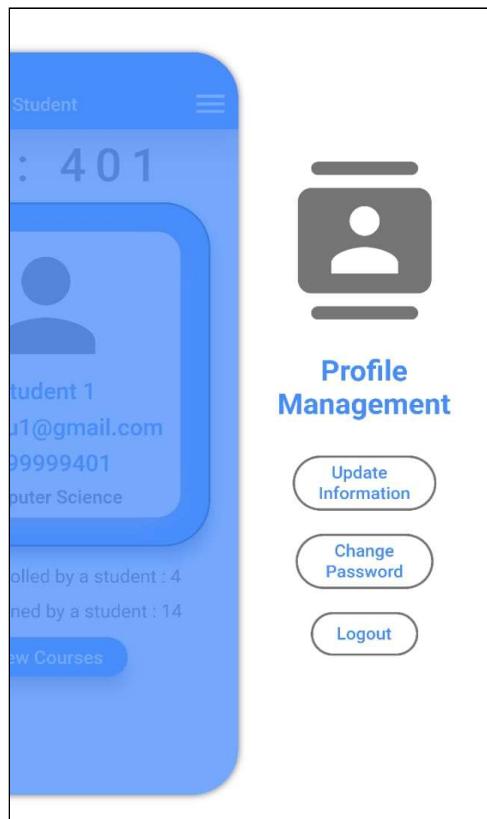
Various fields available on this form will be:

- User ID>Email:** Alphanumeric and user ID is the registered email ID of the user. User ID must contain @ and . characters. Blank spaces are not allowed.
- Password:** Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed.



IV. PROFILE MANAGEMENT

This form will be accessible only to all the users- administrator/student/faculty. It will allow him/her to update/view personal information, change password and logout of the system.



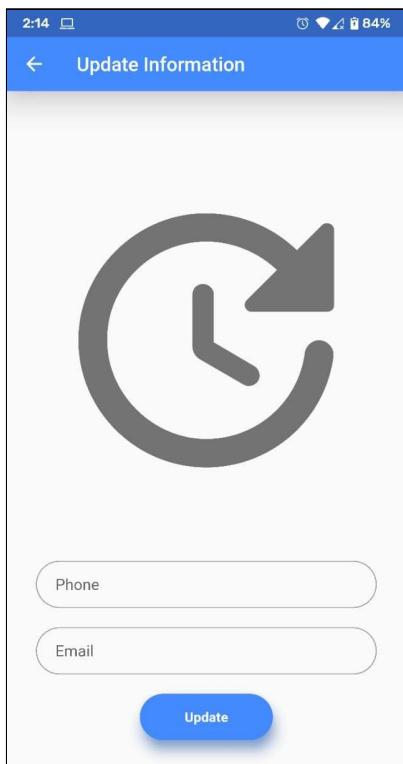
i) View Personal details:

Three mobile application interfaces side-by-side.
1. **Student:** Title "Student", ID "ID : 401", placeholder profile picture, "Student 1", "emailstu1@gmail.com", "999999401", "Computer Science". Below: "Courses enrolled by a student : 4" and "Credits attained by a student : 14". Blue button "View Courses".
2. **Faculty:** Title "Faculty", ID "ID : 302", placeholder profile picture, "Faculty 2", "emailfac2@gmail.com", "9999999302", "Basic Programming". Below: "Students taught by the faculty : 5". Blue button "View Students".
3. **Admin:** Title "Anmolpreet", email "axndtu@gmail.com", phone "8826225725". Placeholder profile picture. Four buttons: "Maintain Student Records", "Maintain Faculty Records", "Maintain Degree Records", "Maintain Course Records". Below: "Update", "View", "Delete", and a search bar "Id".

ii) Update Personal details:

Various fields available on this form will be:

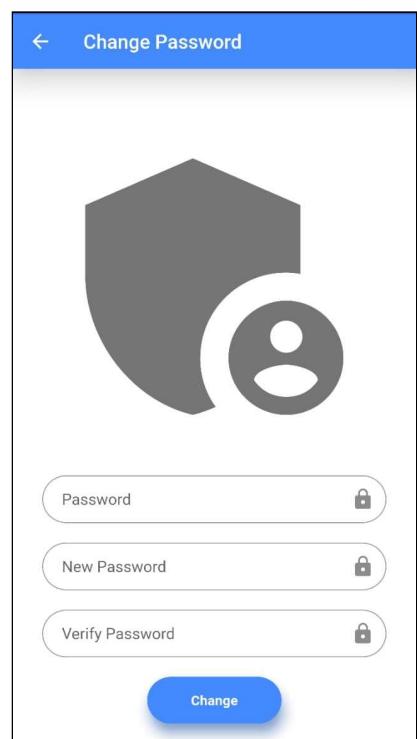
- **Phone No.** : Numeric and must have 10 digits.
- **Email ID**: Alphanumeric and can have length up to 50 characters. Email must have at least one '@' and '.' symbol.



ii) Change Password:

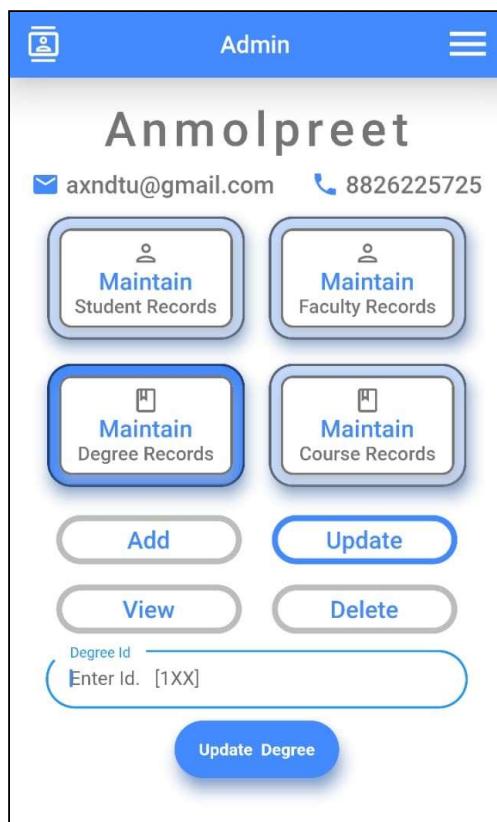
The change password form facilitates the user to change the password. Various fields available on this form will be:

- **Old Password**: Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed.
- **New Password**: Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed.
- **Verify Password**: Alphanumeric of length in the range of 8 to 16 characters. Blank spaces are not allowed. However, special characters are allowed. The content of this field must match with the contents of the new password field.



V. MAINTAIN DEGREE DETAILS

This form will be accessible only to the system administrator. It will allow him/her to add/update/delete/view information about degrees offered by the university.



Various fields available on the following forms will be:

- **Degree ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 1XX.
- **Degree Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **Student IDs:** Numeric of the format 4XX. It can be more than one , should be entered separated by commas.

ADD DEGREE FORM:

The screenshot shows a mobile application form titled 'Maintain Degree' with a sub-section 'Add Degree'. It has three input fields: 'Degree Id', 'Degree Name', and 'Student Ids'. Below these is a large blue button labeled 'Add Degree'.

UPDATE DEGREE FORM:

The image shows a mobile application interface. On the left, a card displays four buttons: Add, Update, View, and Delete. Below them is a text input field labeled "Degree Id" with placeholder text "Enter Id. [1XX]" and a blue outline. At the bottom is a blue "Update Degree" button. On the right, a larger card titled "Maintain Degree" shows "Update Degree" in blue text. It has two input fields: "Degree Name" and "Student Ids", followed by a blue "Update Degree" button. To the right of these cards is a vertical blue sidebar with various icons.

DELETE DEGREE FORM:

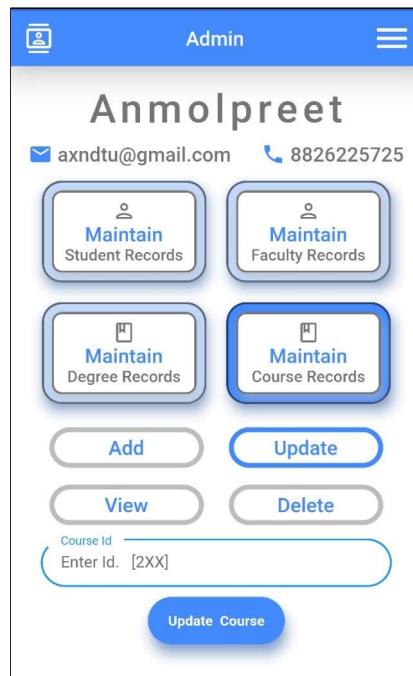
The image shows a mobile application interface. On the left, a card displays four buttons: Add, Update, View, and Delete. Below them is a text input field labeled "Degree Id" with placeholder text "Enter Id. [1XX]" and a blue outline. At the bottom is a blue "Delete Degree" button. On the right, a larger card titled "Maintain Degree" shows "Delete Degree" in blue text. It features a trash can icon with a large white "X" inside. Below it is a message: "Are you sure, you want to delete ? Then press delete". At the bottom is a blue "Delete Degree" button. To the right of these cards is a vertical blue sidebar with various icons.

VIEW DEGREE FORM:

The image shows a mobile application interface. On the left, a card displays four buttons: Add, Update, View, and Delete. Below them is a text input field labeled "Degree Id" with placeholder text "Enter Id. [1XX]" and a blue outline. At the bottom is a blue "View Degree" button. On the right, a larger card titled "Maintain Degree" shows "View Degree" in blue text. It displays a card with degree information: "101 Computer Science No. of Students: 6". At the bottom is a grey "View All" button. To the right of these cards is a vertical blue sidebar with various icons.

VI. MAINTAIN COURSE DETAILS

This form will be accessible only to the system administrator. It will allow him/her to add/update/delete/view information about courses offered in a particular semester by the university.



Various fields available on the following form will be:

- **Course ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 2XX.
- **Course Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **No. of credits:** Numeric and can range from 1 to 6.
- **No.of seats:** Numeric and can range from 1 to 999.
- **Student IDs:** Numeric of the format 4XX. It can be more than one , should be entered separated by commas.

ADD COURSE FORM:

A screenshot of a mobile application interface titled "Maintain Course" with a sub-section "Add Course". It contains six input fields: "Course Id", "Course Name", "Faculty Id", "Course Credits", "Course Seats", and "Student Ids". Each input field has a placeholder text below it. At the bottom is a blue "Add Course" button.

UPDATE COURSE FORM:

The mobile interface displays the 'Update Course' form. It includes a header 'Maintain Course' and a sub-header 'Update Course'. Below these are five input fields: 'Course Name', 'Faculty Id', 'Course Credits', 'Course Seats', and 'Student Ids'. At the bottom is a blue 'Update Course' button.

Update Course

Course Name

Faculty Id

Course Credits

Course Seats

Student Ids

Update Course

DELETE COURSE FORM:

The mobile interface displays the 'Delete Course' form. It features a large trash can icon with a red 'X' inside. Below the icon is a confirmation message: 'Are you sure, you want to delete ? Then press delete'. At the bottom is a blue 'Delete Course' button.

Delete Course

Are you sure, you want to delete ?
Then press delete

Delete Course

VIEW COURSE FORM:

The mobile interface displays the 'View Course' form. It shows course details for 'Data Structures': Course ID 201, Credits 5, Seats 8, Faculty ID 304. There is also a 'View All' button at the bottom.

View Course

201
Data Structures
Credits - 5 Seats - 8
Faculty ID - 304

View All

VII. MAINTAIN STUDENT DETAILS

This form will be accessible only to the system administrator. It will allow him/her to update/delete/view information about students enrolled in the university.

The screenshot shows a mobile application interface for an administrator. At the top, it displays the word "Admin". Below that is the name "Anmolpreet" and two contact details: an email address "axndtu@gmail.com" and a phone number "8826225725". There are four main buttons arranged in a 2x2 grid: "Maintain Student Records", "Maintain Faculty Records", "Maintain Degree Records", and "Maintain Course Records". Below these are three action buttons: "Update", "View", and "Delete". A text input field labeled "Student Id" contains the placeholder "Enter Id. [4XX]". At the bottom is a large blue button labeled "Delete Student".

Various fields available on the following forms will be:

- **Student ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 4XX.
- **Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **Phone No. :** Numeric and must have 10 digits.

UPDATE STUDENT FORM:

The image displays two mobile screens side-by-side. The left screen is titled "Maintain Student" and has a sub-section titled "Update Student". It features three buttons: "Update", "View", and "Delete". Below these is a text input field labeled "Student Id" with the placeholder "Enter Id. [4XX]". At the bottom is a blue button labeled "Update Student". The right screen is also titled "Maintain Student" and has a sub-section titled "Update Student". It contains two text input fields: one for "Student Name" and one for "Student Phone No.". At the bottom is a blue button labeled "Update Student".

DELETE STUDENT FORM:

The image shows two side-by-side mobile screen prototypes. The left screen is titled 'Maintain Student' and has a 'Delete Student' button at the top. Below it is a trash can icon with a large white 'X'. A message asks, 'Are you sure, you want to delete ? Then press delete'. At the bottom is another 'Delete Student' button. The right screen is also titled 'Maintain Student' and has a 'Delete Student' button at the top. It features a large trash can icon with a white 'X'. A message below it says, 'Are you sure, you want to delete ? Then press delete'. At the bottom is another 'Delete Student' button.

VIEW STUDENT FORM:

The image shows two side-by-side mobile screen prototypes. The left screen is titled 'Maintain Student' and has a 'View Student' button at the top. Below it is a person icon. A message box displays 'Student 1 - 401' with a phone icon, '9999999401' with a call icon, and 'emailstu1@gmail.com' with an envelope icon. At the bottom is a 'View All' button. The right screen is also titled 'Maintain Student' and has a 'View Student' button at the top. It features a person icon. A message box displays 'Student 1 - 401' with a phone icon, '9999999401' with a call icon, and 'emailstu1@gmail.com' with an envelope icon. At the bottom is a 'View All' button.

VIII. MAINTAIN FACULTY DETAILS

This form will be accessible only to the system administrator. It will allow him/her to update/delete/view information about faculty in the university.

The image shows a mobile screen prototype titled 'Admin' at the top. Below it is a profile picture and the name 'Anmolpreet'. To the right are icons for email (axndtu@gmail.com) and phone (8826225725). The main area contains four rounded rectangular buttons: 'Maintain Student Records', 'Maintain Faculty Records', 'Maintain Degree Records', and 'Maintain Course Records'. Below these are three buttons: 'Update', 'View', and 'Delete'. A text input field for 'Faculty Id' with placeholder 'Enter Id. [3XX]' is followed by a 'View Faculty' button.

Various fields available on the following form will be:

- **Faculty ID:** Numeric of length 3 digits and cannot contain alphabets, special characters and blank spaces and should be of the format 3XX.
- **Name:** Alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.
- **Phone No. :** Numeric and must have 10 digits.

UPDATE FACULTY FORM:

Maintain Faculty

Update Faculty

Faculty Name

Faculty Phone No.

Update Faculty

DELETE FACULTY FORM:

Maintain Faculty

Delete Faculty

Are you sure, you want to delete ?
Then press delete

Delete Faculty

VIEW FACULTY FORM:

Maintain Faculty

View Faculty

Faculty 1 - 301
9999999301
emailfac1@gmail.com

View All

3.1.1 HARDWARE INTERFACES

As stated in section 2.1.3.

3.1.2 SOFTWARE INTERFACES

As stated in section 2.1.4.

3.1.3 COMMUNICATION INTERFACES

None

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 LOGIN

A. USE CASE DESCRIPTION

Introduction: This use case documents the steps that must be followed to access the system.
Actors: Administrator Student Faculty
Precondition: The administrator/student/faculty must have a valid User ID and password.
Postcondition: If the use case is successful, the administrator/student/faculty is granted access to the system. If not, the system state remains unchanged. If the actor has the role ‘Administrator’, he/she will have access to only screens corresponding to admin privileges. If the actor has the role ‘Student’, he/she will have access to only screens corresponding to student privileges. If the actor has the role ‘Faculty’, he/she will have access to only screens corresponding to faculty privileges.
Flow of Event: Basic Flow : This use case starts when the administrator/student/faculty wishes to Login to the University Registration System(URS). 1. The system requests that the administrator/student/faculty enters his/her user ID, password and role.The role can be anyone one of Admin, Student and Faculty. 2. Administrator/student/faculty selects his/her role and enters user ID and password. 3. The administrator/student/faculty’s user ID and password are authenticated with the system’s database and credentials are validated ,then he/she is granted access to the system.
Alternative Flows: Alternative Flow 1: Invalid User ID/Password If in the Basic Flow, the system does not validate the administrator/student/faculty’s user ID or password (due to the user’s credentials not being in the system database or user entering invalid credentials) or the user ID or password field is left empty, then an appropriate error message is flagged and the use case returns to the beginning of the basic flow. Alternative Flow 2: User exits This allows the user to exit at any time during the use case. The use case ends.
Special Requirements: None
Associated use cases: None

B. VALIDITY CHECKS

- Every user will have an unique user ID.
- User ID cannot be blank.
- User ID is the email ID of the user and should contain one @ and . character.

- User ID will not accept special characters and blank spaces.
- Password cannot be blank.
- Length of password can only be 8 to 16 alphanumeric characters.
- Alphabets, digits, hyphen and underscore characters are allowed in the password field.
- Passwords will not accept blank spaces.

C. SEQUENCING INFORMATION

None

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.2. INITIAL REGISTRATION

A. USE CASE DESCRIPTION

Introduction: This use case documents the steps that must be followed for students/faculty to register in the system.
Actors: Student Faculty
Precondition: Student must have a valid Student ID corresponding to successful admission in the University. Faculty must be enlisted in the university with a corresponding valid Faculty ID. The university must update the required database containing the Student IDs of enrolled students in a particular degree and Faculty IDs of the faculties onto the system before the use case begins which is used for verification.
Postcondition: If the use case is successful, the student/faculty is successfully registered in the system after successful validation. Otherwise, the system state remains unchanged.
Flow of Event: Basic Flow : This use case starts when the student/faculty wishes to initially register themselves in the University database. <ol style="list-style-type: none"> 1. The system requests that the student/faculty specify the function he/she would like to perform-Student Registration or Faculty Registration. Students perform Student Registration and faculty performs Faculty Registration. 2. Once the student/faculty provides the requested information, one of the sub flows is executed. <ul style="list-style-type: none"> ● The student selects ‘Student Registration’, the Student Registration subflow is executed. ● The faculty selects ‘Faculty Registration’, the Faculty Registration subflow is executed. Basic Flow1: Student Registration: The student shall enter his personal information to register himself/herself. This include- <ul style="list-style-type: none"> ● Name- It should not have any special characters except blank spaces ● Phone Number - Numeric data of length 10. ● Email ID - A valid email format. ● Password - The student shall enter a password, alphanumeric of length in the range of 4 to 15 characters. Blank spaces are not allowed. However, special characters are allowed. ● Verify Password- The student shall enter the same password again and the password should match

- Student ID- Issued by the University corresponding to successful admission in the University
- Select Degree - The student selects the degree he/she is enrolled in.

Basic Flow 2:Faculty Registration:

The faculty shall enter his personal information to register himself/herself. This include-

- Name - It should not have any special characters except blank spaces.
- Phone Number - Numeric data of length 10.
- Email ID - A valid email format.
- Password - The faculty shall enter a password, alphanumeric of length in the range of 4 to 15 characters. Blank spaces are not allowed. However, special characters are allowed.
- Verify Password- The faculty shall enter the password again and the password should match
- Faculty ID- Issued by the University on hiring.
- Select Course - The faculty selects the course he/she has qualification in.

Once the student/faculty fills all the information in the registration form, the form is validated by the system in order to successfully register in the database and create an user account in the system.

- If all the details filled are verified and are correct , then the student/faculty are successfully registered into the Student/Faculty database .
- The account is created successfully on the system , the system provides a Registration Confirmation Slip via email to the registered student/faculty.

Alternative Flows:

Alternative Flow 1: Validation Fails:

If in the Basic Flow, the validation of the student/faculty registration form fails due to incorrect details or invalid name/phone number/email/student ID/faculty ID/password/verify password descriptor or when the student selects a degree he/she is not enrolled in or when the faculty selects a course he/she does not teach or leaves any of the descriptor empty, then an appropriate error message is displayed. The student/faculty remains unregistered and are returned to the beginning of ‘Initial Registration’ use case to re-enter the correct details.

Alternative Flow 2: Password Does Not Match :

If in the Basic Flow, if the password entered by the student/faculty does not match the verify password field, then an appropriate error message is displayed. The student/faculty remains unregistered and are required to re-enter the passwords.

Alternative Flow 3: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None

Associated use cases: None

B. VALIDITY CHECKS

- Every student must have an unique Student ID and every faculty must have an Faculty ID.
- Student ID and Faculty ID will not accept alphabets, special characters and blank spaces.
- Student ID and Faculty ID cannot be blank.
- Student ID should be in the format 4XX.
- Faculty ID should be in the format 3XX.
- Student name/ faculty name cannot be blank.

- Length of student name /faculty name can be of 3 to 50 characters.
- Student name/faculty name will only accept alphabetic characters and blank spaces.
- Phone no. cannot be blank.
- Phone no. cannot include alphabets, special characters and blank spaces.
- Phone no. must be up to 10 digits.
- Email cannot be blank.
- Email can have up to 50 characters.
- Email should contain @ and . characters.
- Email cannot include blank spaces.
- Select Degree cannot be left blank and must be chosen from the drop list.
- Select Course cannot be left blank and must be chosen from the drop list.
- Password cannot be blank
- Password can have length from 8 to 16 characters.
- Alphabets, digits, hyphen and underscore characters are allowed in the password field. However, blank spaces are not allowed.
- Password and verify password fields must be the same.

C. SEQUENCING INFORMATION

Student and faculty details corresponding to Student ID and Faculty ID respectively should be available in the system for validation.

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.3 PROFILE MANAGEMENT

A.USC CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed in order to manage user profiles. This includes viewing profiles, updating personal information, changing password and logging out of the system.

Actors:

Administrator
Student
Faculty

Precondition: The administrator/student/faculty must be logged onto the system.

Postcondition: If the use case is successful, the administrator/student/faculty shall be able to view profile, update personal information, change password and logout of the system. Otherwise, the system state shall remain unchanged.

Flow of Event:

Basic Flow :

This use case starts when the administrator/student/faculty wishes to manage their profile i.e. update personal information , change password and logout of the system.

1. The system requests the administrator/student/faculty specify the function he/she would like to perform- Update personal information, change password and logout of the system.

- Once the administrator/student/faculty provides the requested information, one of the sub flows is executed.
 - If the administrator/student/faculty selects ‘View Profile’, the **View Profile subflow is executed.**
 - If the administrator/student/faculty selects ‘Update Personal Information’, the **Update Personal Information subflow is executed.**
 - If the administrator/student/faculty selects ‘Change Password’, the **Change Password subflow is executed.**
 - If the administrator/student/faculty selects ‘Logout’, the **Logout subflow is executed.**

Basic Flow 1:View Profile:

The student can view the following in his/her profile-

- Student ID
- Name
- Email ID
- Phone Number
- Degree Name he/she is enrolled in.
- No. of Courses enrolled by the student.
- Credits attained by the student

The faculty can view the following in his/her profile-

- Faculty ID
- Name
- Email ID
- Phone Number
- Course Name taught by him/her.
- No. of students taught by the faculty.

The administrator can view the following in his/her profile-

- Name
- Email
- Phone Number

Basic Flow 2:Update Personal Information:

The system allows the administrator/student/faculty to change their personal information. This includes:

- Phone Number
- Email ID

Basic Flow 3:Change Password:

The system allows the administrator/student/faculty to change their password.

- The system requests the user to enter their old password, new password and verify new password.
- The user enters the old password along with a new password and verify new password.
- The confirm new password should match the new password in order to successfully change the password

Basic Flow 4:Logout:

The system allows the administrator/student/faculty to logout of the system.

Alternative Flows:**Alternative Flow 1: Invalid Entry**

In the ‘Update Personal Information’ subflow, the administrator/student/faculty enters invalid phone number/email ID descriptor or leaves the phone number/email descriptor empty or in the ‘Change Password’ subflow, if the user enters invalid old password, new password or the new password does not match the confirm new password, , the system displays an appropriate error message. The user returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None**Associated use cases:** Login**B. VALIDITY CHECKS**

- Every student/faculty will have a unique identification- Student ID for student and Faculty ID for faculty.
- Phone no. cannot be blank.
- Phone no. cannot include alphabets, special characters and blank spaces.
- Phone no. must be up to 10 digits.
- Email cannot be blank.
- Email can have up to 50 characters.
- Email should contain @ and . characters.
- Email cannot include blank spaces.
- Password cannot be blank
- Password can have length from 8 to 16 characters.
- Alphabets, digits, hyphen and underscore characters are allowed in the password field. However, blank spaces are not allowed.
- Password and confirm password field must be the same.

C. SEQUENCING INFORMATION

Student and faculty personal details should be available in the system.

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.4. MAINTAIN DEGREE DETAILS**A. USE CASE DESCRIPTION:**

Introduction: This use case documents the steps that must be followed in order to maintain the degree and course database. This includes adding, changing, deleting and viewing degree information from the system.

Actors:

Administrator

Precondition: The administrator must be logged onto the system .

Postcondition: If the use case is successful, the degree information is added/updated/deleted/viewed from the system database. Otherwise, the system state remains unchanged.

Flow of Event:

Basic Flow:

This use case starts when the administrator wishes to add/update/delete/view degree information.

1. The system requests that the administrator specify the function he/she would like to perform(either Add a degree, Update a degree, Delete a degree, View a degree)
2. Once the administrator provides the requested information, one of the sub flows is executed.
 - If the Administrator selects ‘Add a Degree’, the **Add a Degree subflow is executed.**
 - If the Administrator selects ‘Update a Degree’, the **Update a Degree subflow is executed.**
 - If the Administrator selects ‘Delete a Degree’, the **Delete a Degree subflow is executed.**
 - If the Administrator selects ‘View a Degree’, the **View a Degree subflow is executed**

Basic Flow 1: Add a degree:

The system requests that the administrator enter the new degree information to be added in the database. This includes:

- Degree ID
- Name of the degree
- Student IDs of the students enrolled in the degree

Once the administrator provides the above information, the degree is added to the system degree database.

Basic Flow 2: Update a degree:

1. The system requests that the administrator enters the Degree ID.
2. The administrator enters the Degree ID.
3. The administrator makes the desired changes to the degree information. This includes any of the information specified in the Add a degree sub-flow.
4. Once the administrator updates the necessary information, the system updates the degree information with the updated information.

Basic Flow 3: Delete a degree:

1. The system requests that the administrator enters the Degree ID.
2. The administrator enters the Degree ID.
3. The system prompts the administrator to confirm the deletion of the degree record.
4. The administrator confirms the deletion.
5. The system deletes the record.

Basic Flow 1.4:View a degree:

1. The system requests that the administrator enters the Degree ID.
2. The administrator enters the Degree ID.
3. The system retrieves and displays the required degree information.

Alternative Flows:

Alternative Flow 1: Invalid Degree Entry:

If in the **Add a Degree or Update a Degree** flows, the administrator enters invalid Degree ID/degree name/Student IDs descriptor or leaves the Degree ID/degree name/Student IDs descriptor empty, the system displays an appropriate error message. The administrator returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Degree already exist:

If in the **Add a Degree**, a degree with a specified Degree ID already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the details.

Alternative Flow 3: Degree not found:

If in the **Update a Degree or Delete a Degree or View a Degree**, the degree with a specified Degree ID does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Degree ID.

Alternative Flow 4: Student Already Enrolled in other degree :

If in the **Add a Degree or Update a Degree**, the administrator enters the Student ID of a student who is already enrolled in another degree, then the system displays an error message. The administrator returns to the basic flow and may reenter the details.

Alternative Flow 5: Update cancelled:

If in the **Update a Degree** flow, the administrator decides not to Update the degree, the Update is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 6: Delete cancelled:

If in the **Delete a Degree** flow, the administrator decides not to Delete the degree, the Delete is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 7: User exits:

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None

Associated use cases: Login

B. VALIDITY CHECKS

- Only the administrator will be authorized to access the degree details module.
- Degree ID should be in the format 1XX.
- Degree ID cannot contain special characters and blank spaces.
- Degree Name should be alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.

C. SEQUENCING INFORMATION

None

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.5 MAINTAIN COURSE DETAILS

A. USE CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed in order to maintain the degree and course database. This includes adding, changing, deleting and viewing course information from the system.

Actors:
Administrator

Precondition: The administrator must be logged onto the system .

Postcondition: If the use case is successful, the course information is added/updated/deleted/viewed from the system database.Otherwise, the system state remains unchanged.

Flow of Event:

Basic Flow:

This use case starts when the administrator wishes to add/update/delete/view course information.

1. The system requests that the administrator specify the function he/she would like to perform(either Add a course, Update a course, Delete a course, View a course)
2. Once the administrator provides the requested information, one of the sub flows is executed.
 - If the Administrator selects ‘Add a Course’, the **Add a Course subflow is executed.**
 - If the Administrator selects ‘Update a Course’, the **Update a Course subflow is executed.**
 - If the Administrator selects ‘Delete a Course’, the **Delete a Course subflow is executed.**
 - If the Administrator selects ‘View a Course’, the **View a Course subflow is executed**

Basic Flow 1: Add a course:

The system requests that the administrator enter the new course information to be added in the database.This includes:

- Course ID
- Name of the course
- Faculty ID of the Faculty teaching this course
- No. of Credits
- No. of seats
- Student IDs of the students enrolled in the course

Once the administrator provides the above information, the course is added to the system course database.

Basic Flow 2: Update a course:

1. The system requests that the administrator enters the Course ID.
2. The administrator enters the Course ID.
3. The system retrieves and displays the course information.
4. The administrator makes the desired changes to the course information.This includes any of the information specified in the **Add a course sub-flow.**
5. Once the administrator updates the necessary information, the system updates the course information with the updated information.

Basic Flow 3: Delete a course:

1. The system requests that the administrator enters the Course ID.
2. The administrator enters the Course ID.The system retrieves and displays the required course information.
3. The system prompts the administrator to confirm the deletion of the course record.
4. The administrator verifies the deletion.
5. The system deletes the record.

Basic Flow 4:View a course:

1. The system requests that the administrator enters the Course ID.

2. The administrator enters the Course ID. The system retrieves and displays the required course information.

Alternative Flows:

Alternative Flow 1: Invalid Course Entry:

If in the **Add a Course or Update a Course** flows, the actor enters invalid Course ID/course name/ descriptor or leaves the Course ID/course name/ descriptor empty, the system displays an appropriate error message. The administrator returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Course already exist:

If in the **Add a Course**, a degree/course with a specified Course ID already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the details.

Alternative Flow 3: Course not found:

If in the **Update a Course or Delete a Course or View a course**, the course with a specified Course ID does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Course ID.

Alternative Flow 4: Update cancelled:

If in the **Update a Course** flow, the administrator decides not to Update the course, the Update is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 5: Delete cancelled:

If in the **Delete a Course**, the administrator decides not to Delete the course, the Delete is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 6: No. of students enrolled in a course exceed no. of seats:

If in the **Add a Course or Update a Course** flows, the administrator enters more Student IDs than the no. of seats available in the course, then the system displays an error message. The administrator returns to the basic flow and may reenter the details.

Alternative Flow 7: User exits:

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None

Associated use cases: Login

B. VALIDITY CHECKS

- Only the administrator will be authorized to access the course details module.
- Course ID should be of the format 2XX.
- Course ID cannot contain special characters and blank spaces.
- Course Name should be alphanumeric, with length 3 to 50 characters. Blank spaces are allowed. Special characters are not allowed.

C. SEQUENCING INFORMATION

None

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.6 MAINTAIN STUDENT DETAILS

A.USE CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed in order to maintain the student and faculty database. This includes changing, deleting and viewing student information from the system.

Actors:

Administrator

Precondition: The administrator must be logged onto the system .

Postcondition: If the use case is successful, the student information is updated/deleted/viewed from the system database. Otherwise, the system state remains unchanged.

Flow of Event:

Basic Flow:

This use case starts when the administrator wishes to update/delete/view student information.

1. The system requests that the administrator specify the function he/she would like to perform(either Update a student, Delete a student, View a student)
2. Once the administrator provides the requested information, one of the sub flows is executed.
 - If the Administrator selects ‘Update a Student’, the **Update a Student subflow is executed.**
 - If the Administrator selects ‘Delete a Student, the **Delete a Student subflow is executed.**
 - If the Administrator selects ‘View a Student’, the **View a Student subflow is executed**

Basic Flow 1: Update a student:

1. The system requests that the administrator enters the Student ID.
2. The administrator enters the Student ID.
3. The administrator makes the desired changes to the student information. This includes student name and phone no.
4. Once the administrator updates the necessary information, the system updates the student information with the updated information.

Basic Flow 2: Delete a student:

1. The system requests that the administrator enters the Student ID.
2. The administrator enters the Student ID.
3. The system prompts the administrator to confirm the deletion of the student record.
4. The administrator verifies the deletion.
5. The system deletes the record.

Basic Flow 3:View a student:

1. The system requests that the administrator enters the Student ID.
2. The administrator enters the Student ID.
3. The system retrieves and displays the required student information.

Alternative Flows:

Alternative Flow 1: Invalid Student Entry:

If in the **Update a Student** sub-flow, the administrator enters invalid Student ID/name/phone no. descriptor or leaves the Student ID/name/phone no. descriptor empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry.

Alternative Flow 2: Student not found:

If in the **Update a Student or Delete a Student or View a Student**, a student with a specified Student ID does not exist in the database, the system displays an error message. The administrator returns to the basic flow and may re-enter the Student ID.

Alternative Flow 3: Update cancelled:

If in the **Update a Student** sub-flow, the administrator decides not to Update the student, Updation is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 4: Delete cancelled:

If in the **Delete a Student**, the administrator decides not to Delete the student, Deletion is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 5: User exits:

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None

Associated use cases: Login

B. VALIDITY CHECKS

- Only the administrator will be authorized to access the Student Details module.
- Every student has an unique identification- Student ID.
- Student ID will not accept alphabets, special characters and blank spaces.
- Student ID cannot be blank
- Student ID should be of the format 4XX.
- Student name cannot be blank.
- Length of student name can be of 3 to 50 characters.
- Student names will only accept alphabetic characters and blank spaces.
- Phone no. cannot be blank.
- Phone no. cannot include alphabets, special characters and blank spaces.
- Phone no. must be up to 10 digits.

C. SEQUENCING INFORMATION

None

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.7 MAINTAIN FACULTY DETAILS

A.USE CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed in order to maintain the student and faculty database. This includes changing, deleting and viewing faculty information from the system.

Actors:

Administrator

Precondition: The administrator must be logged onto the system .

Postcondition: If the use case is successful, the faculty information is updated/deleted/viewed from the system database. Otherwise, the system state remains unchanged.

Flow of Event:

Basic Flow:

1. The system requests that the administrator specify the function he/she would like to perform(either Update a faculty, Delete a faculty, View a faculty)
2. Once the administrator provides the requested information, one of the sub flows is executed.
 - If the Administrator selects ‘Update a Faculty’, the **Update a Faculty subflow is executed.**
 - If the Administrator selects ‘Delete a Faculty’, the **Delete a Faculty subflow is executed.**
 - If the Administrator selects ‘View a Faculty’, the **View a Faculty subflow is executed**

Basic Flow 1: Update a faculty:

1. The system requests that the administrator enters the Faculty ID.
2. The administrator enters the Faculty ID.
3. The administrator makes the desired changes to the faculty information. This includes Faculty name and Phone no.
4. Once the administrator updates the necessary information, the system updates the faculty information with the updated information.

Basic Flow 2: Delete a faculty:

1. The system requests that the administrator enters the Faculty ID.
2. The administrator enters the Faculty ID.
3. The system prompts the administrator to confirm the deletion of the faculty record.
4. The administrator verifies the deletion.
5. The system deletes the record.

Basic Flow 3:View a faculty:

1. The system requests that the administrator enters the Faculty ID.
2. The administrator enters the Faculty ID.
3. The system retrieves and displays the required faculty information.

Alternative Flows:

Alternative Flow 1: Invalid Faculty Entry:

If in the **Update a Faculty** flows, the administrator enters invalid Faculty ID/name/phone no. descriptor or leaves the Faculty ID/name/phone no. descriptor empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry.

Alternative Flow 2: Faculty not found:

If in the **Update a Faculty or Delete a Faculty or View a Faculty**, a faculty with a specified Faculty ID does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Faculty ID.

Alternative Flow 3: Update cancelled:

If in the **Update a Faculty** sub-flow, the administrator decides not to Update the faculty, updation is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 4:Delete cancelled:

If in the **Delete a Faculty**, the administrator decides not to Delete the faculty, Deletion is cancelled and the Basic Flow is re-started at the beginning.

Alternative Flow 5: User exits:

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None**Associated use cases:** Login**B. VALIDITY CHECKS**

- Only the administrator will be authorized to access the Faculty Details module.
- Every faculty has an unique identification- Faculty ID.
- Faculty ID will not accept alphabets, special characters and blank spaces.
- Faculty ID cannot be blank.
- Faculty ID should be of the format 3XX.
- Faculty name cannot be blank.
- Length of faculty name can be of 3 to 50 characters.
- Faculty names will only accept alphabetic characters and blank spaces.
- Phone no. cannot be blank.
- Phone no. cannot include alphabets, special characters and blank spaces.
- Phone no. must be up to 10 digits.

C. SEQUENCING INFORMATION

None

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.2.8 VIEW STUDENT-COURSE DETAILS**A.USE CASE DESCRIPTION:**

Introduction: This use case documents the steps that must be followed by students to view the details of the courses registered by them in the semester and the faculty to view the list of students registered in the course taught by them.

Actors:

Administrator
Student
Faculty

Precondition: The administrator/student/faculty must be logged onto the system .

Postcondition: If the use case is successful,the student shall be able to view the details of the courses registered by them in the semester and the faculty shall be able to view the list of students registered in the course taught by them.The administrator shall be able to view both details. Otherwise, the system state remains unchanged.

Flow of Event:**Basic Flow :**

This use case starts when the student wishes to view the details of the courses registered by them in the semester ;when the faculty wishes to view the list of students registered in the course taught by them and when the administrator wishes to view details related to both student and faculty.

Basic Flow 1:View Details by Student:

1. The system displays the list of courses enrolled by the student in a particular semester when the student selects the view courses option.
2. The system displays the list of courses the student is enrolled in and relevant information related to the course .This includes: Course ID, Course Name, No. of credits, No. of seats and Faculty ID of the professor who will teach the course.

Basic Flow 2:View Detail by Faculty:

1. The system displays the list of courses enrolled by the student in a particular semester when the student selects the view courses option.
2. The system displays the list of students registered in a course taught by the faculty. The system only displays the list of those students who have registered themselves on URS. This includes Student ID, name, email and phone number.

Alternative Flows:

None

Special Requirements: None

Associated use cases: Login

3.2.9 CONTACT UNIVERSITY

A.USE CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed to contact the University incase of complaint/query or submitting feedback.

Actors: Administrator Student Faculty
Precondition: None
Postcondition: If the use case is successful, the administrator/student/faculty shall be able to contact the university incase of a complaint or query and submit a feedback.
Flow of Events:
<p>Basic Flow : This use case starts when the administrator/students/faculty wishes to contact the University in case of a complaint/query or wishes to submit a feedback.</p> <ol style="list-style-type: none"> 1. The system displays an option for “Contact University” incase of complaint/query or provide feedback. 2. On selecting the Contact University option , the system displays the official email ID, contact numbers and the location of the University. The system also displays an option for submitting feedback to the University. 3. Administrator/student/faculty may contact the university to resolve their complaints and queries through email or submit feedback to the University regarding the system.
<p>Alternative Flows : Alternative Flow 1: User exits This allows the user to exit at any time during the use case. The use case ends.</p>
Special Requirements: None
Associated use cases: None

3.2.10 REPORT GENERATION

A. USE CASE DESCRIPTION:

Introduction: This use case documents the steps that must be followed in order to generate the following reports:
<ol style="list-style-type: none"> 1. Student List Report 2. Faculty List Report. 3. Degree List Report. 4. Course List Report
Actors: Administrator
Precondition: The administrator must be logged onto the system.
Postcondition: If the use case is successful, the system generates the desired report. Otherwise the system state remains unchanged.

Flow of Event:**Basic Flow :**

This use case starts when the administrator wishes to generate reports.

1. The system requests the administrator specify the report he/she would like to generate.
2. Once the administrator provides the requested information, one of the sub-flows is executed:
 - If the Administrator selected “Student List Report” the **Generate Student List Report** sub flow is executed.
 - If the Administrator selected “Faculty List Report” the **Generate Faculty List Report** sub flow is executed.
 - If the Administrator selected “Degree List Report” the **Generate Degree List Report** sub flow is executed.
 - If the Administrator selected “Course List Report” the **Generate Course List Report** sub flow is executed.

Basic Flow 1: Generate Student List Report:

1. The system requests the administrator to provide any one valid Student ID to generate the Student List report .
2. Once the administrator provides the requested information, the system generates the Student List report, containing the list of students currently enrolled in the university along with their information.

Basic Flow 2: Generate Faculty List Report:

1. The system requests the administrator to provide any one valid Faculty ID to generate the Faculty List report .
2. Once the administrator provides the requested information, the system generates the Faculty List report, containing the list of faculty currently teaching in the University along with their information

Basic Flow 3: Generate Degree List Report:

1. The system requests the administrator to provide any one valid Degree ID to generate the Degree List report .
2. Once the administrator provides the requested information, the system generates the Degree List report, containing the list of degrees offered by the University and no. of students enrolled in that degree.

Basic Flow 4: Generate Course List Report:

1. The system requests the administrator to provide any one valid Course ID to generate the CourseList report .
2. Once the administrator provides the requested information, the system generates the Course List report, containing the list of courses offered by the University and no. of students enrolled in that degree.

Alternative Flows:**Alternative Flow 1: Invalid Entry :**

If in the basic flow, the administrator enters invalid Student ID/Faculty ID/Degree ID/Course ID to generate a desired report, then an appropriate error message is displayed.The administrator is returned to the beginning of the use case.

Alternative Flow 2: User exits

This allows the user to exit at any time during the use case. The use case ends.

Special Requirements: None**Associated use cases:** Login

B. VALIDITY CHECK

- Only the administrator will be authorized to access the Report Generation module.
- Degree ID should be of the format 1XX.
- Course ID should be of the format 2XX.
- Faculty ID should be of the format 3XX.
- Student ID should be of the format 4XX.

C. SEQUENCING INFORMATION

Student, faculty, degree and course details should be available in the system.

D. ERROR HANDLING/RESPONSE TO ABNORMAL SITUATION

If any of the validation flows does not hold true, appropriate error messages will be prompted to the user for doing the needful.

3.3 PERFORMANCE REQUIREMENTS

- (a) Android version 4.0+ and IOS version 12+.
- (b) Responses should be within 2 seconds.

3.4 DESIGN CONSTRAINTS

None

3.5 SOFTWARE SYSTEM

Attributes Usability - The application will be user-friendly and easy to operate and the functions will be easily understandable.

Reliability - The applications will be available to the students throughout the registration period and have a high degree of fault tolerance.

Security - The application will be password protected. Users will have to enter the correct login ID and password to access the application.

Maintainability - The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

Portability - The application will be easily portable on any mobile system that has an Android or IOS operating system.

3.6 LOGICAL DATABASE REQUIREMENTS

The following information will be placed in a database:

- Login : Records the login details of the user.
- Student : Records Student details.
- Faculty : Records Faculty details.
- Registered: Records Registered student/faculty details.
- Admin : Records Admin details.
- Degree : Records Degree details.
- Course : Records Course details.
- Feedback: Record feedback details.

3.7 OTHER REQUIREMENTS

None

UML DIAGRAMS

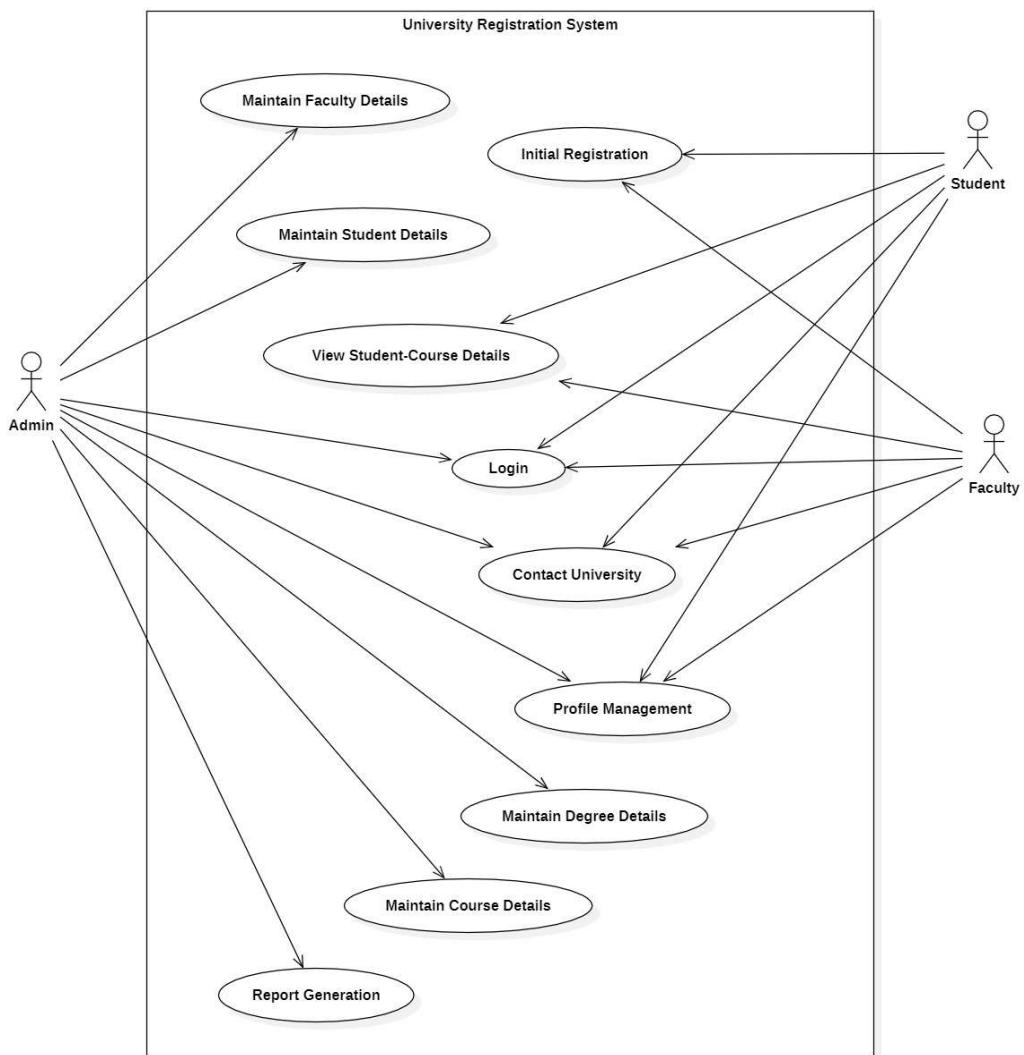
1. USE CASE DIAGRAM

ACTORS:

- ❖ Admin
- ❖ Student
- ❖ Faculty

USE CASES :

- ❖ Login
- ❖ Initial Registration
- ❖ Profile Management
- ❖ Maintain Degree Details
- ❖ Maintain Course Details
- ❖ Maintain Student Details
- ❖ Maintain Faculty Details
- ❖ View Student-Course Details
- ❖ User Account Creation
- ❖ Contact University
- ❖ Report Generation



2. CLASS DIAGRAM

Different classes identified in URS are:

INTERFACE CLASSES:

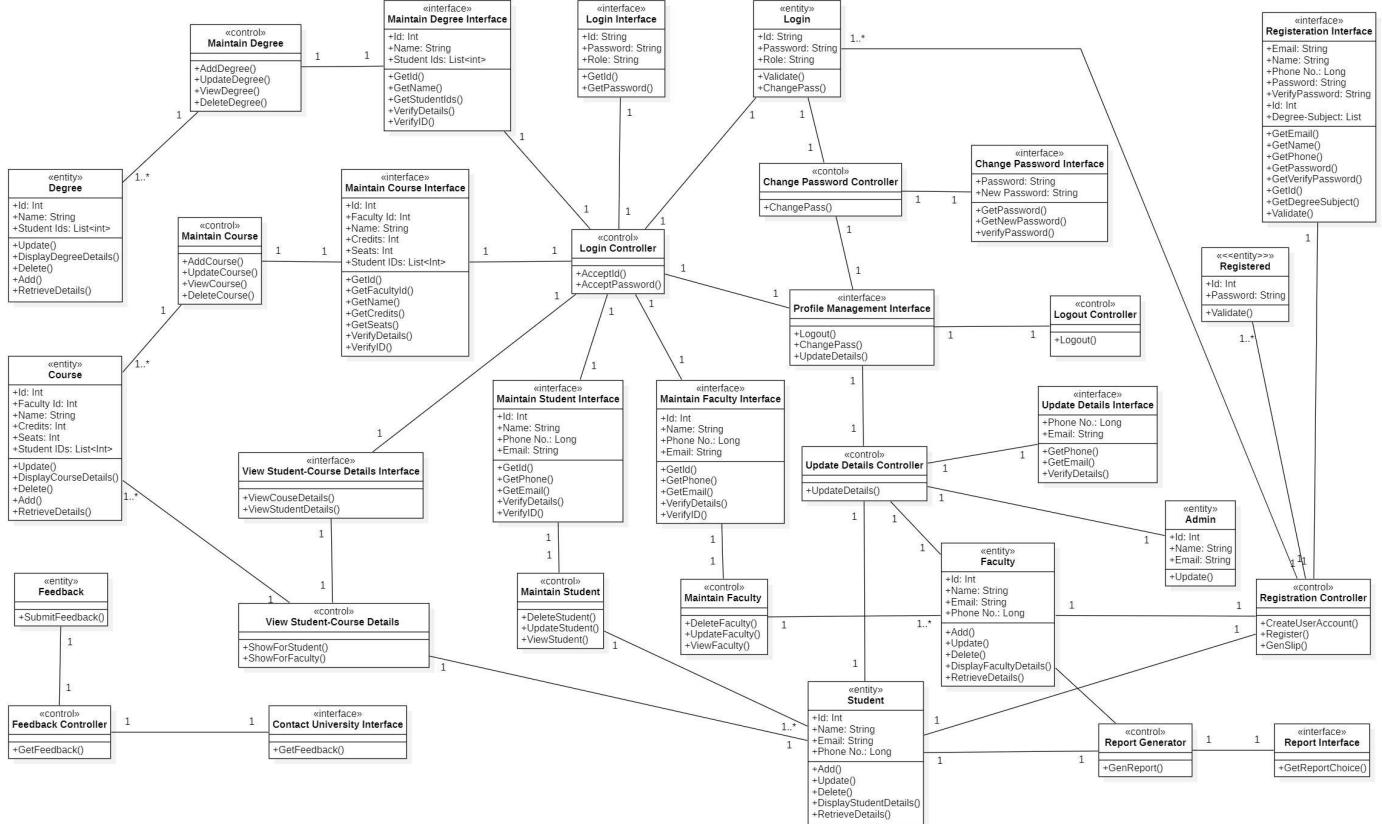
- Login Interface
 - Registration Interface
 - Maintain Degree Interface
 - Maintain Course Interface
 - Maintain Student Interface
 - Maintain Faculty Interface
 - Profile Management Interface
 - Change Password Interface
 - Update Details Interface
 - View Student-Course Details Interface
 - Contact University Interface
 - Report Interface

CONTROL CLASSES:

- Login Controller
 - Registration Controller
 - Maintain Degree Controller
 - Maintain Course Controller
 - Maintain Student Controller
 - Maintain Faculty Controller
 - Change Password Controller
 - Update Details Controller
 - View Student-Course Details Controller
 - Feedback Controller
 - Logout Controller
 - Report Generator

ENTITY CLASSES:

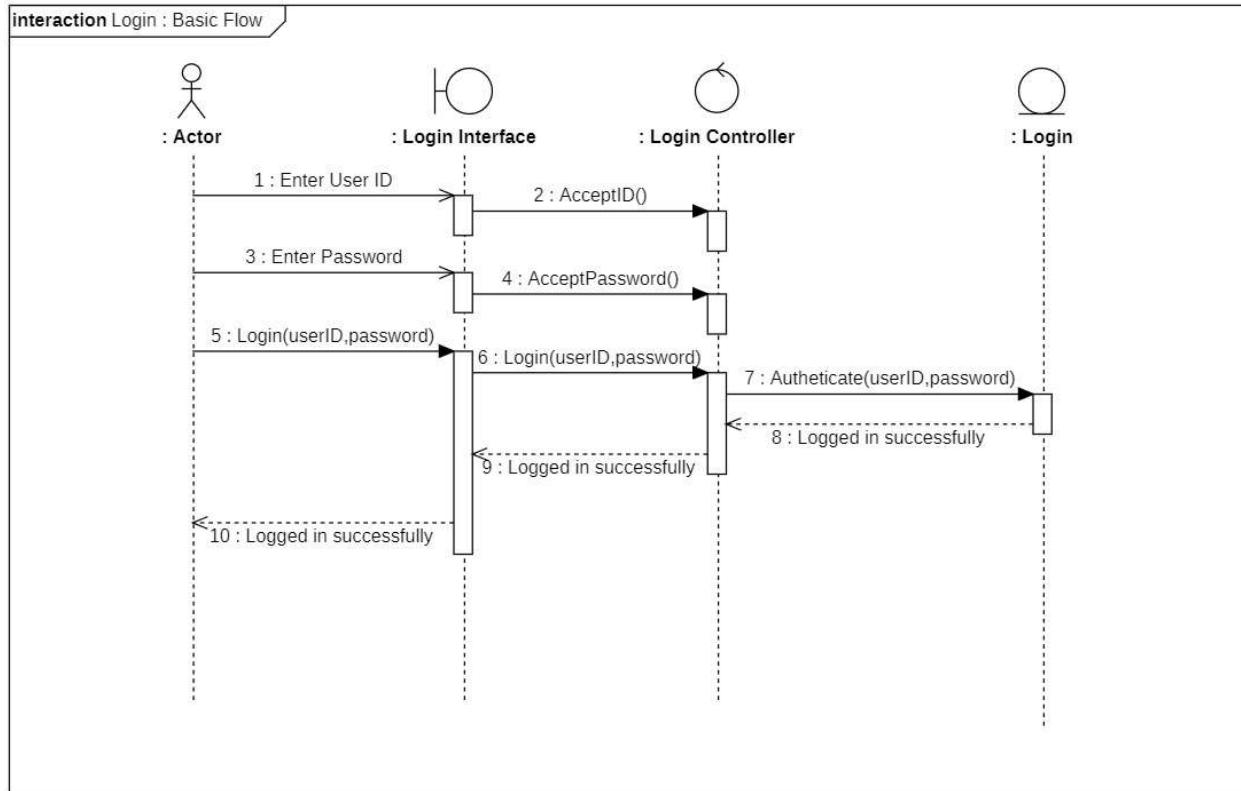
- Student
 - Faculty
 - Admin
 - Degree
 - Course
 - Login
 - Registered
 - Feedback



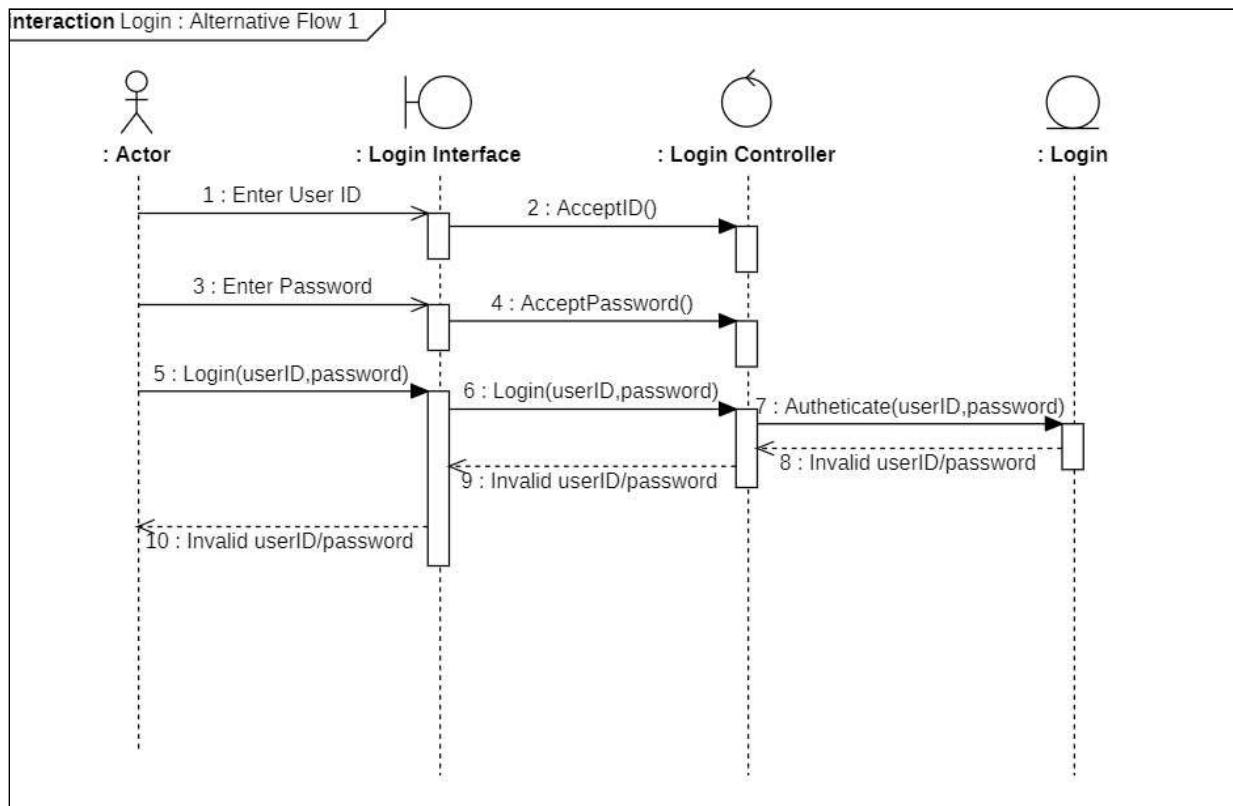
3. SEQUENCE DIAGRAM

1. LOGIN

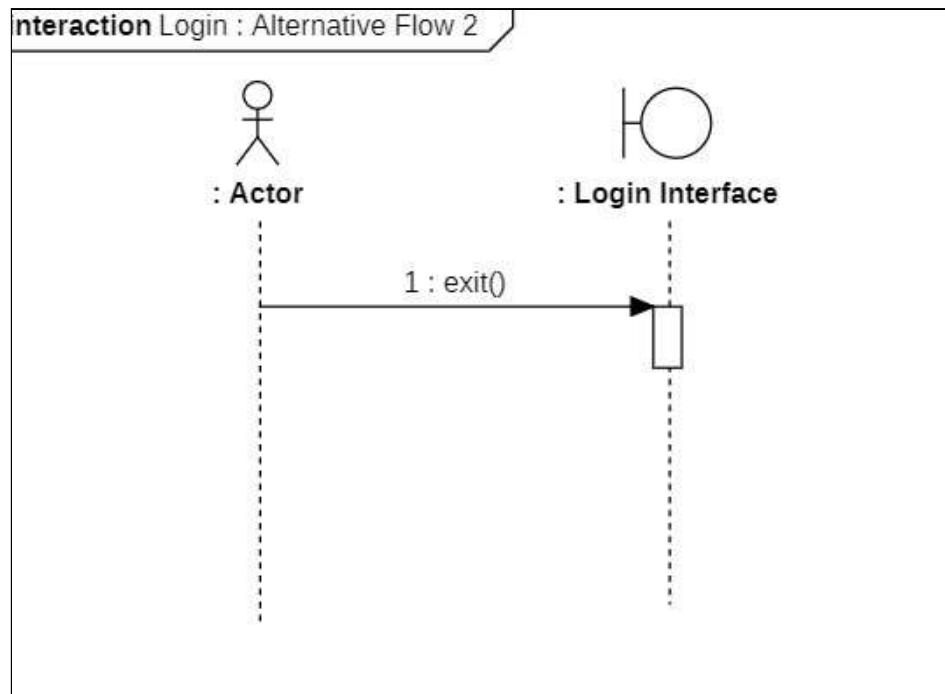
LOGIN BASIC FLOW :



LOGIN ALTERNATIVE FLOW 1 : INVALID USER ID / PASSWORD

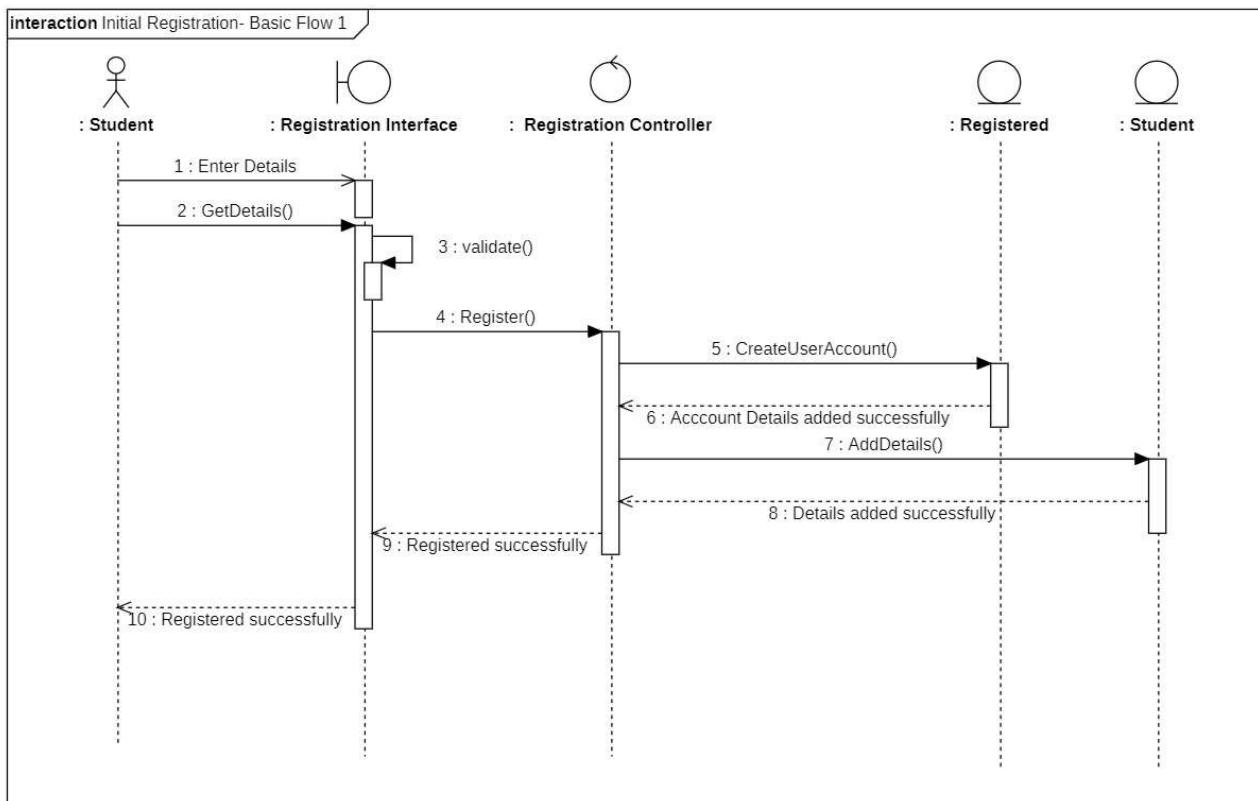


LOGIN ALTERNATIVE FLOW 2 : USER EXITS

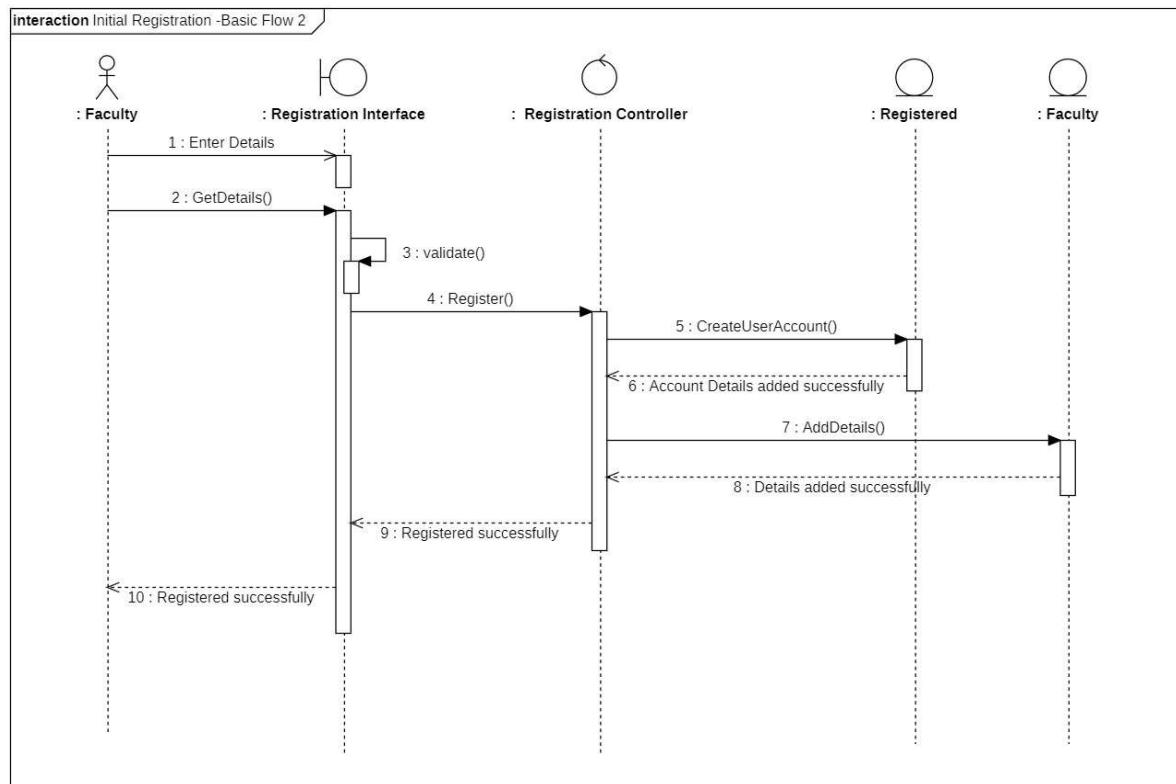


2. INITIAL REGISTRATION

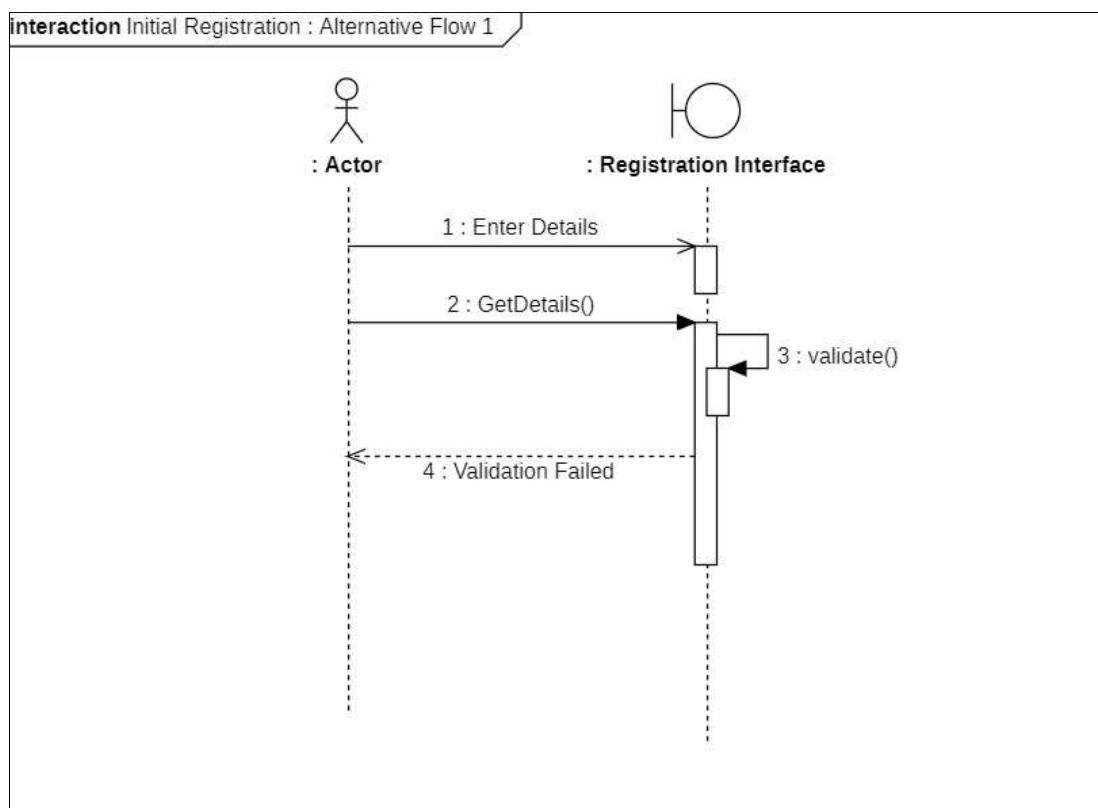
INITIAL REGISTRATION BASIC FLOW 1 : STUDENT REGISTRATION



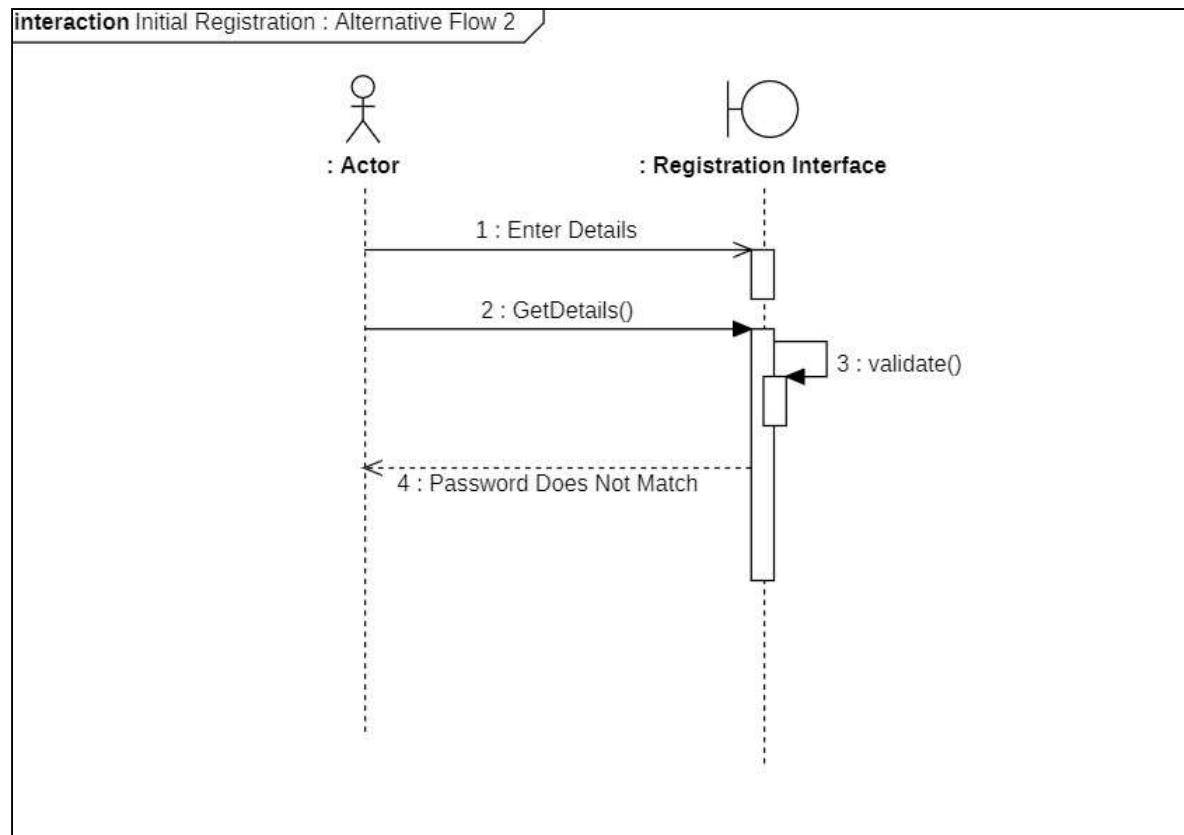
INITIAL REGISTRATION BASIC FLOW 2 : FACULTY REGISTRATION



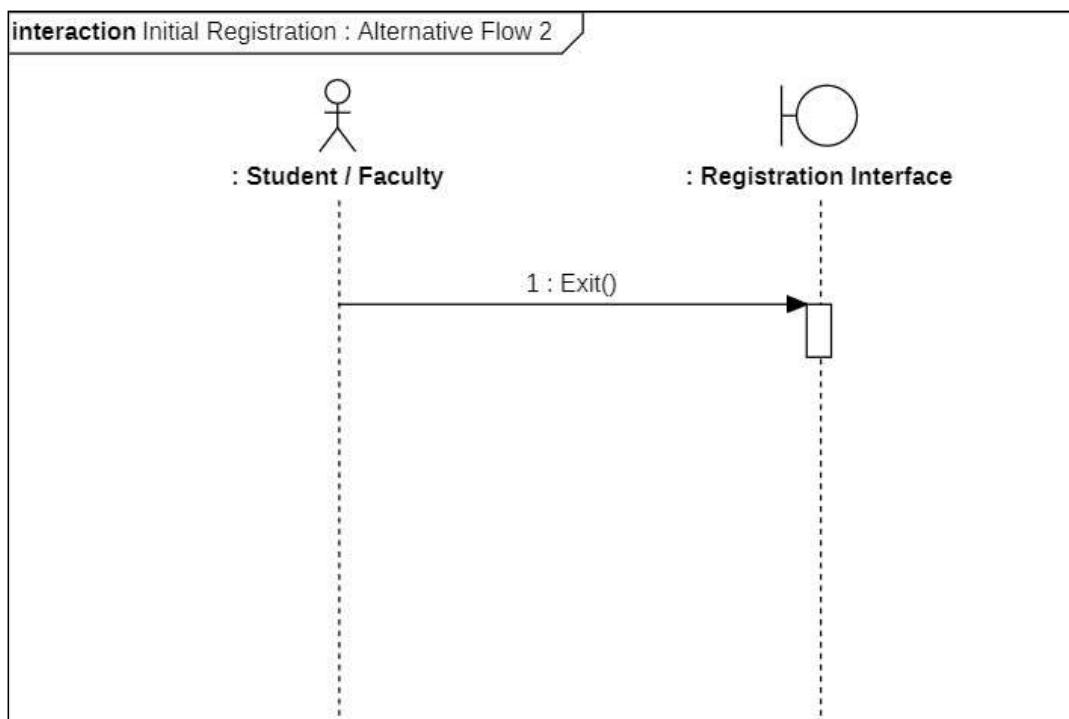
INITIAL REGISTRATION ALTERNATIVE FLOW 1 :VALIDATION FAILS



INITIAL REGISTRATION ALTERNATIVE FLOW 2 : PASSWORD DOES NOT MATCH

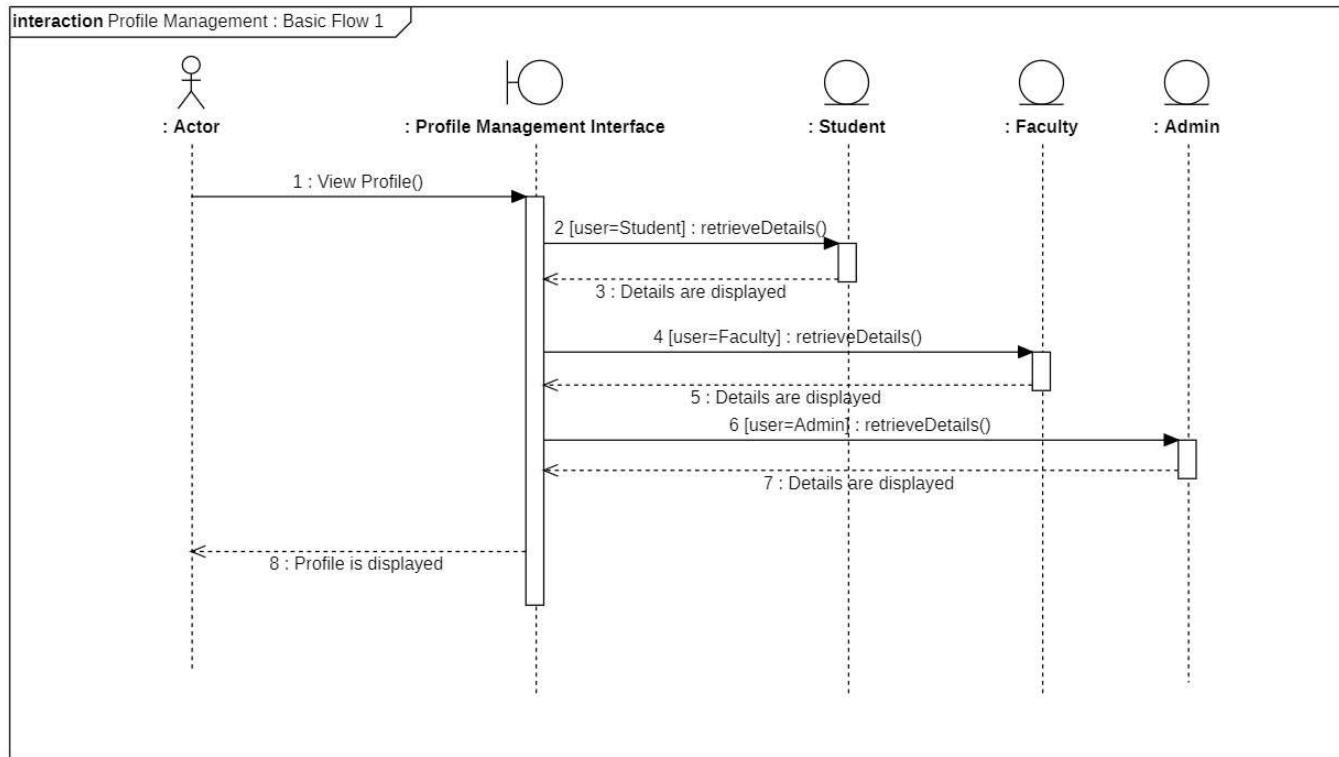


INITIAL REGISTRATION ALTERNATIVE FLOW 3 : USER EXITS

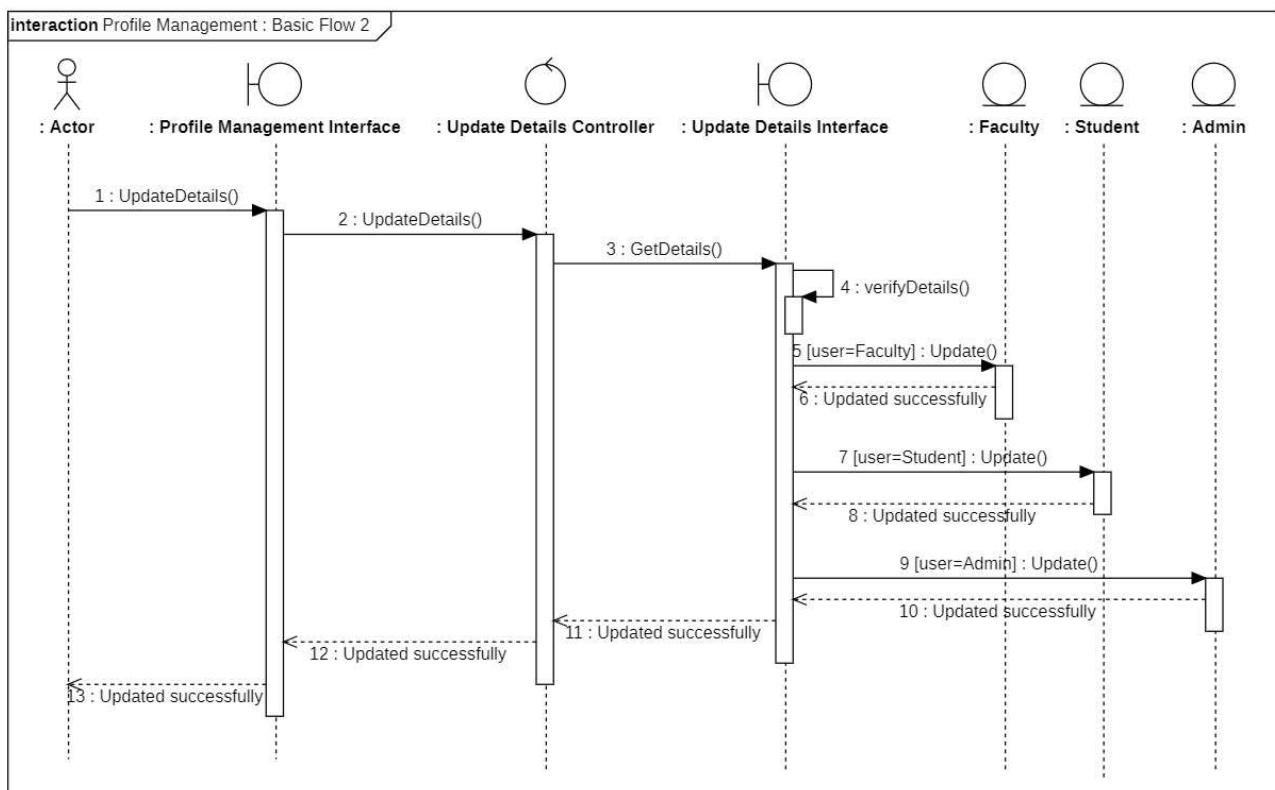


3. PROFILE MANAGEMENT

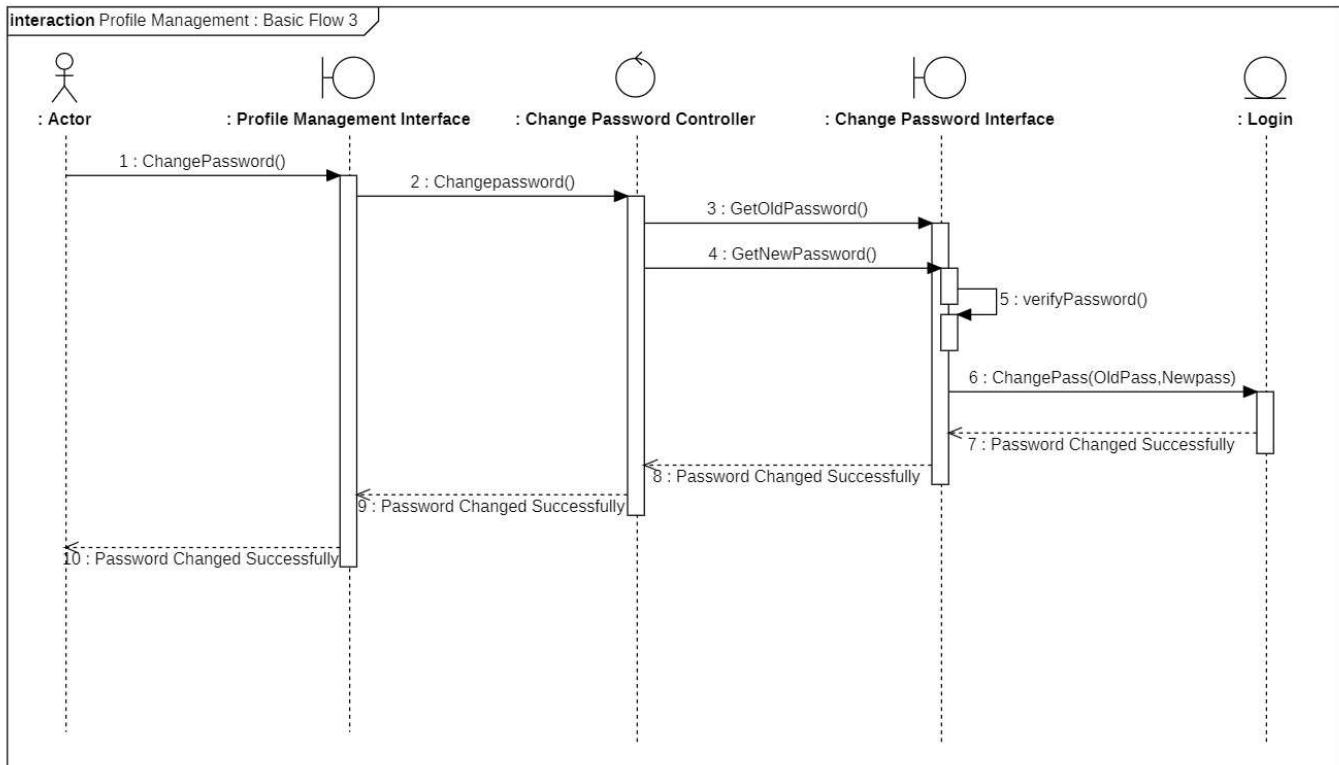
PROFILE MANAGEMENT BASIC FLOW 1: VIEW PROFILE



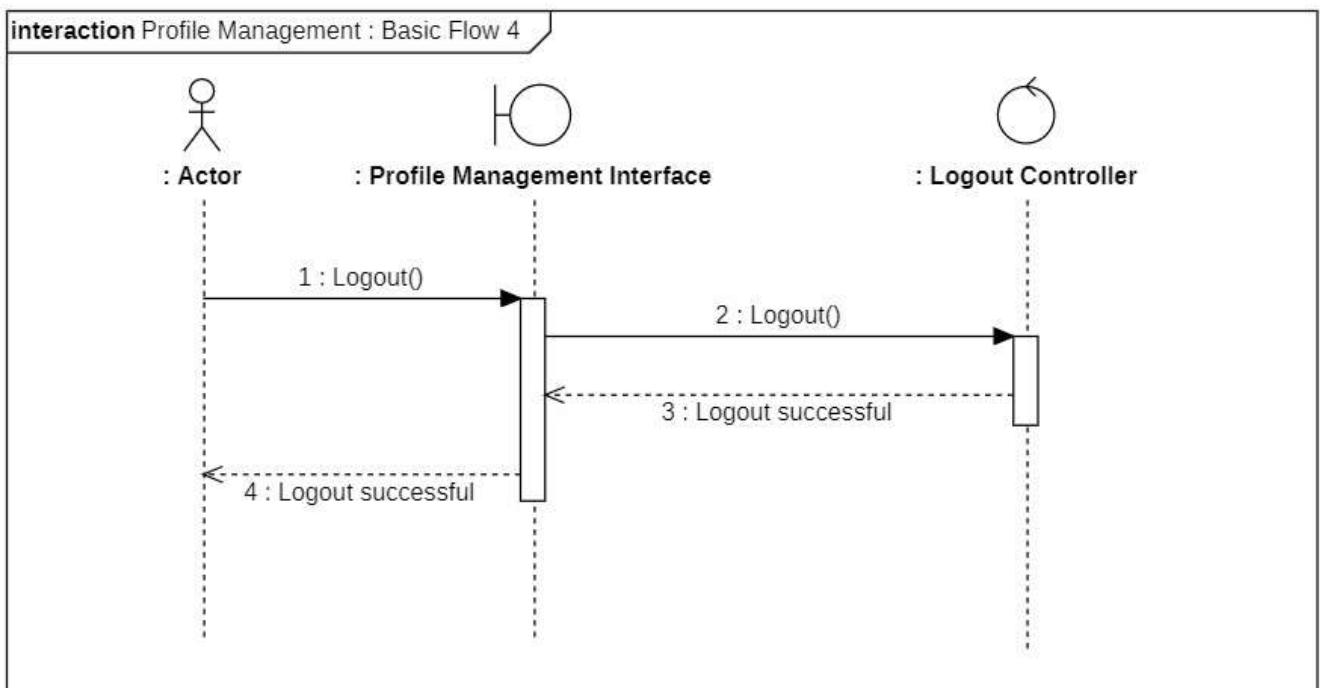
PROFILE MANAGEMENT BASIC FLOW 2: UPDATE PROFILE



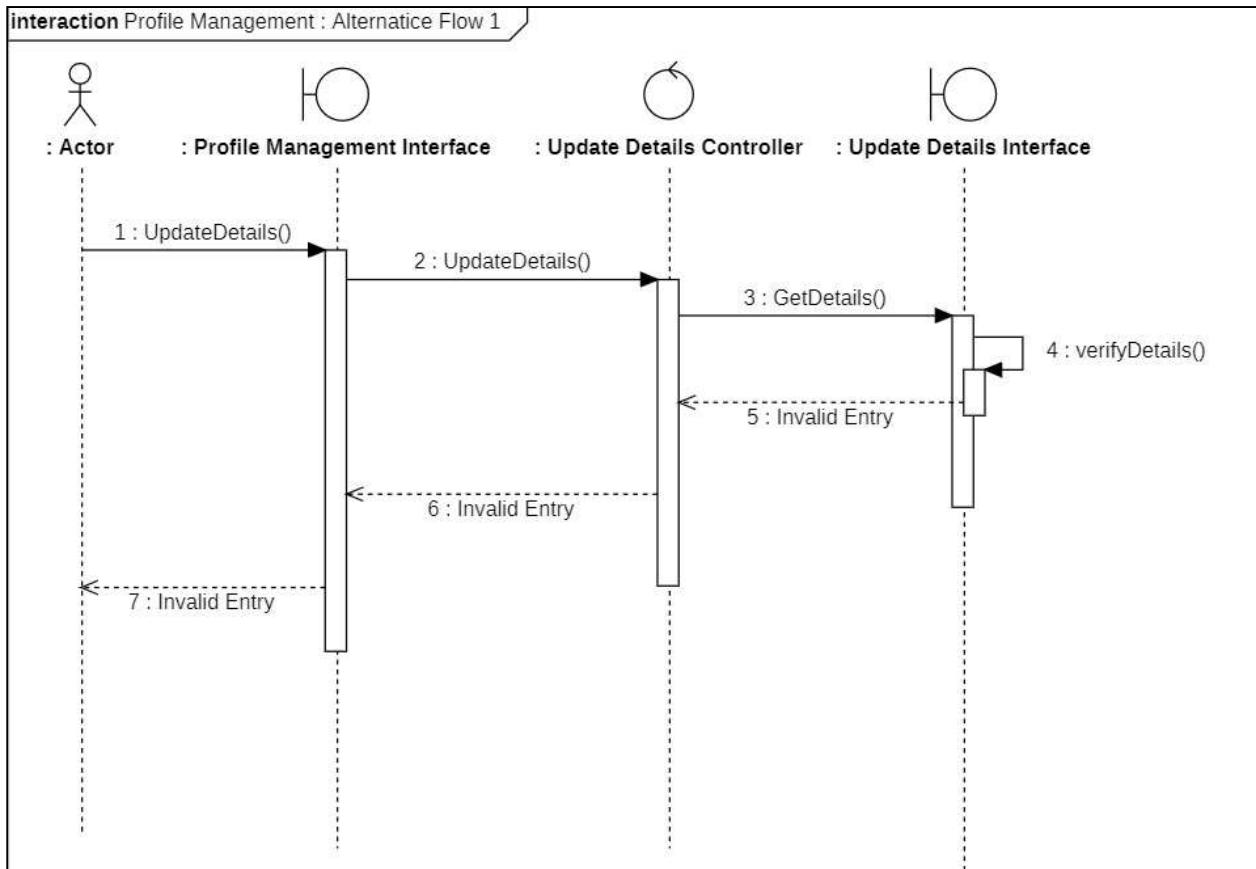
PROFILE MANAGEMENT BASIC FLOW 3: CHANGE PASSWORD



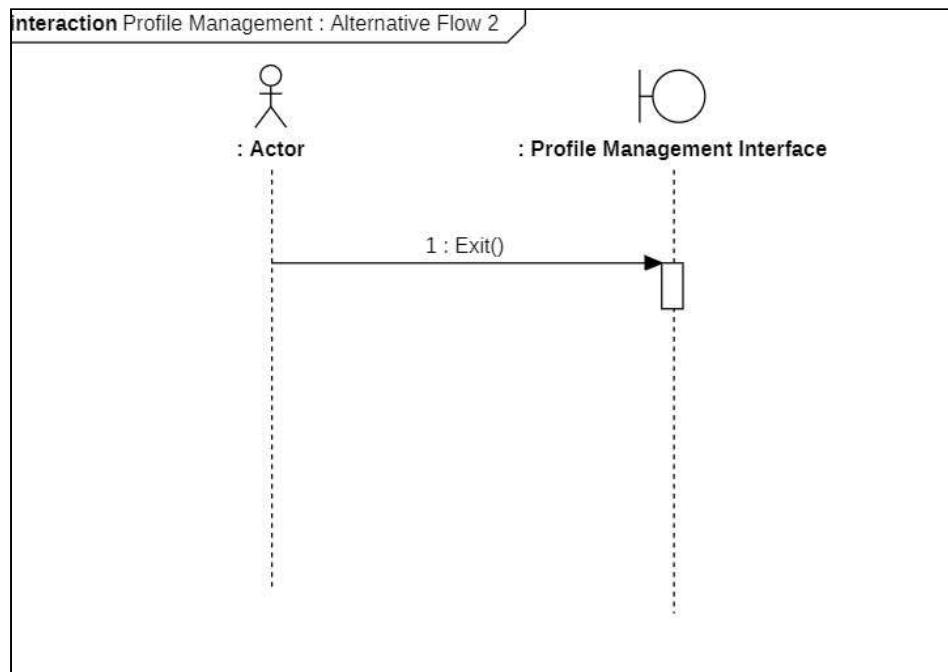
PROFILE MANAGEMENT BASIC FLOW 4: LOGOUT



PROFILE MANAGEMENT ALTERNATIVE FLOW 1: INVALID ENTRY

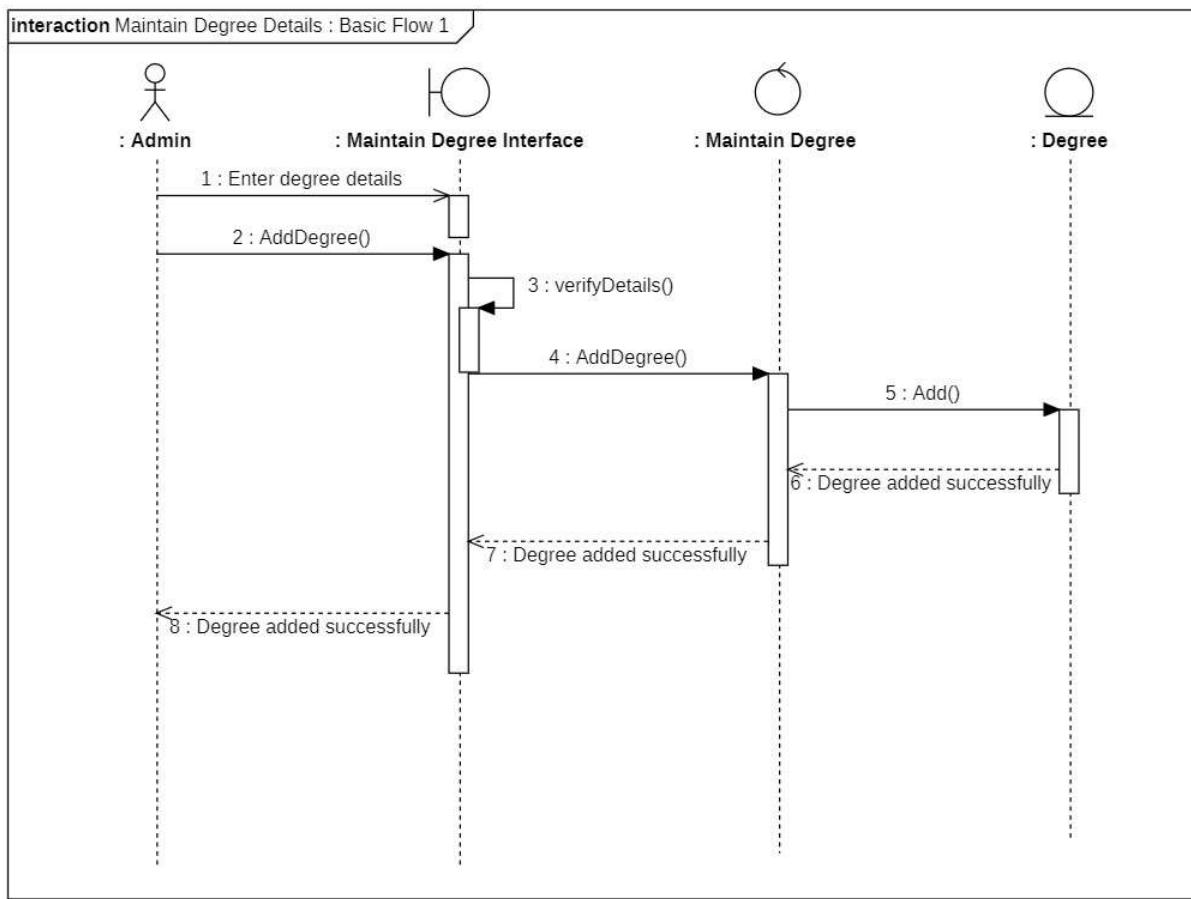


PROFILE MANAGEMENT ALTERNATIVE FLOW 2: USER EXITS

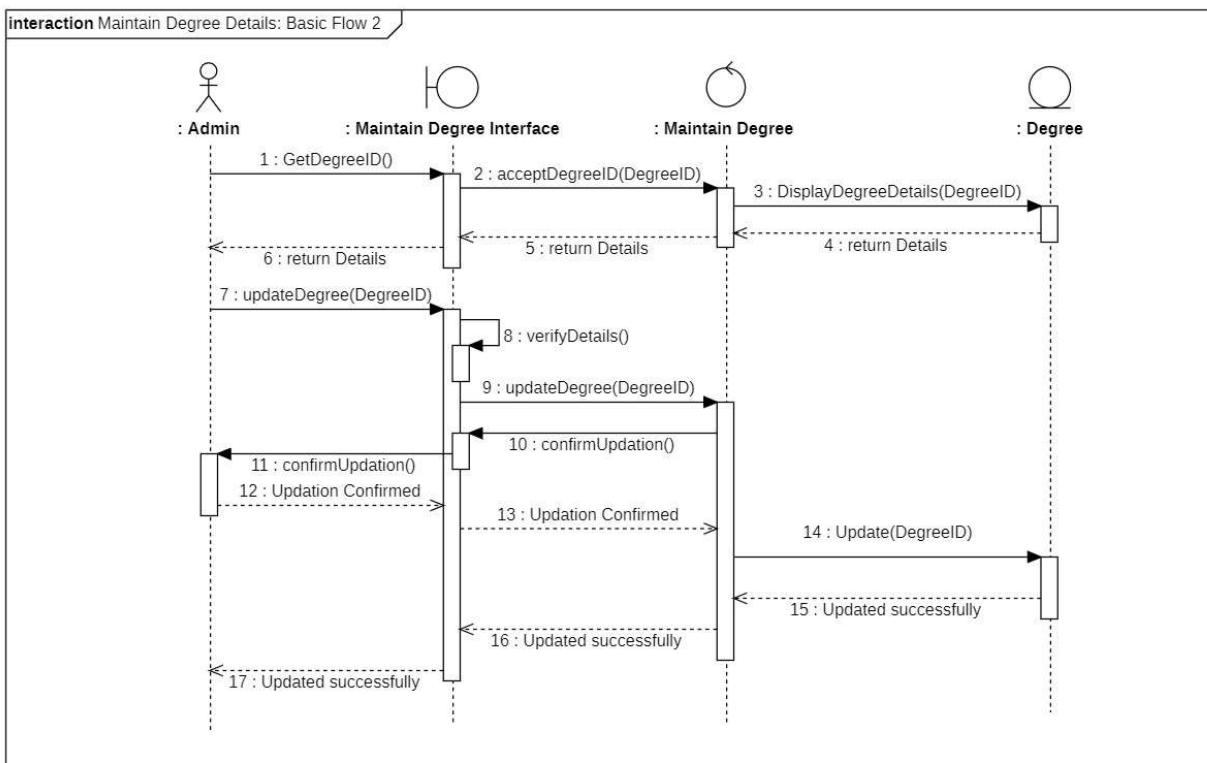


4. MAINTAIN DEGREE DETAILS

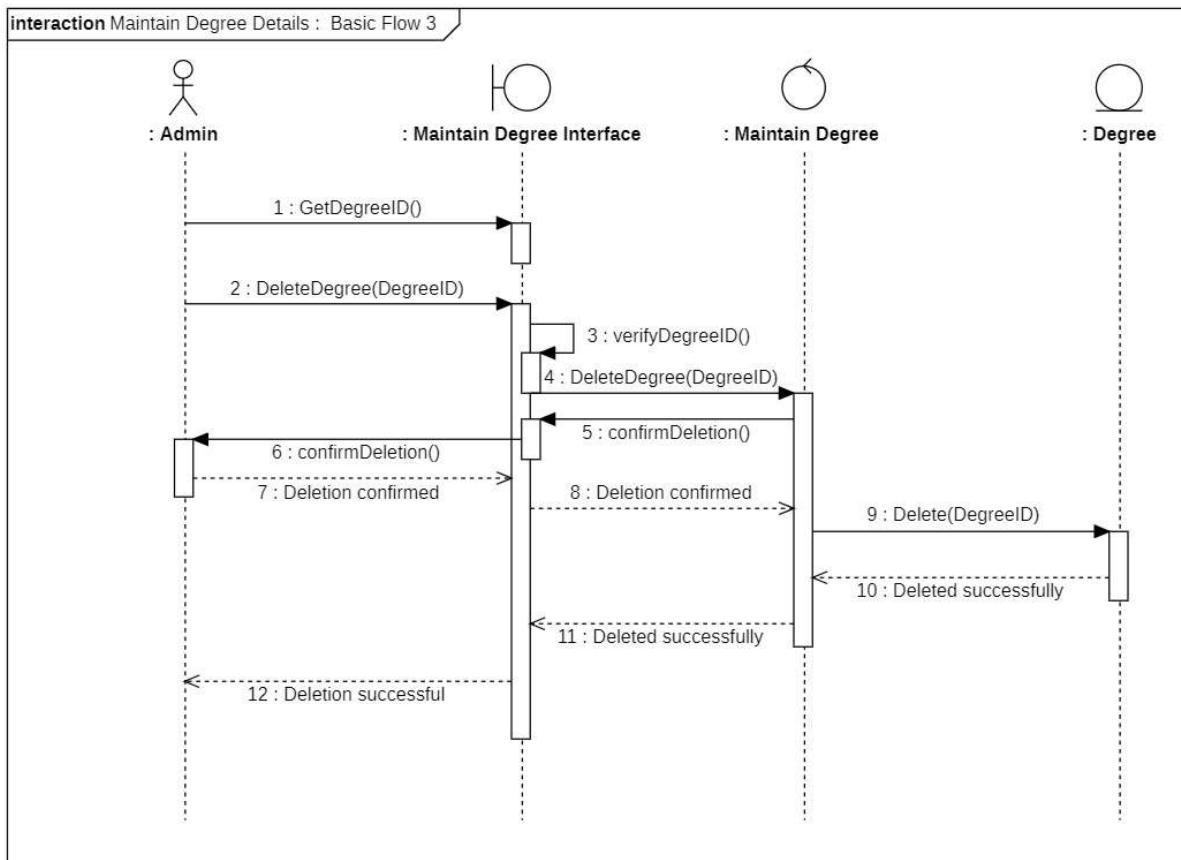
MAINTAIN DEGREE DETAILS BASIC FLOW 1: ADD A DEGREE



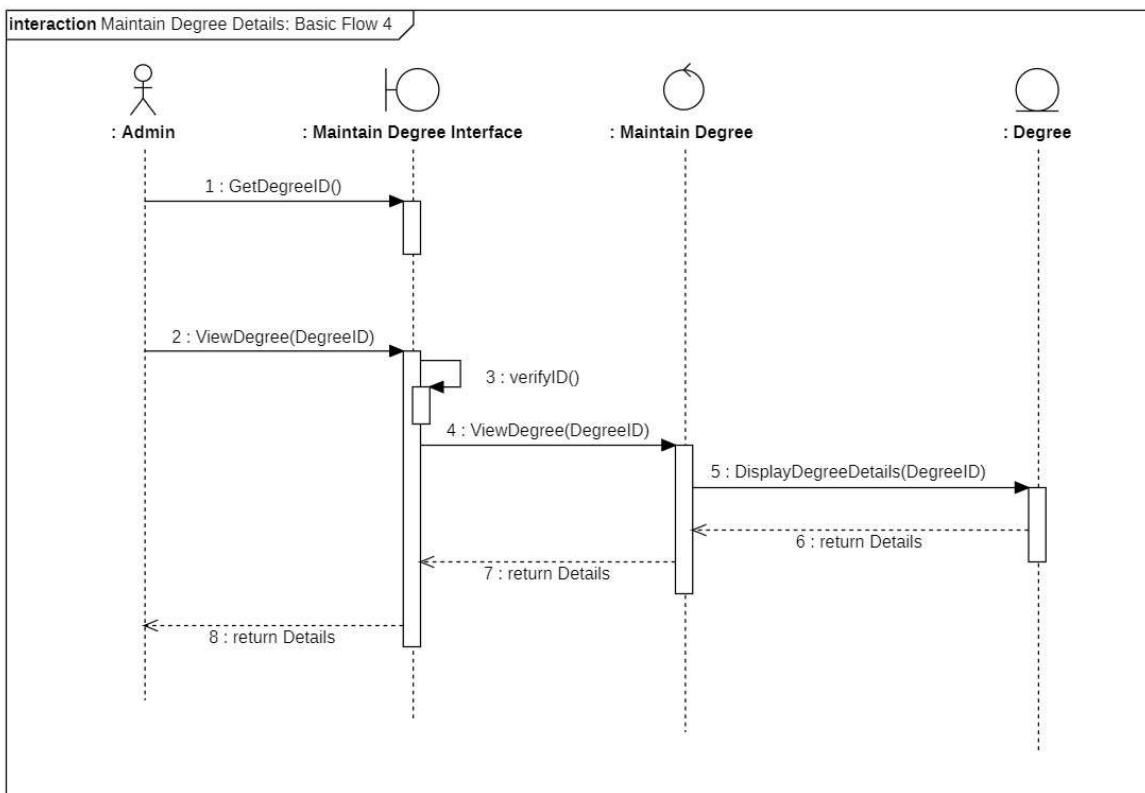
MAINTAIN DEGREE DETAILS BASIC FLOW 2: UPDATE A DEGREE



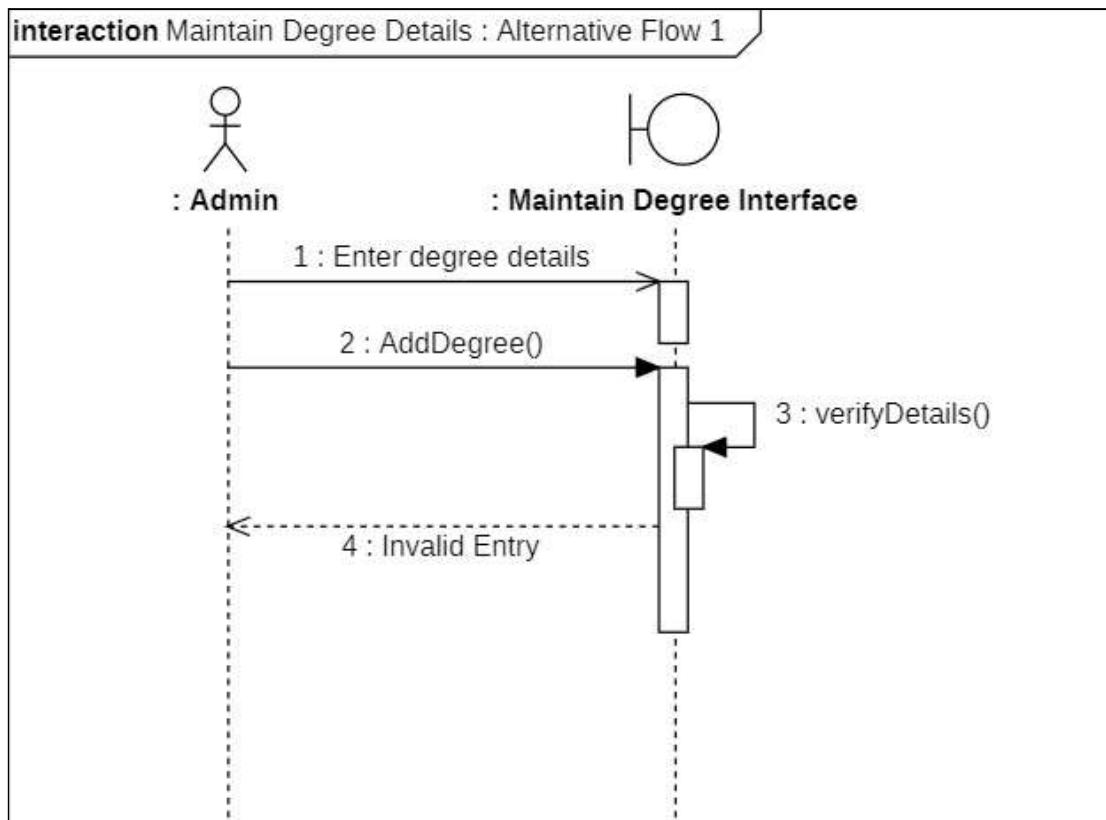
MAINTAIN DEGREE DETAILS BASIC FLOW 3: DELETE A DEGREE



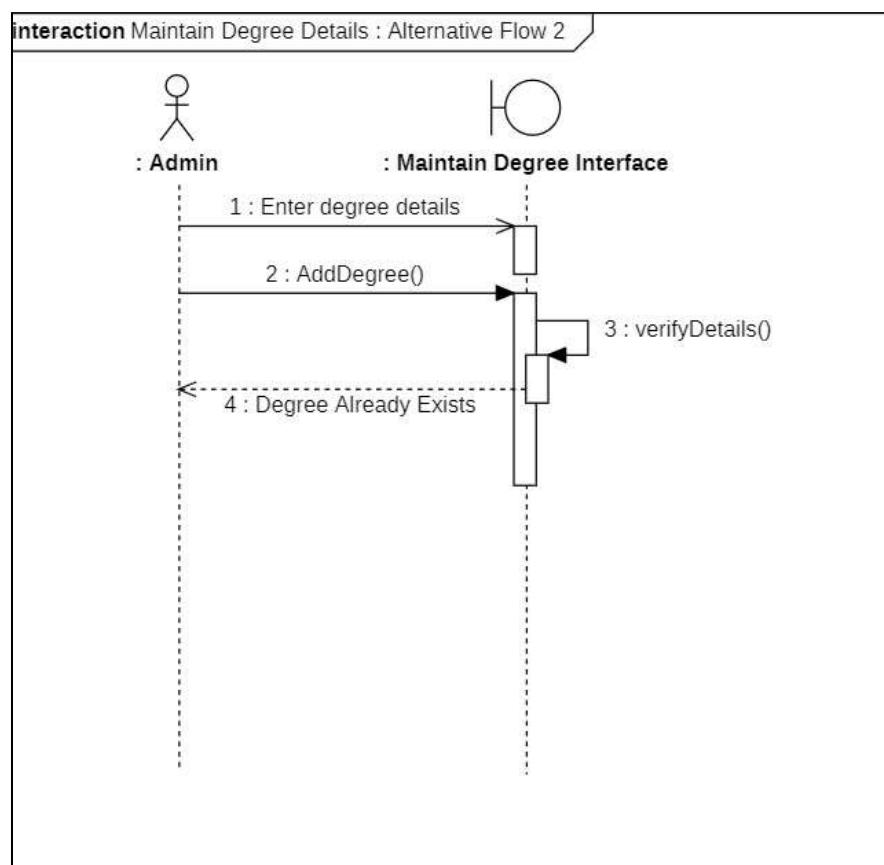
MAINTAIN DEGREE DETAILS BASIC FLOW 4: VIEW A DEGREE



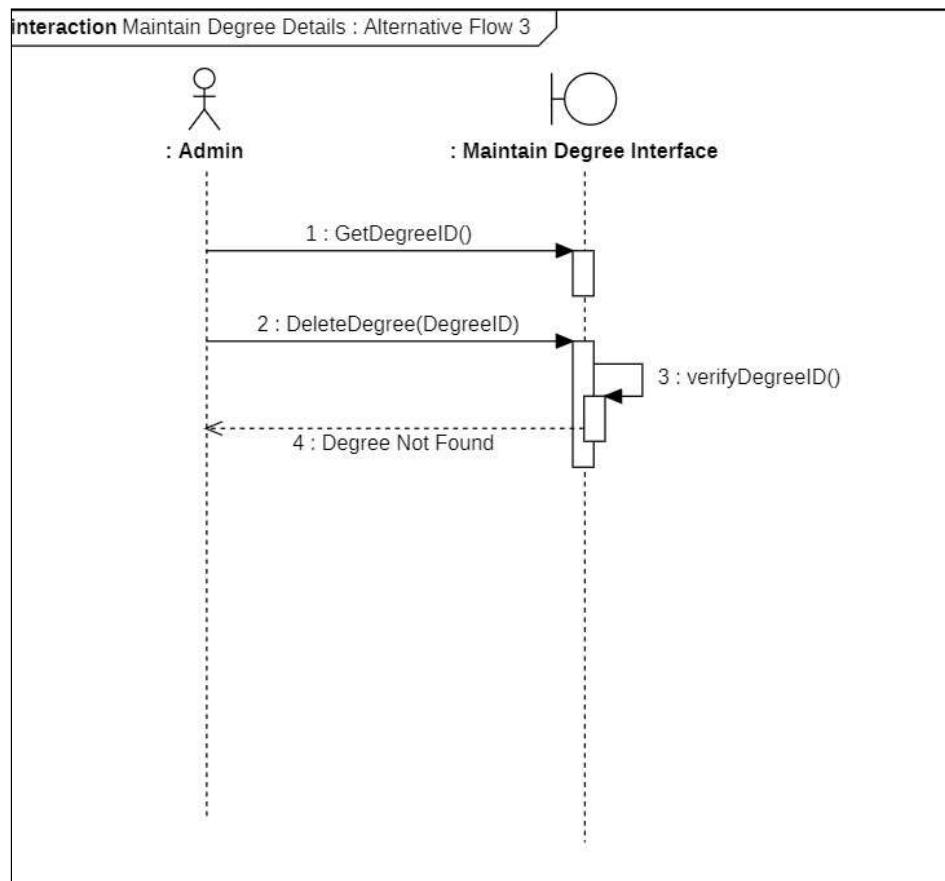
MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 1: INVALID DEGREE ENTRY



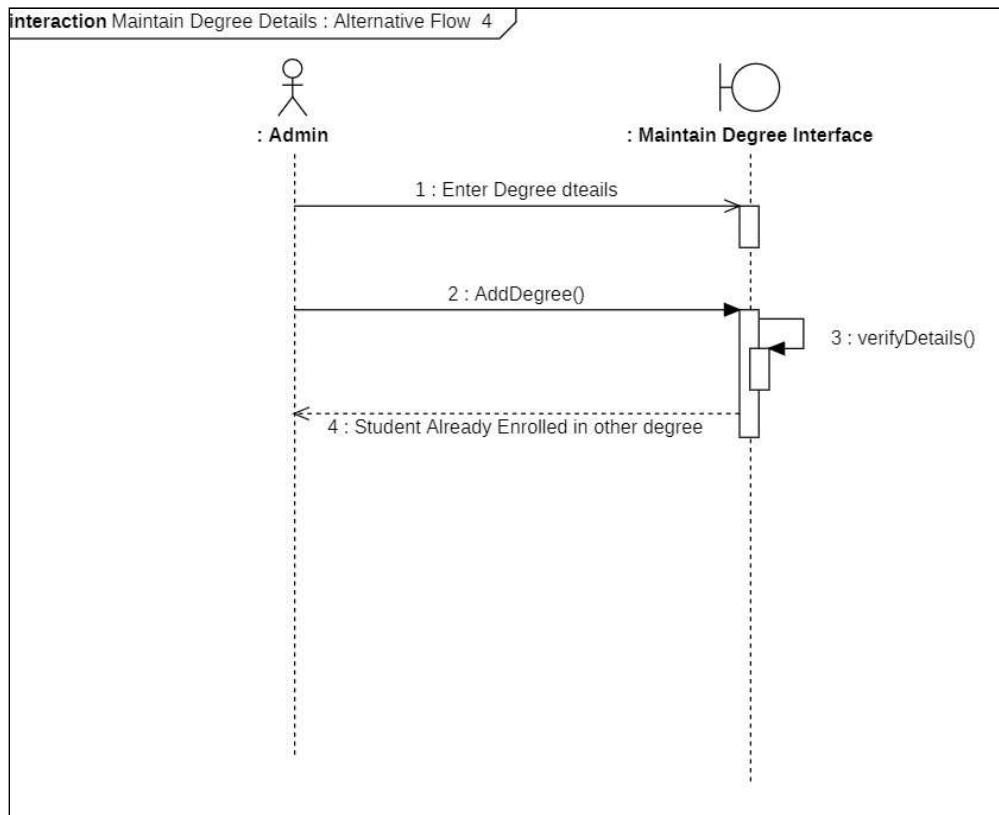
MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 2: DEGREE ALREADY EXIST



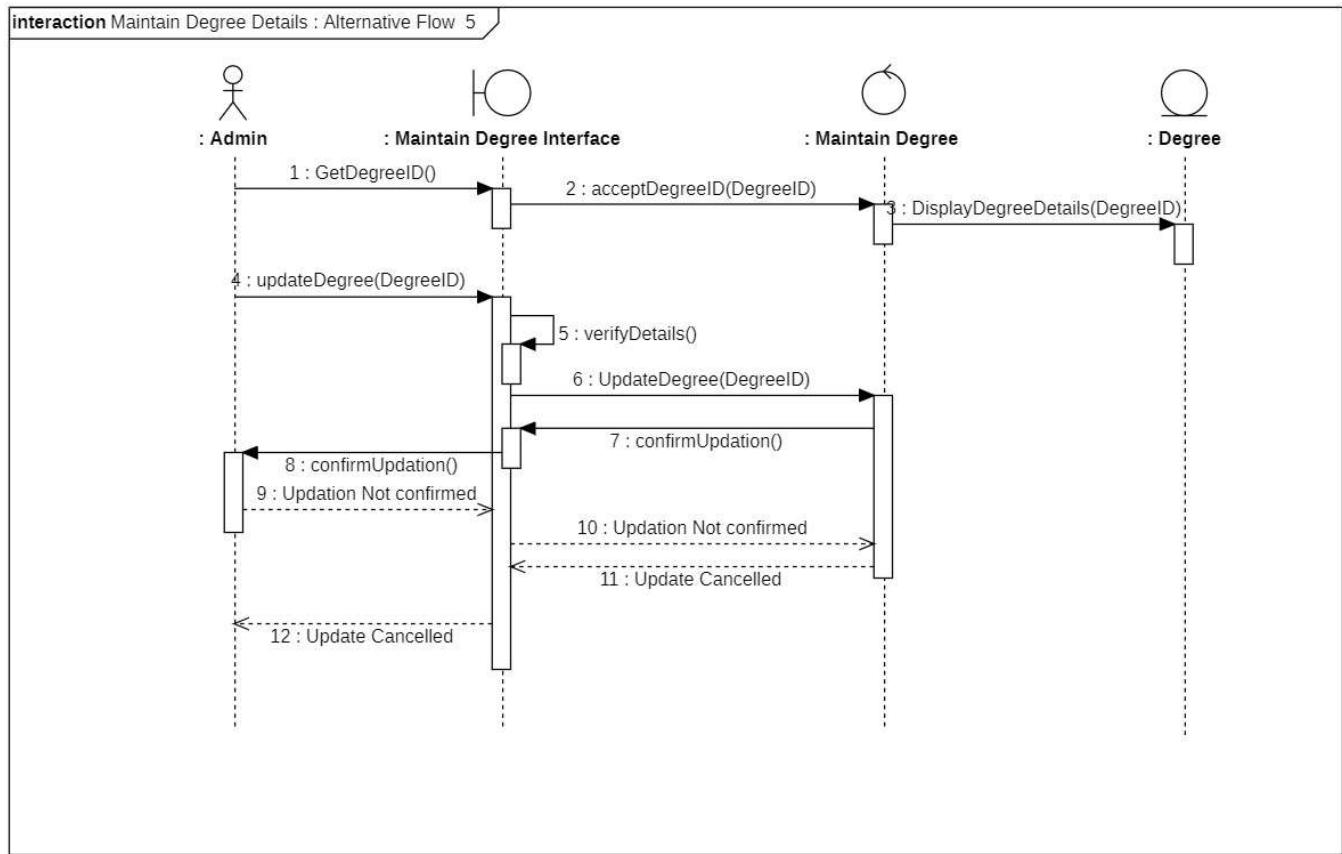
MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 3: DEGREE NOT FOUND



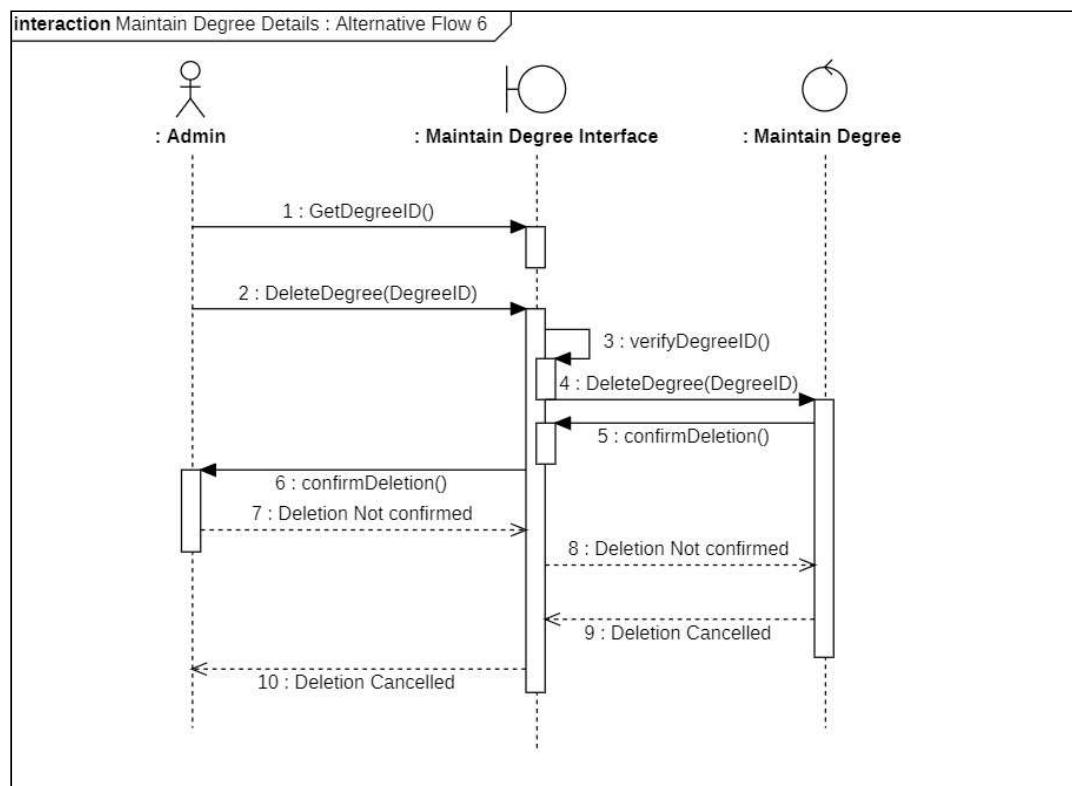
MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 4: STUDENT ALREADY ENROLLED IN OTHER DEGREE



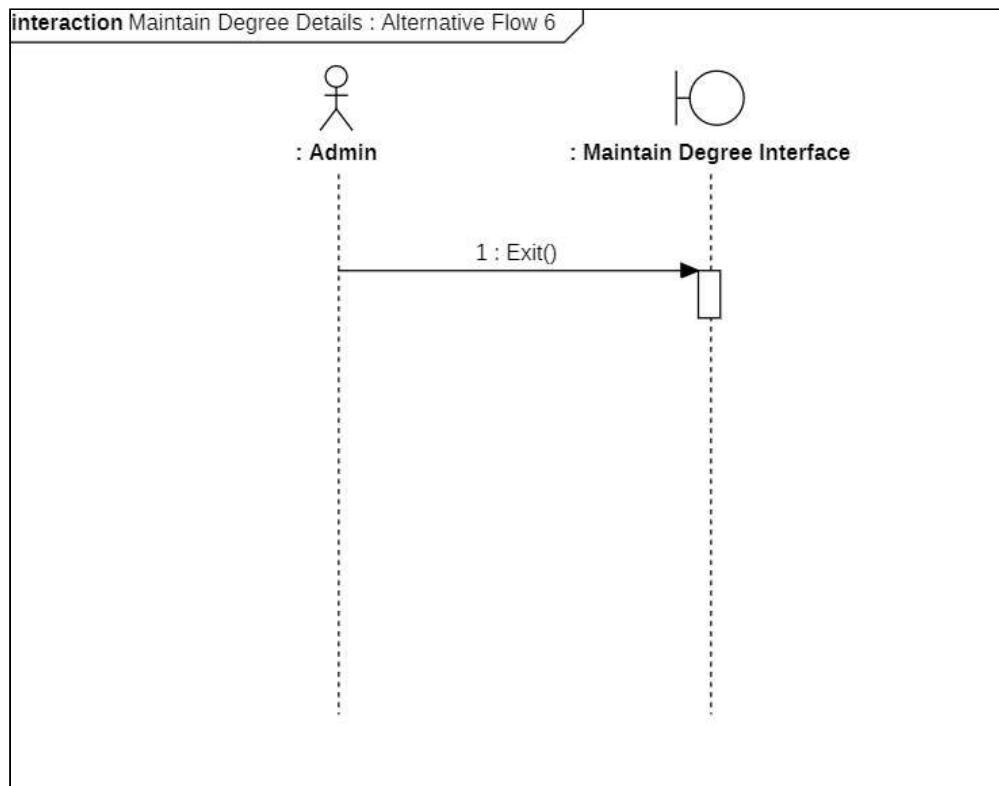
MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 5: UPDATE CANCELLED



MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 6: DELETE CANCELLED

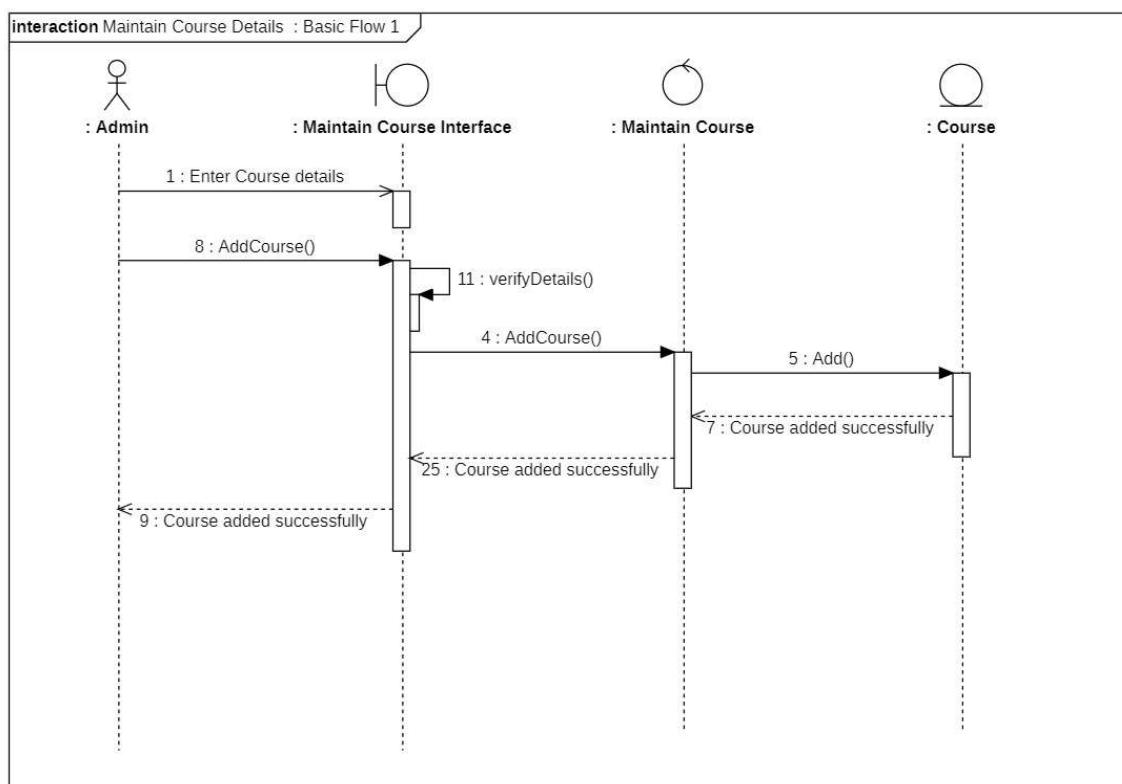


MAINTAIN DEGREE DETAILS ALTERNATIVE FLOW 7: USER EXITS

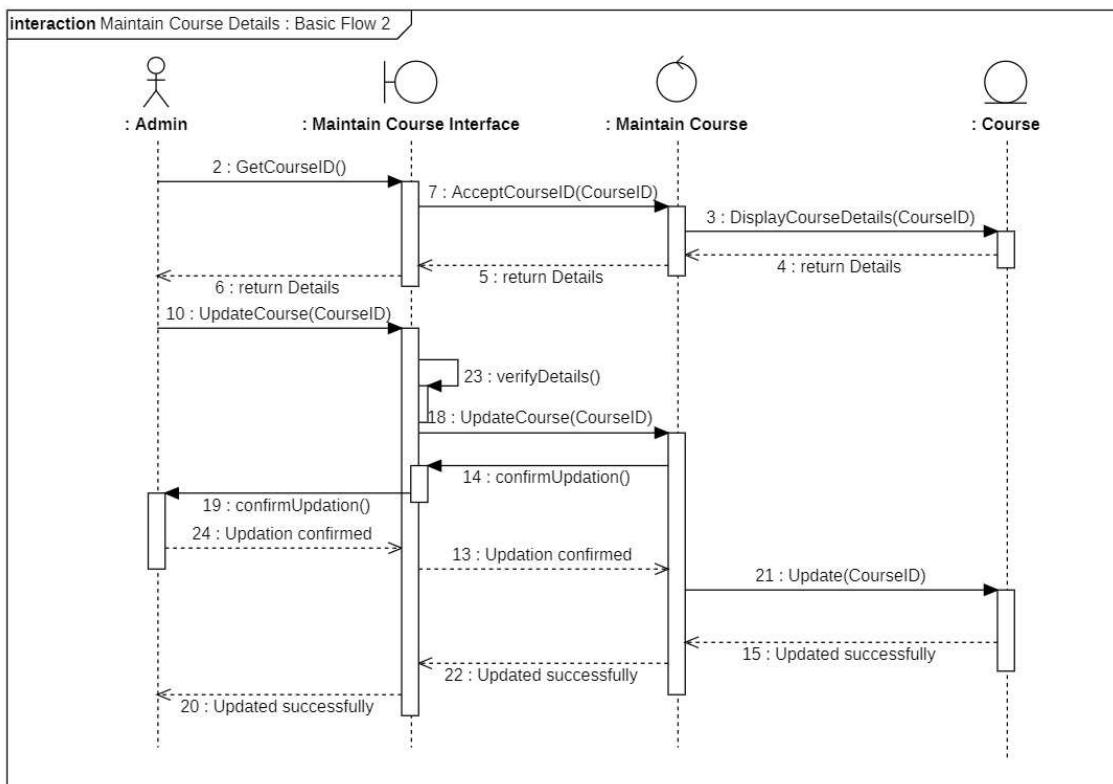


5. MAINTAIN COURSE DETAILS

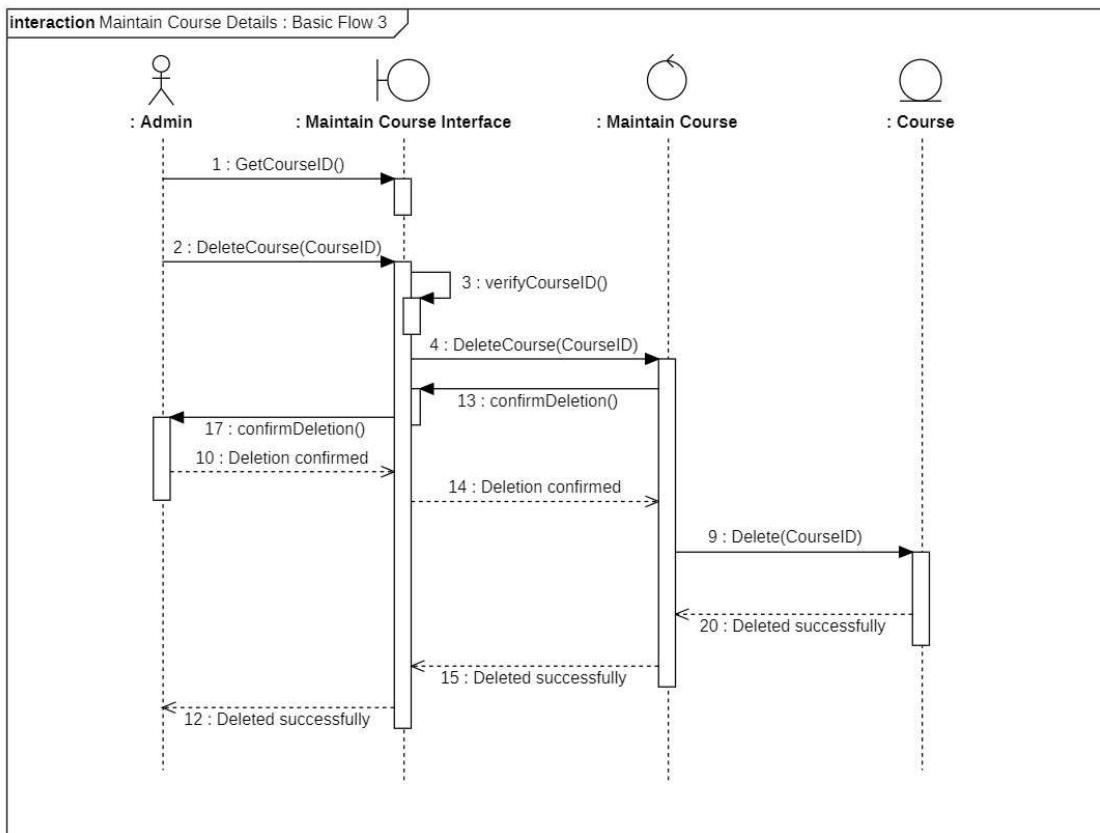
MAINTAIN COURSE DETAILS BASIC FLOW 1: ADD A COURSE



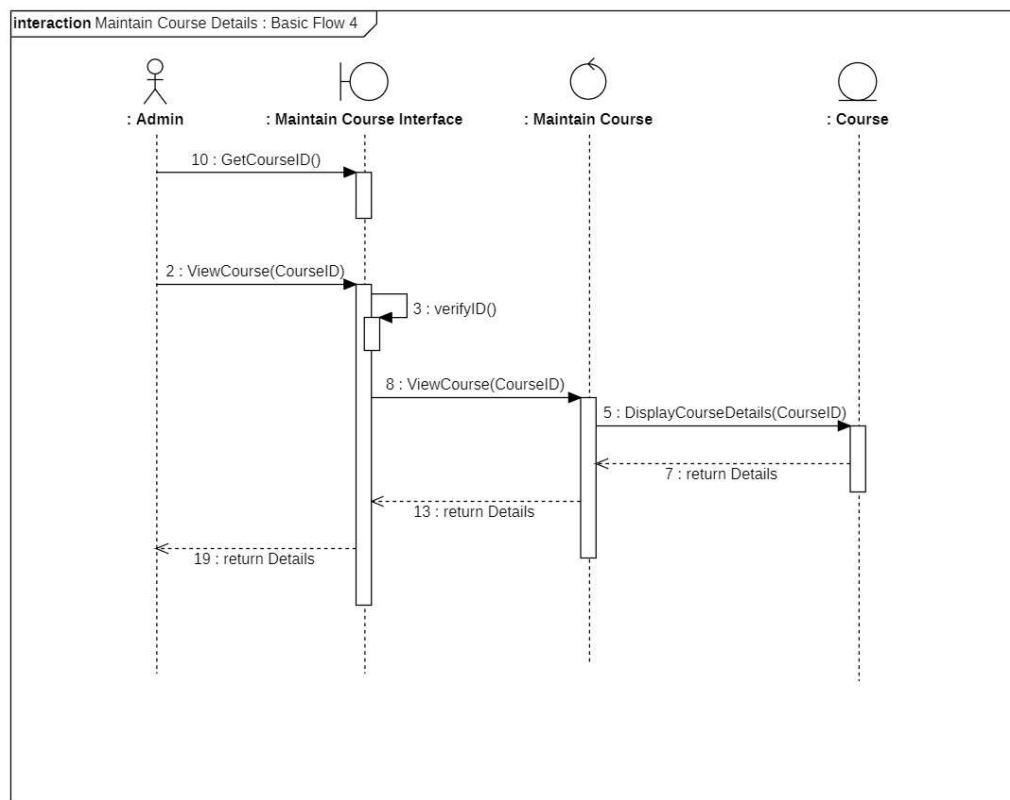
MAINTAIN COURSE DETAILS BASIC FLOW 2 : UPDATE A COURSE



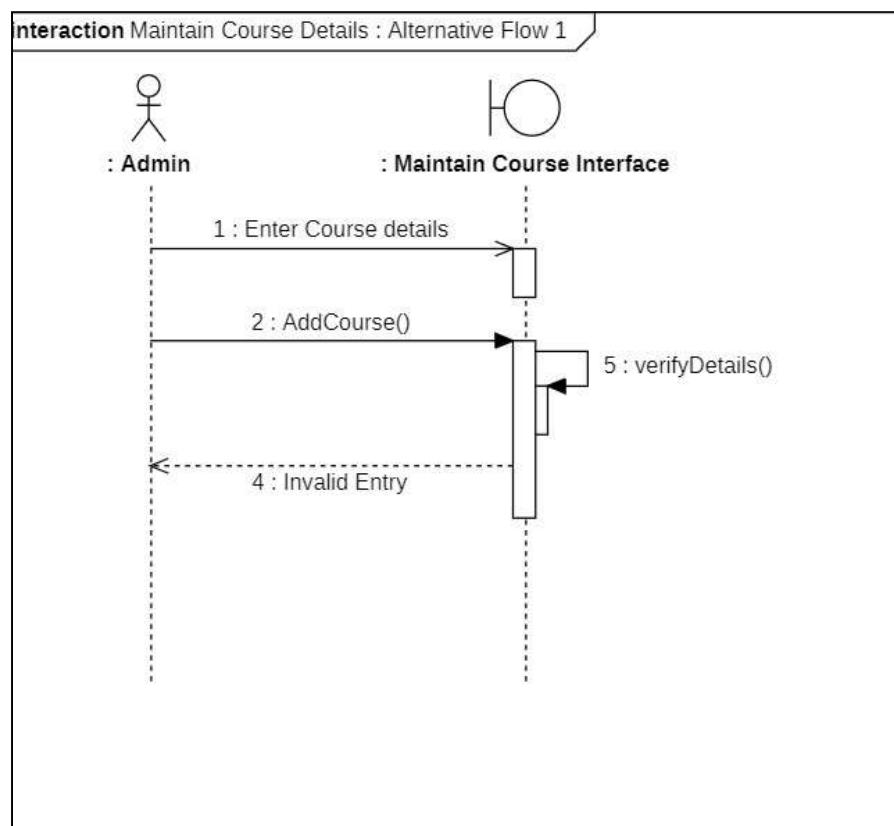
MAINTAIN COURSE DETAILS BASIC FLOW 3 : DELETE A COURSE



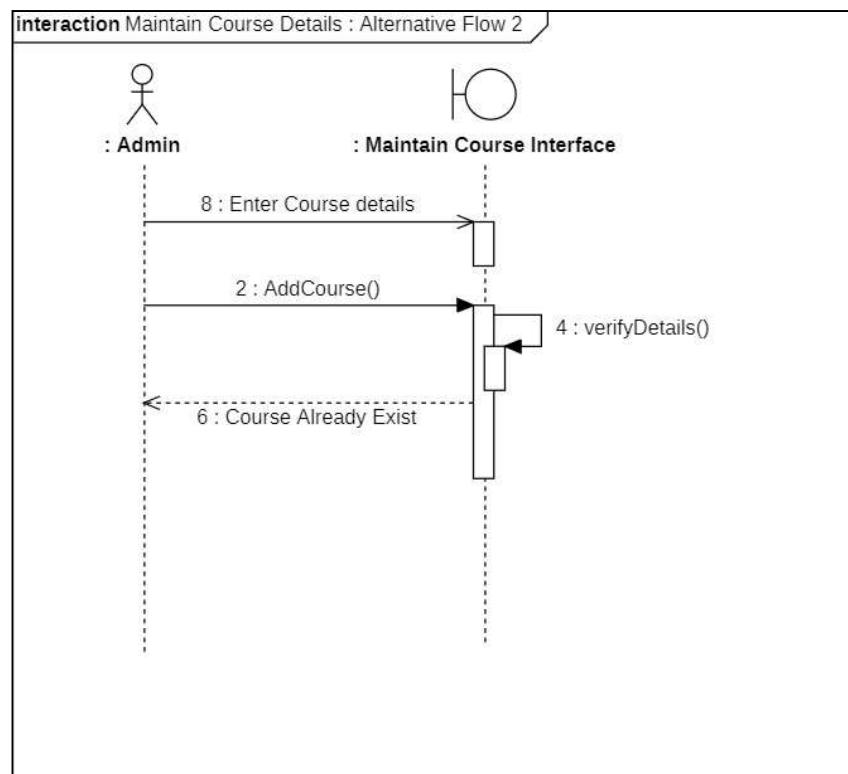
MAINTAIN COURSE DETAILS BASIC FLOW 4 : VIEW A COURSE



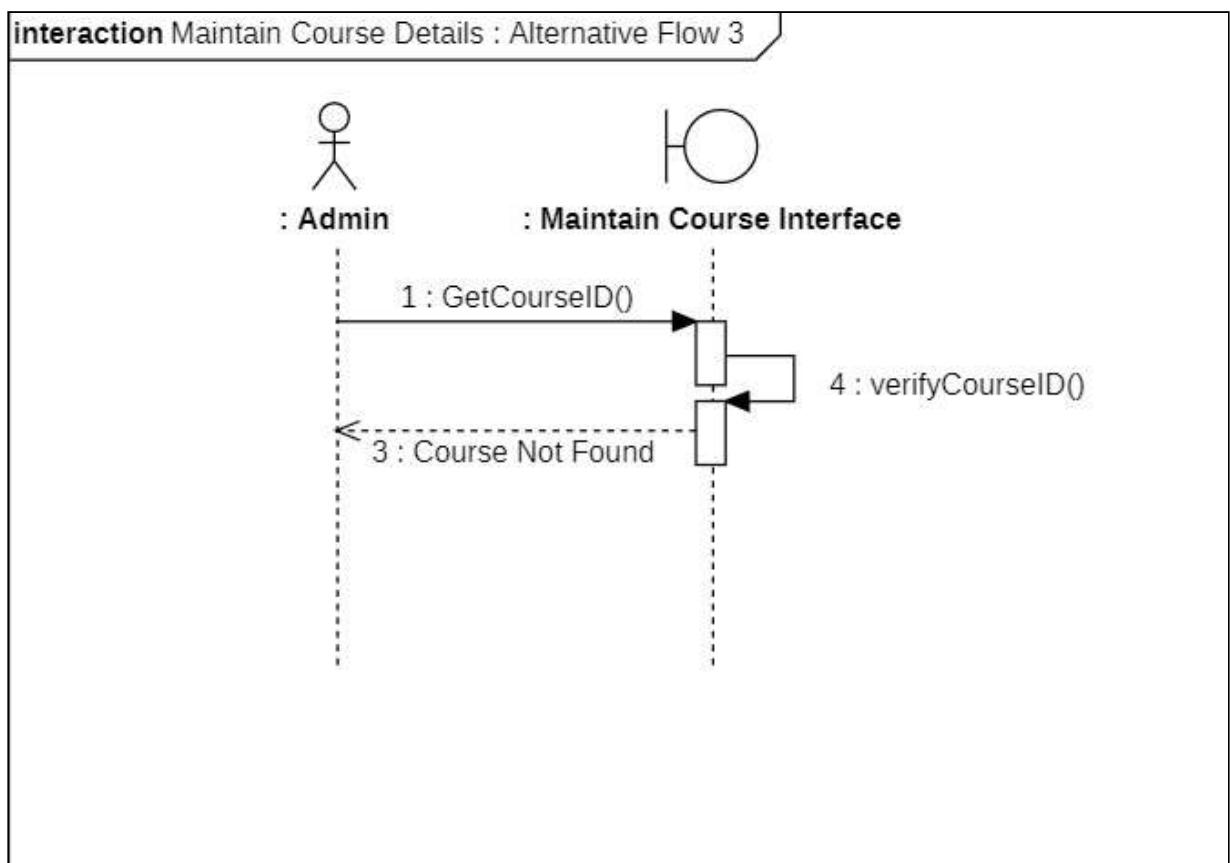
MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 1: INVALID COURSE ENTRY:



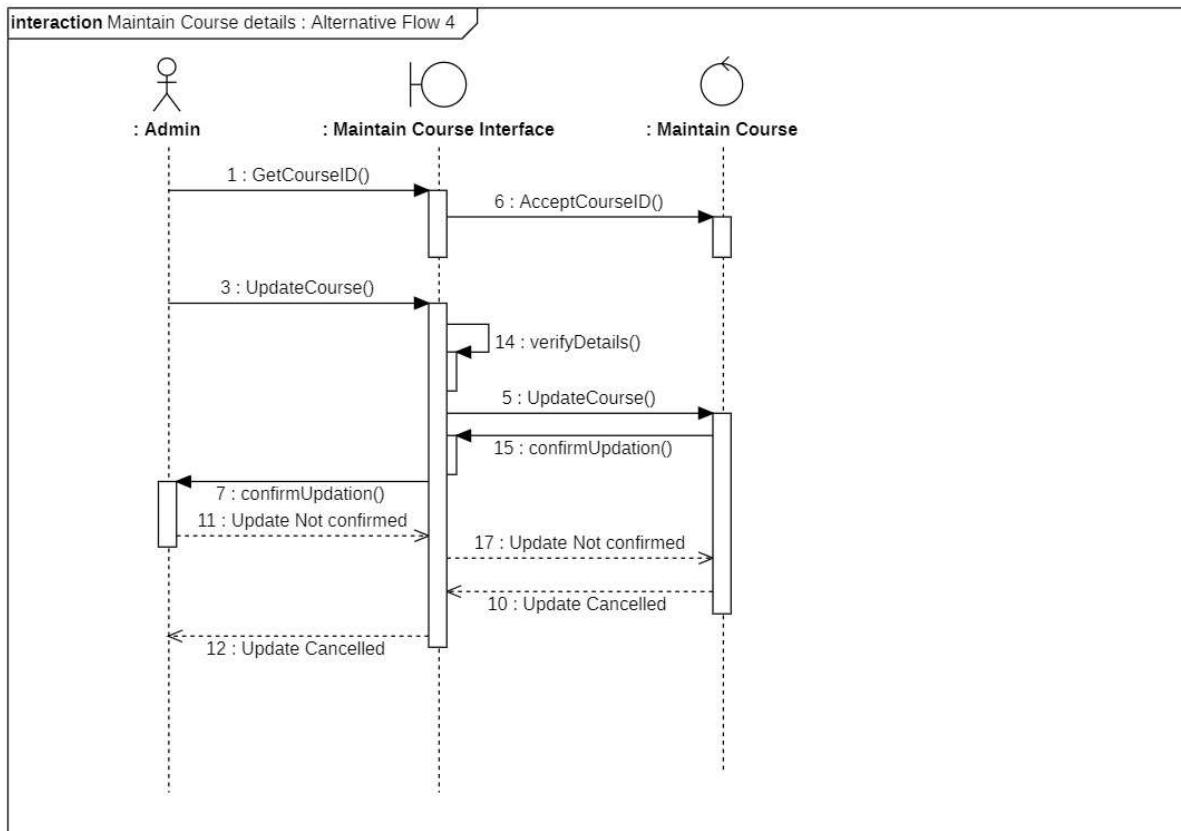
MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 2: COURSE ALREADY EXIST:



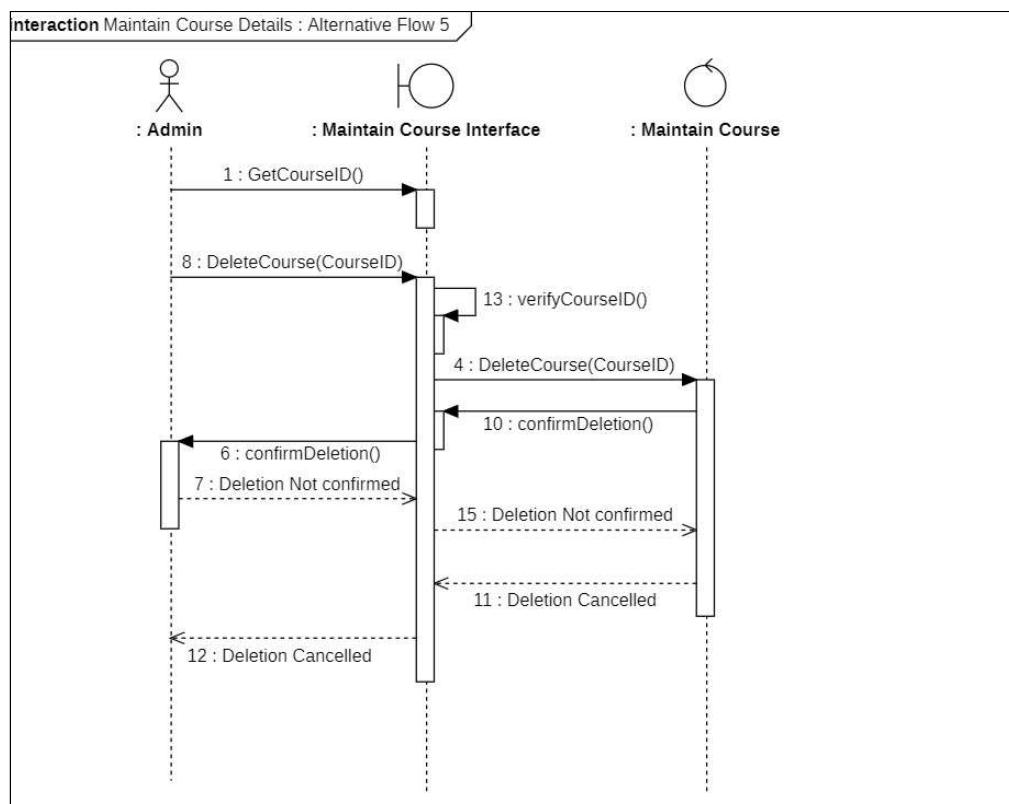
MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 3: COURSE NOT FOUND:



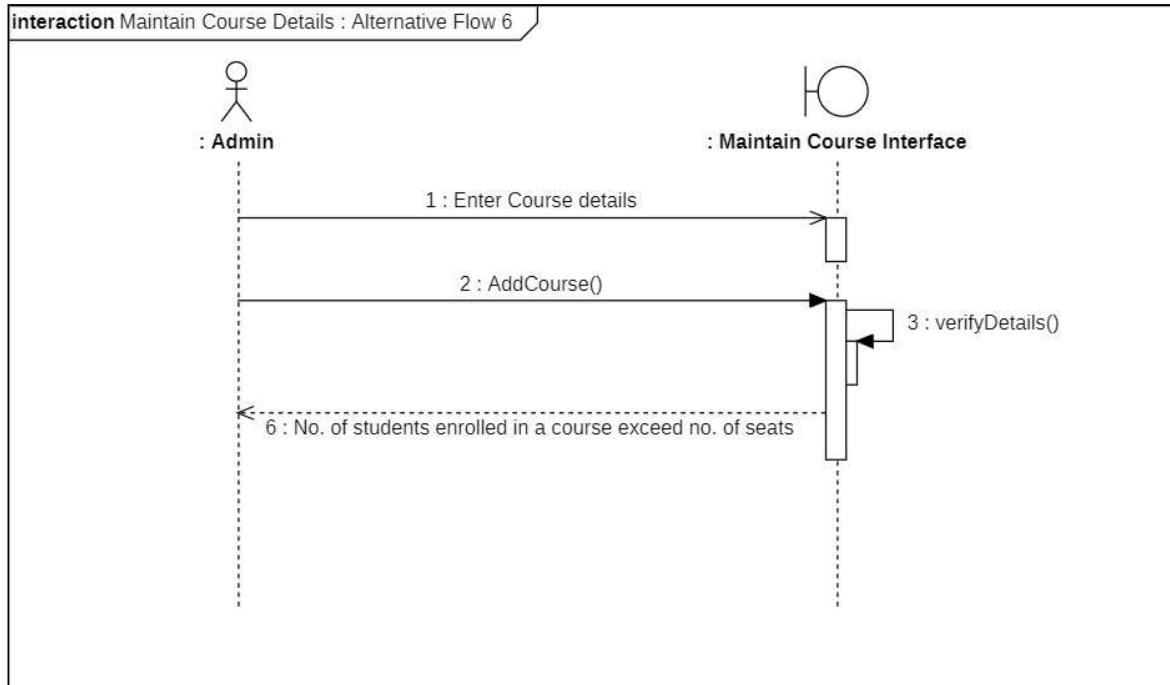
MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 4: UPDATE CANCELLED:



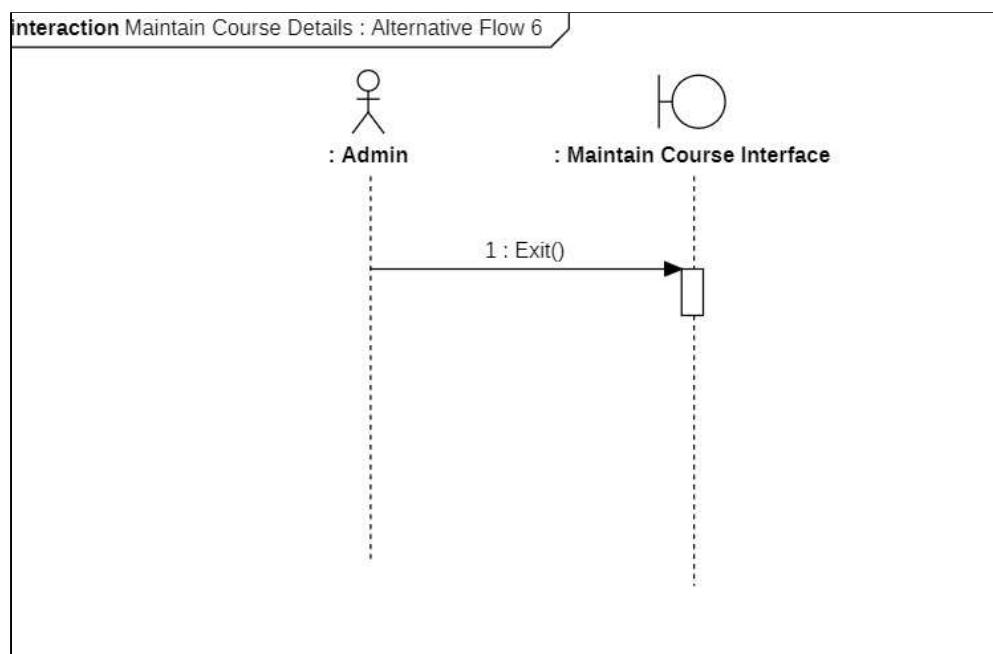
MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 5: DELETE CANCELLED:



MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 6: NO. OF STUDENTS ENROLLED IN A COURSE CANNOT EXCEED NO. OF SEATS:

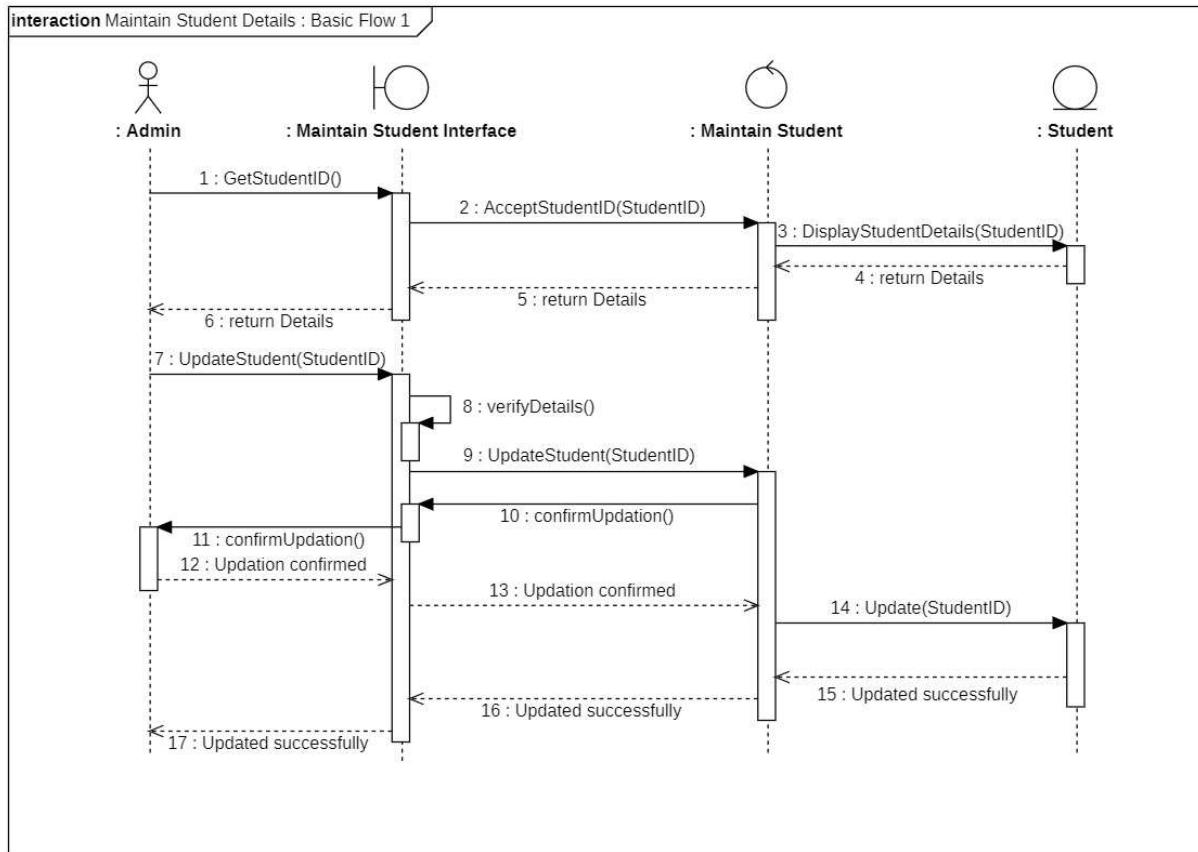


MAINTAIN COURSE DETAILS ALTERNATIVE FLOW 7: USER EXITS:

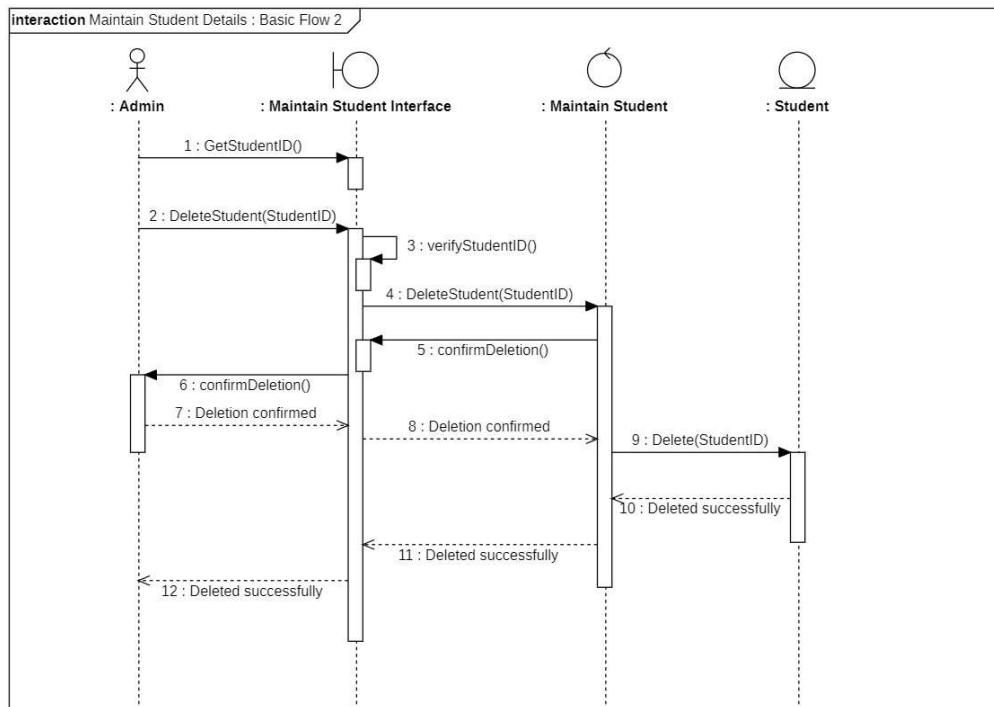


6. MAINTAIN STUDENT DETAILS

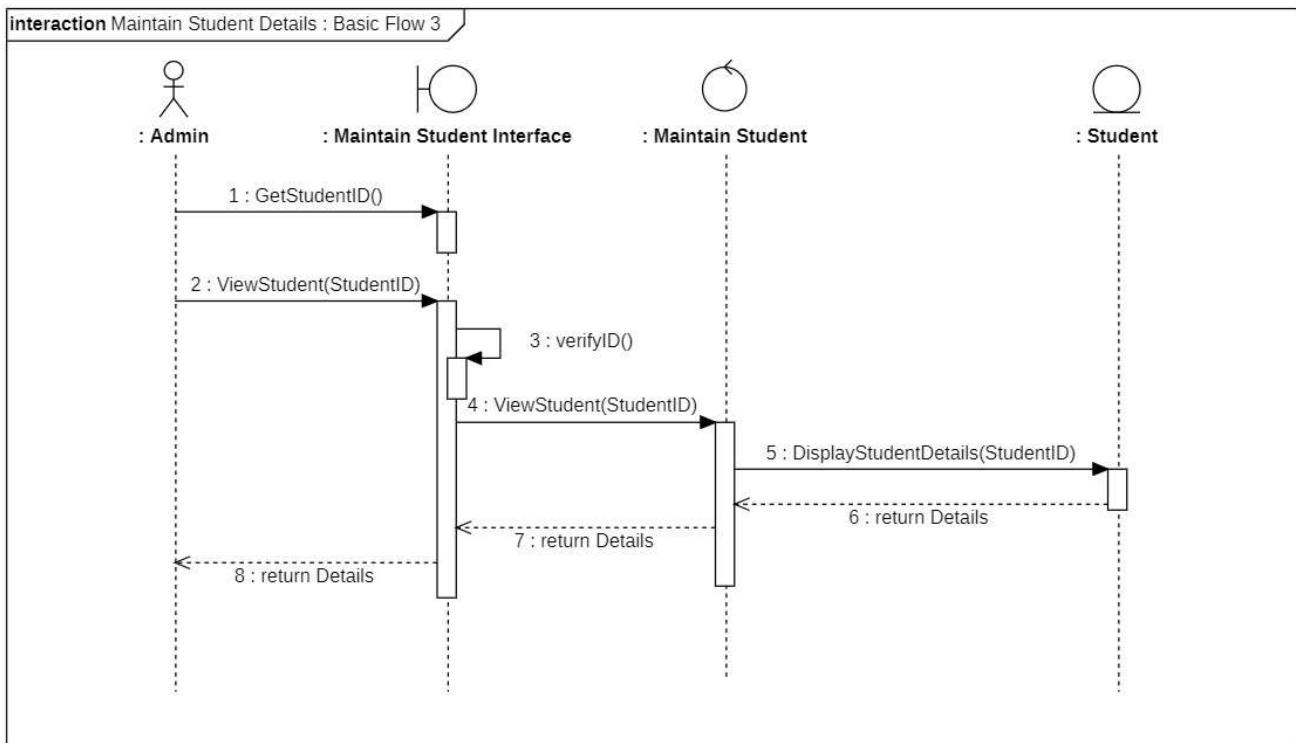
MAINTAIN STUDENT DETAILS BASIC FLOW 1 : UPDATE A STUDENT



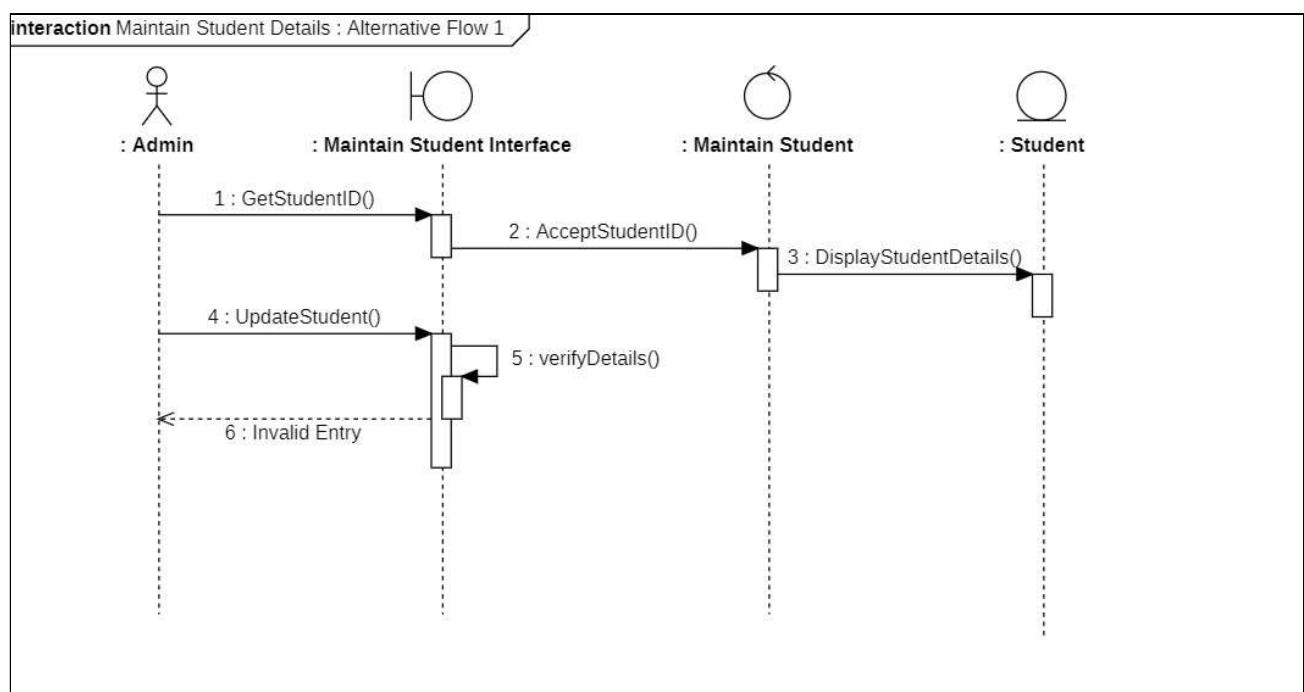
MAINTAIN STUDENT DETAILS BASIC FLOW 2 : DELETE A STUDENT



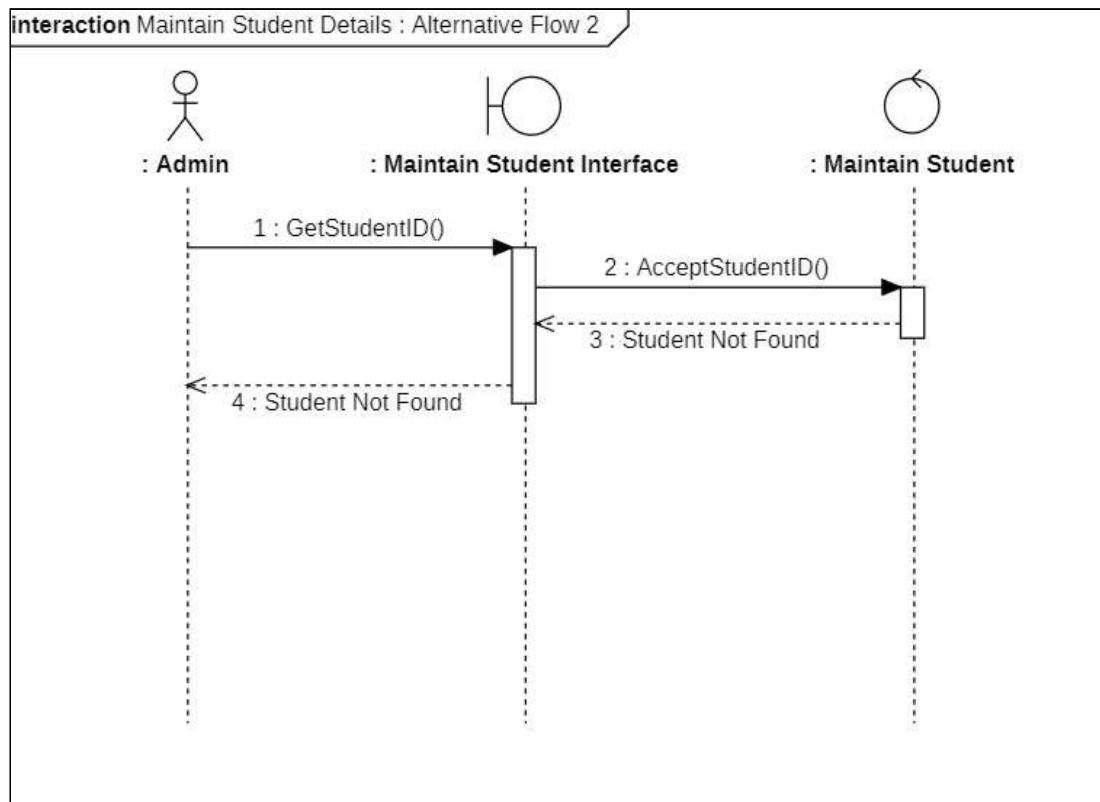
MAINTAIN STUDENT DETAILS BASIC FLOW 3 : VIEW A STUDENT



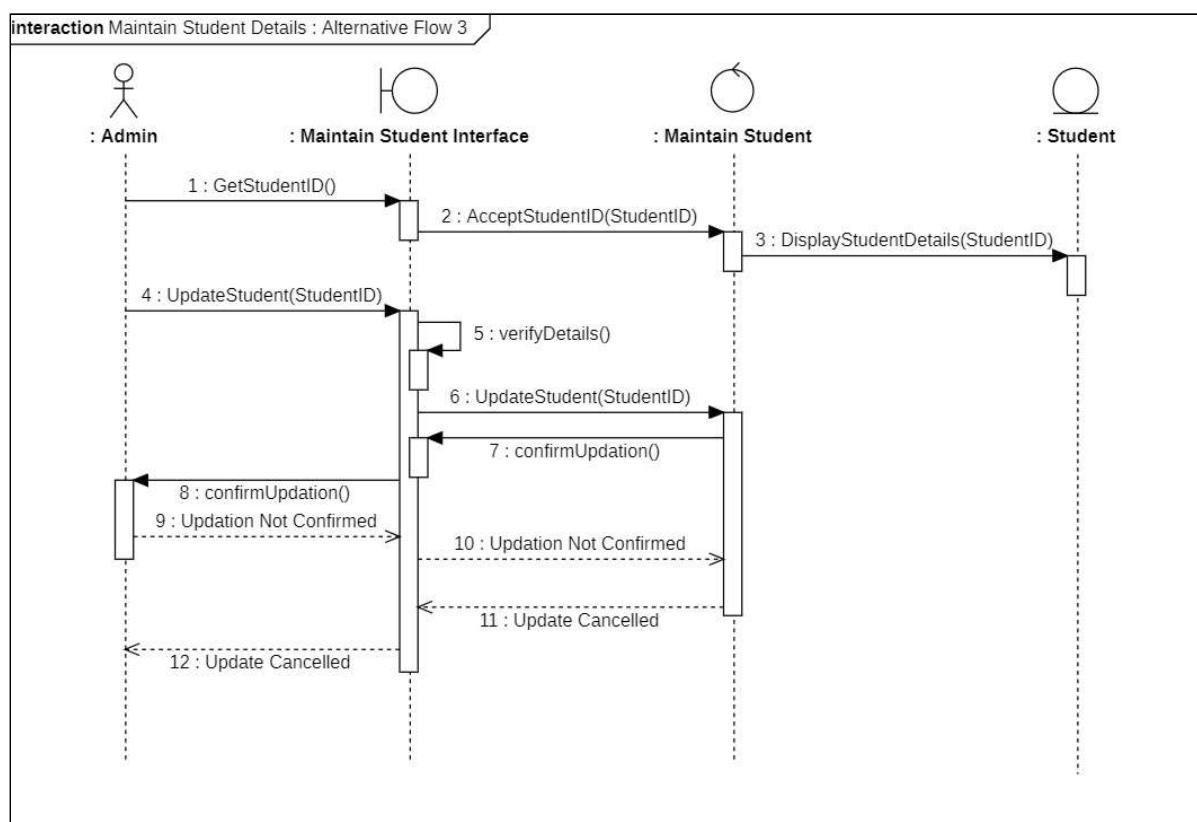
MAINTAIN STUDENT DETAILS ALTERNATIVE FLOW 1: INVALID STUDENT ENTRY



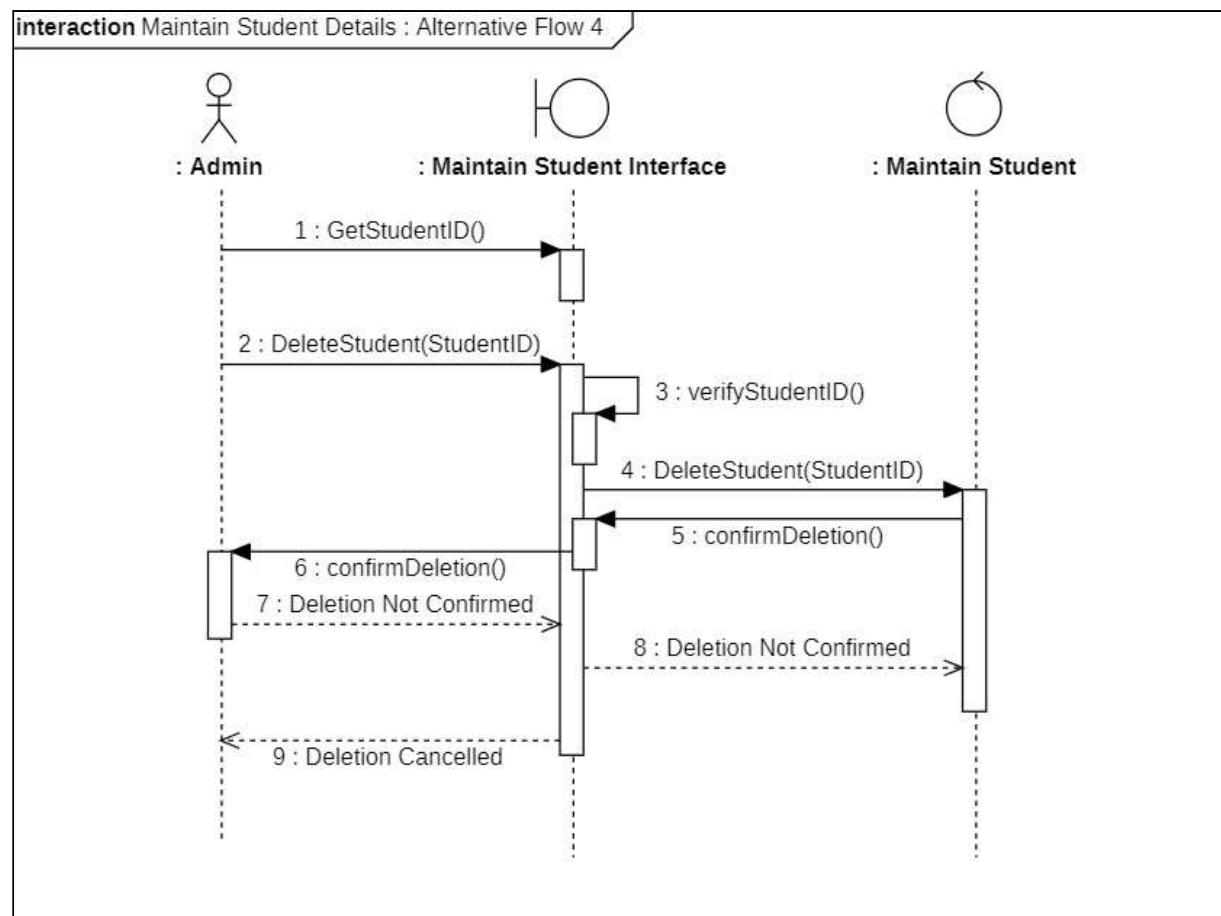
MAINTAIN STUDENT DETAILS ALTERNATIVE FLOW 2: STUDENT NOT FOUND



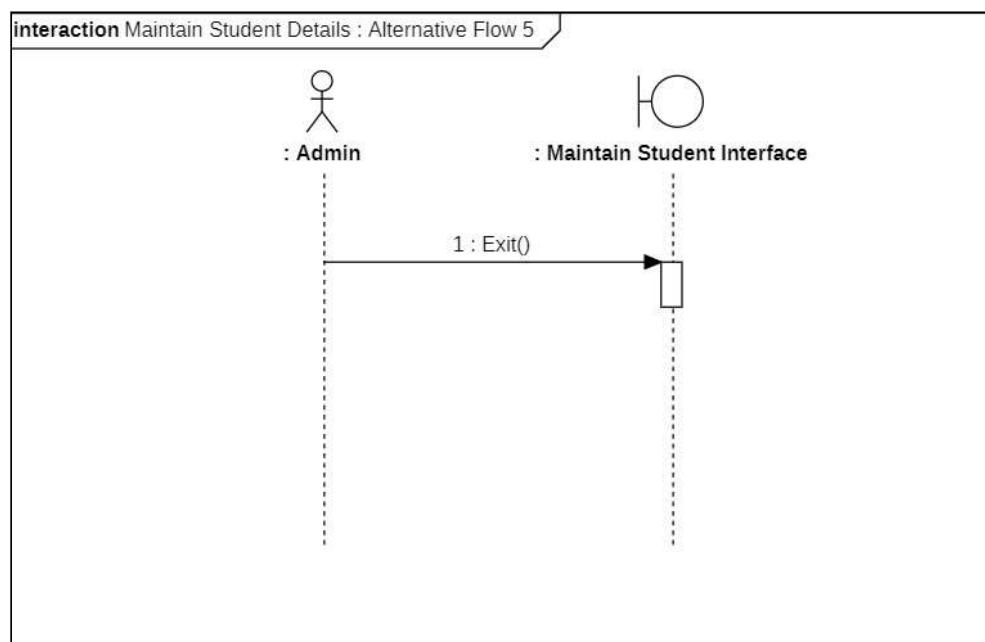
MAINTAIN STUDENT DETAILS ALTERNATIVE FLOW 3:UPDATE CANCELLED



MAINTAIN STUDENT DETAILS ALTERNATIVE FLOW 4:DELETE CANCELLED

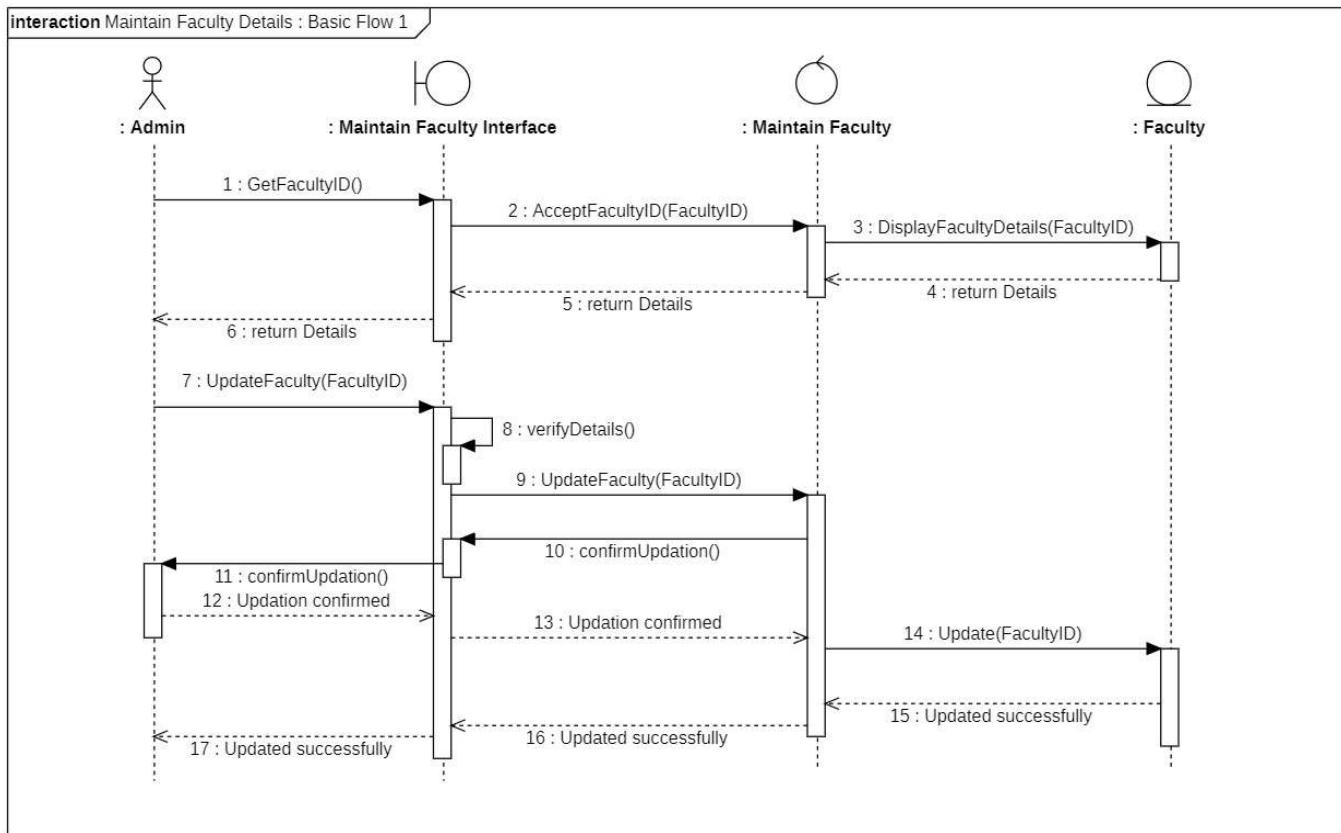


MAINTAIN STUDENT DETAILS ALTERNATIVE FLOW 5: USER EXITS

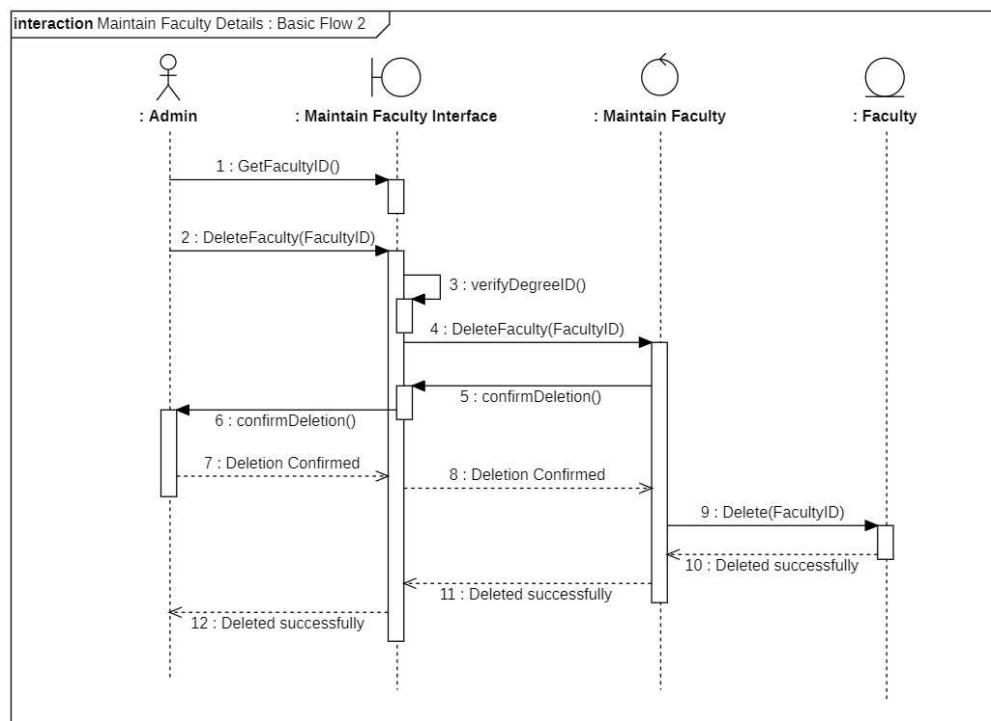


7. MAINTAIN FACULTY DETAILS

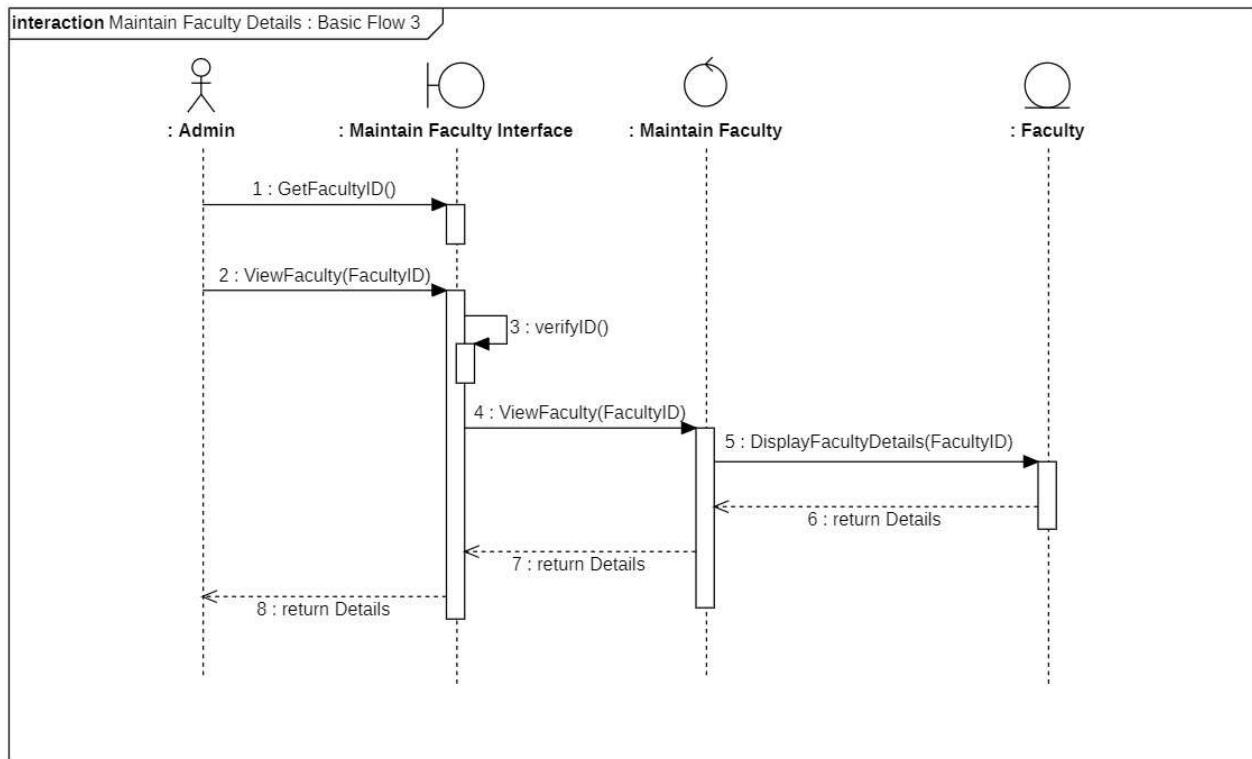
MAINTAIN FACULTY DETAILS BASIC FLOW 1 : UPDATE A FACULTY



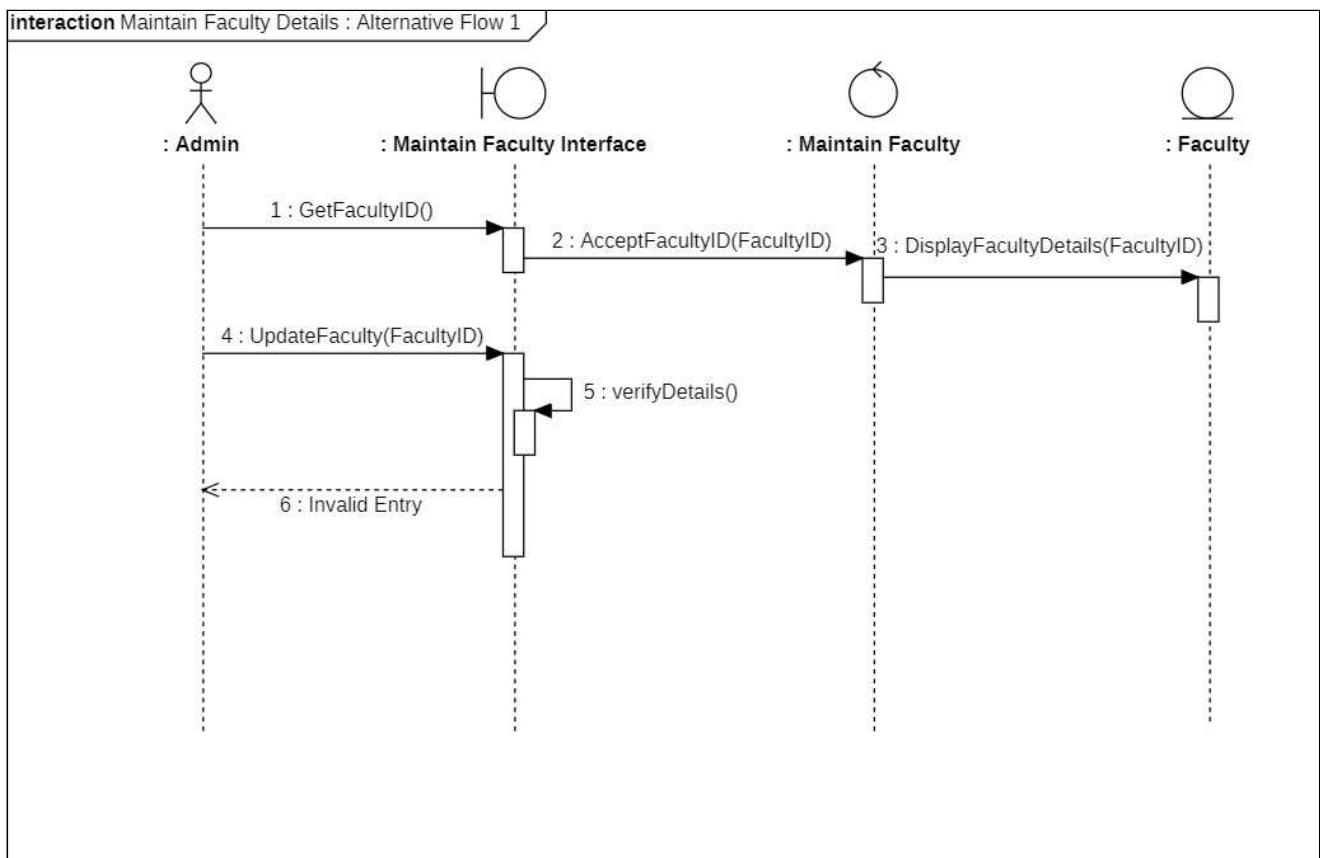
MAINTAIN FACULTY DETAILS BASIC FLOW 2 : DELETE A FACULTY



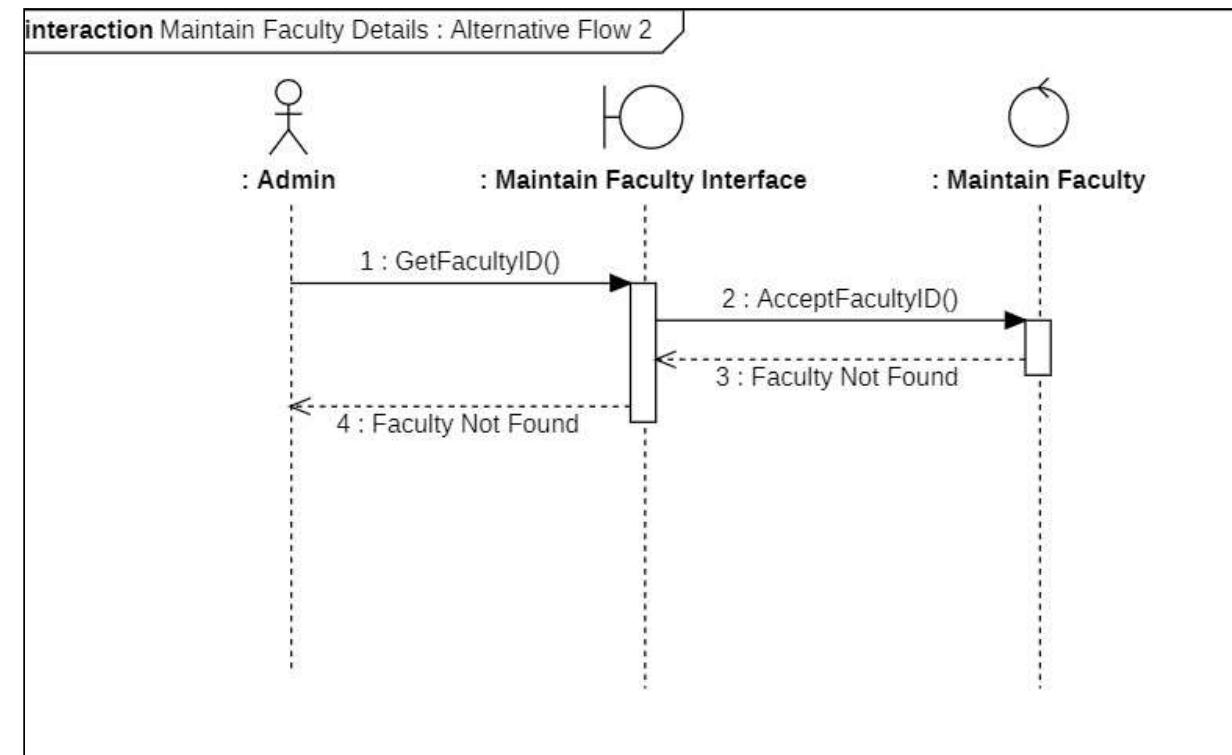
MAINTAIN FACULTY DETAILS BASIC FLOW 3 : VIEW A FACULTY



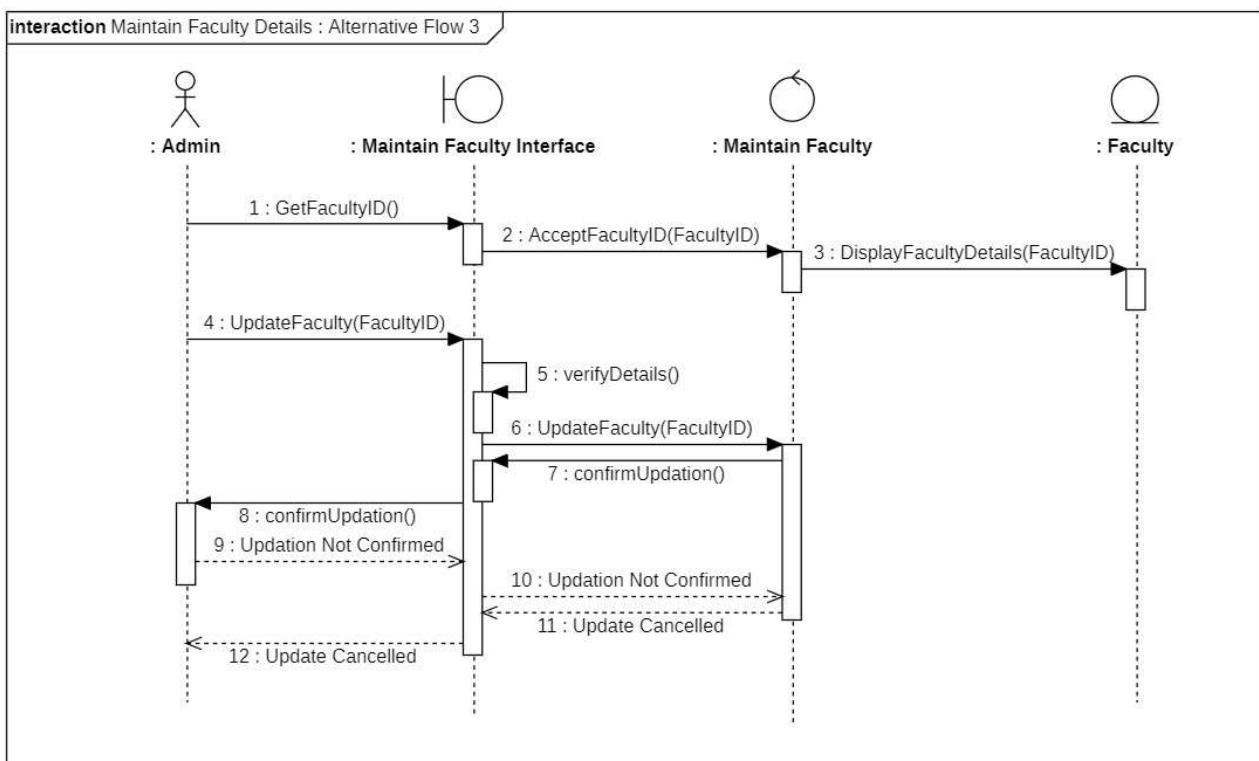
MAINTAIN FACULTY DETAILS ALTERNATIVE FLOW 1: INVALID FACULTY ENTRY



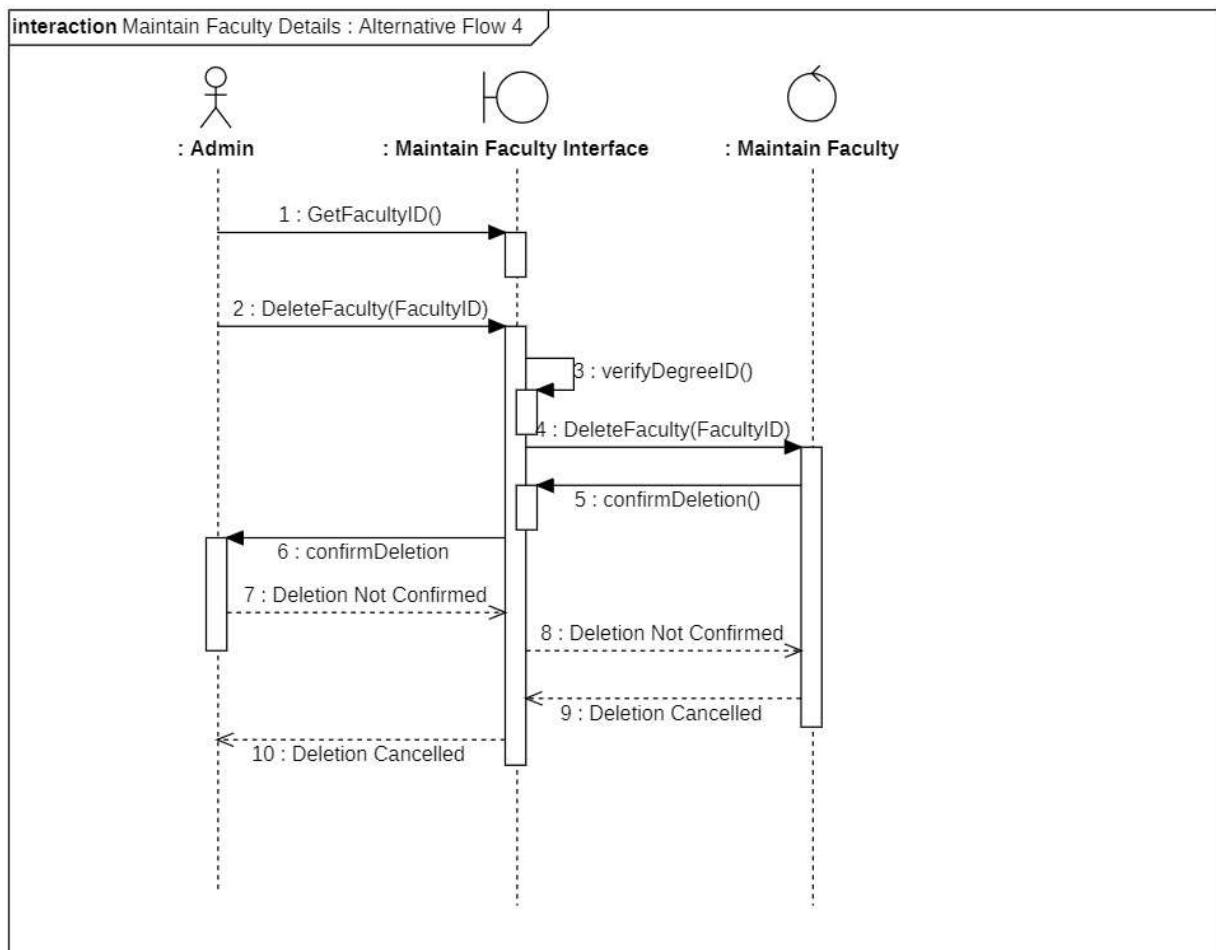
MAINTAIN FACULTY DETAILS ALTERNATIVE FLOW 2: FACULTY NOT FOUND



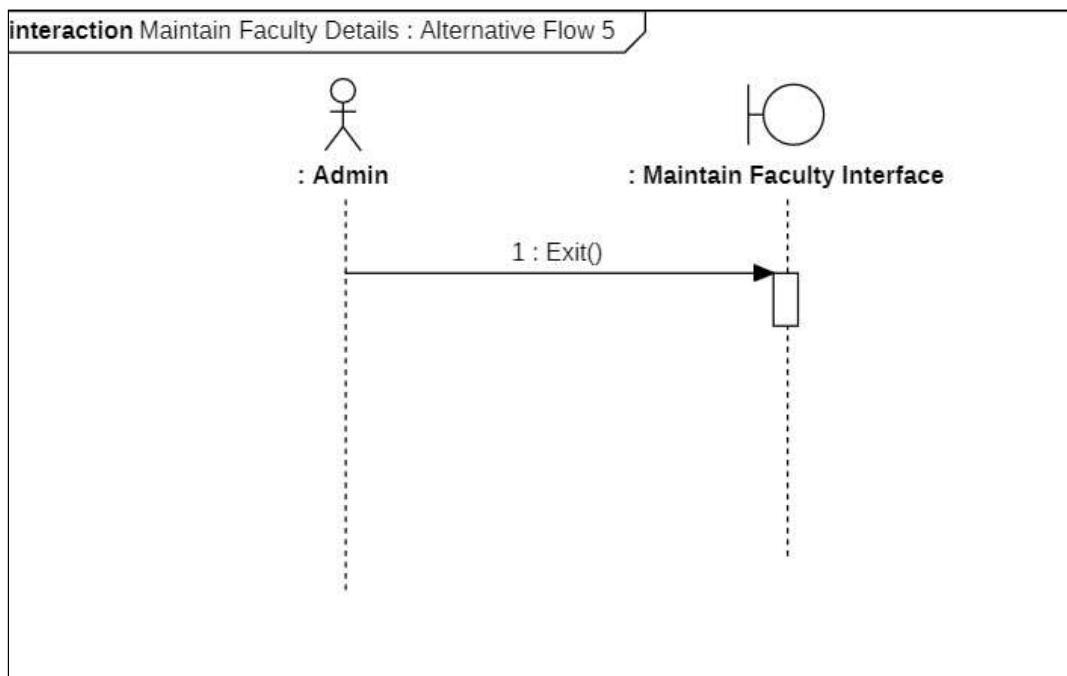
MAINTAIN FACULTY DETAILS ALTERNATIVE FLOW 3:UPDATE CANCELLED



MAINTAIN FACULTY DETAILS ALTERNATIVE FLOW 4:DELETE CANCELLED

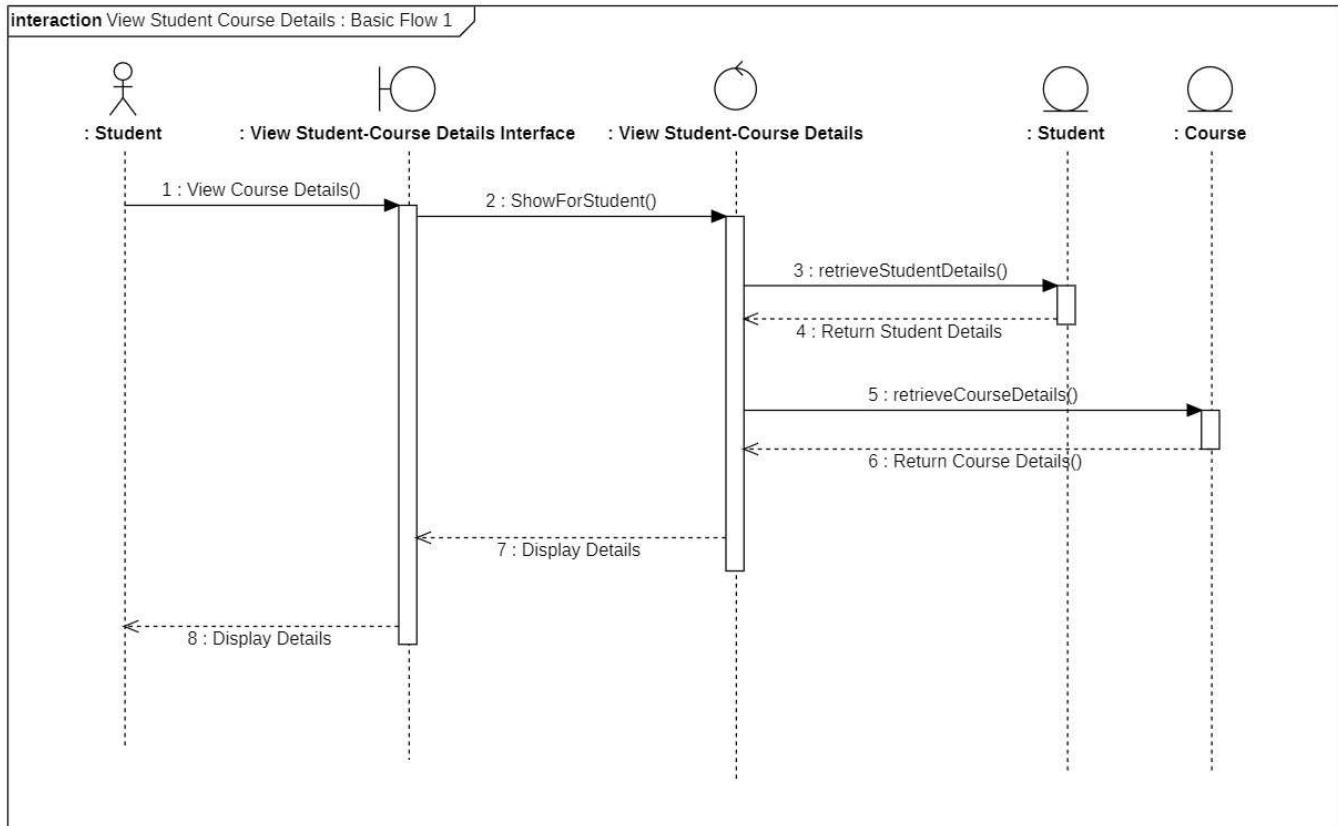


MAINTAIN FACULTY DETAILS ALTERNATIVE FLOW 5: USER EXITS

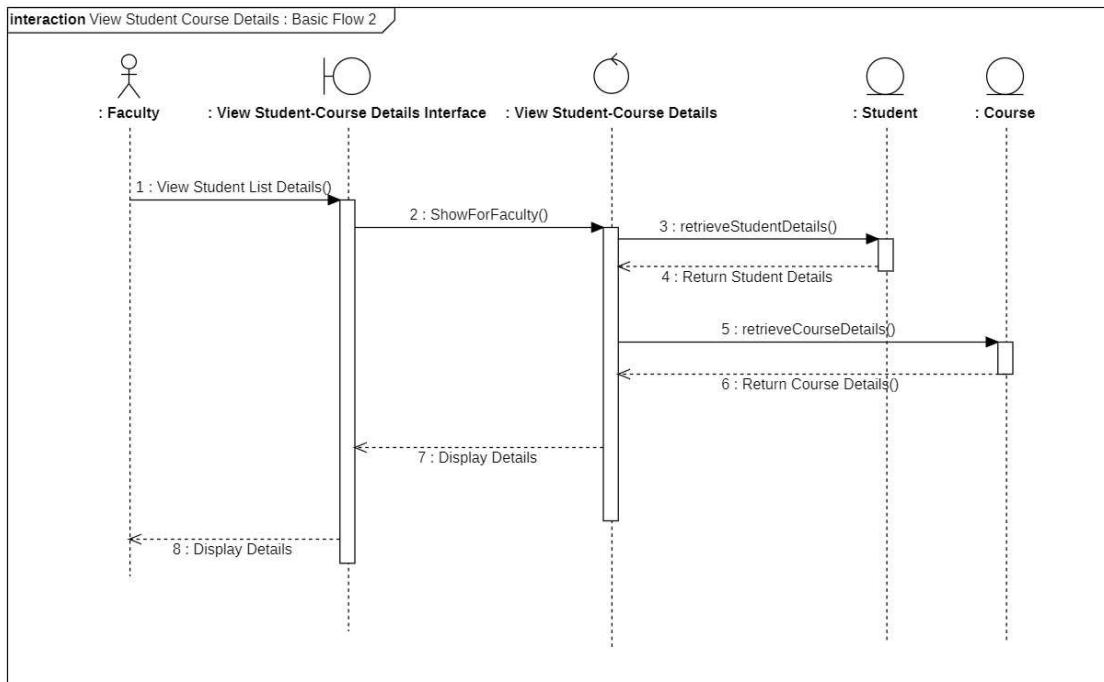


8. VIEW STUDENT-COURSE DETAILS

VIEW STUDENT-COURSE DETAILS BASIC FLOW 1:VIEW DETAILS BY STUDENT

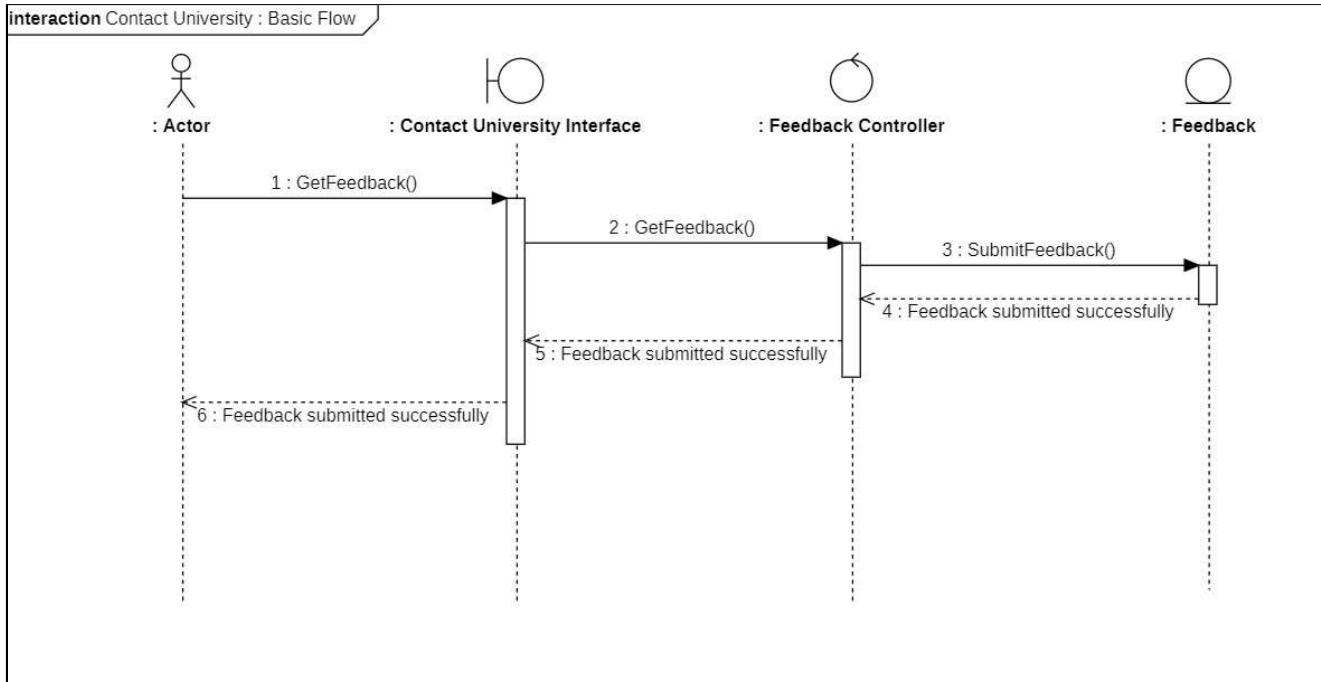


VIEW STUDENT-COURSE DETAILS BASIC FLOW 2:VIEW DETAIL BY FACULTY

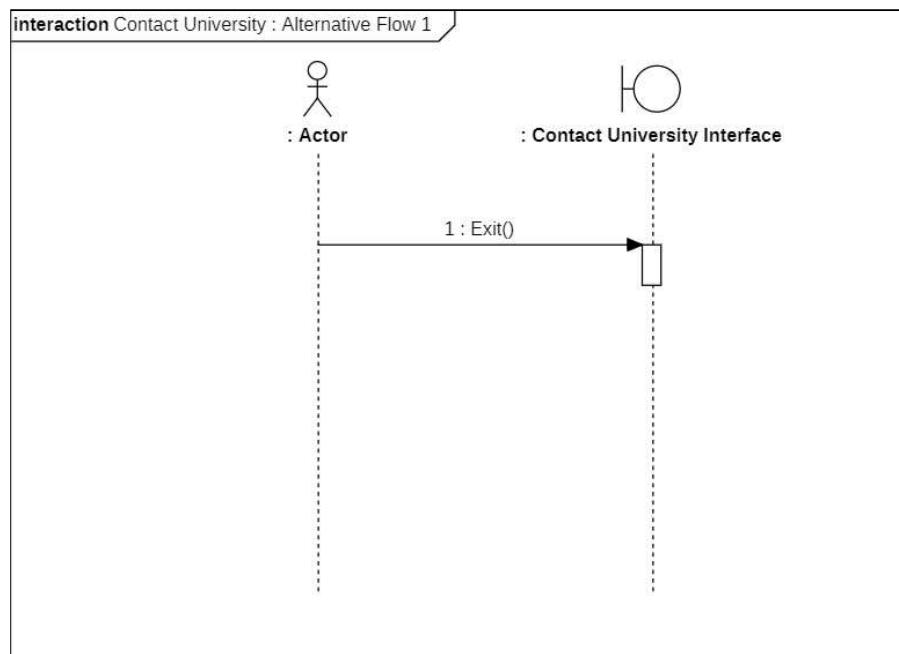


9. CONTACT UNIVERSITY

CONTACT UNIVERSITY BASIC FLOW :

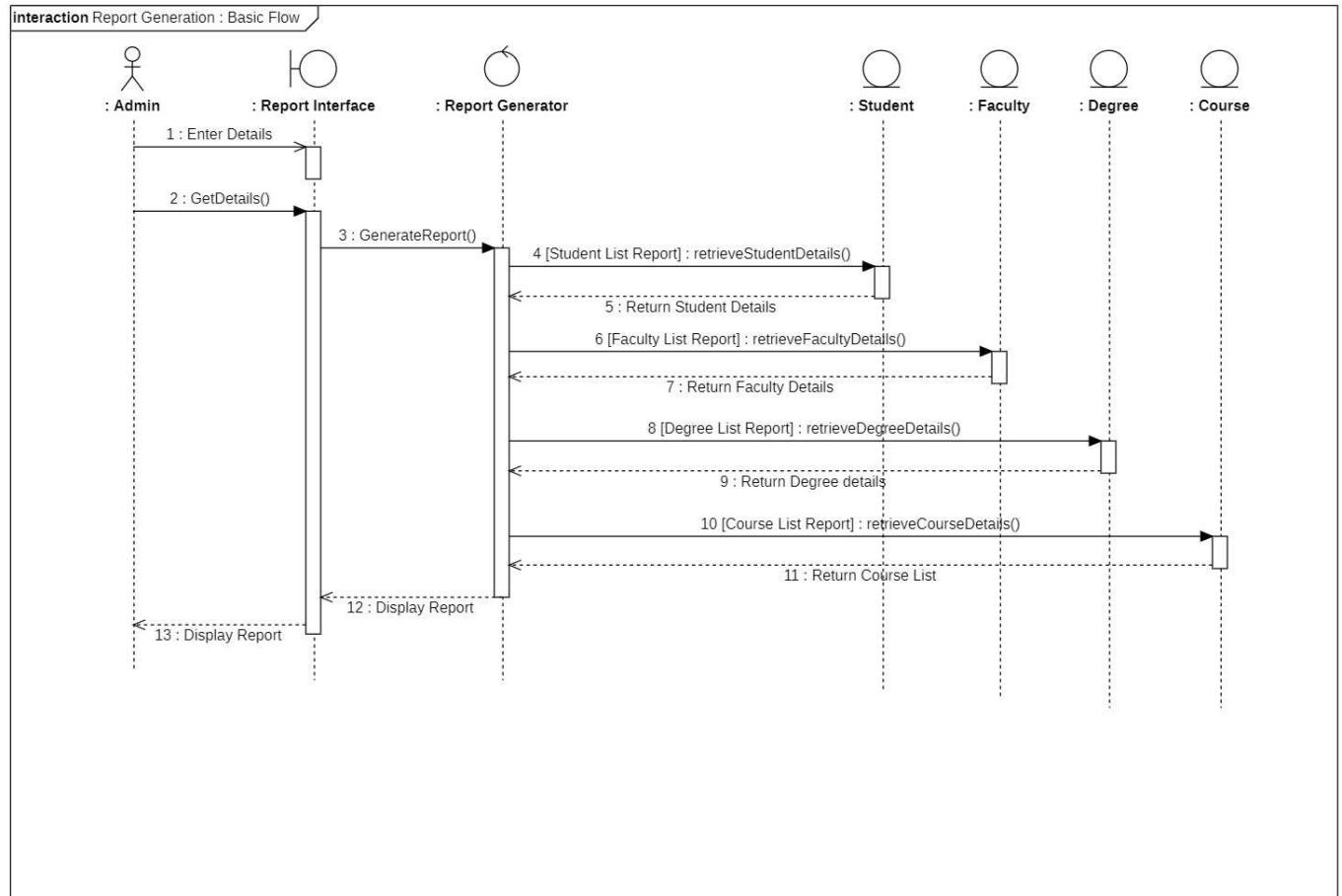


CONTACT UNIVERSITY ALTERNATIVE FLOW 1: USER EXITS

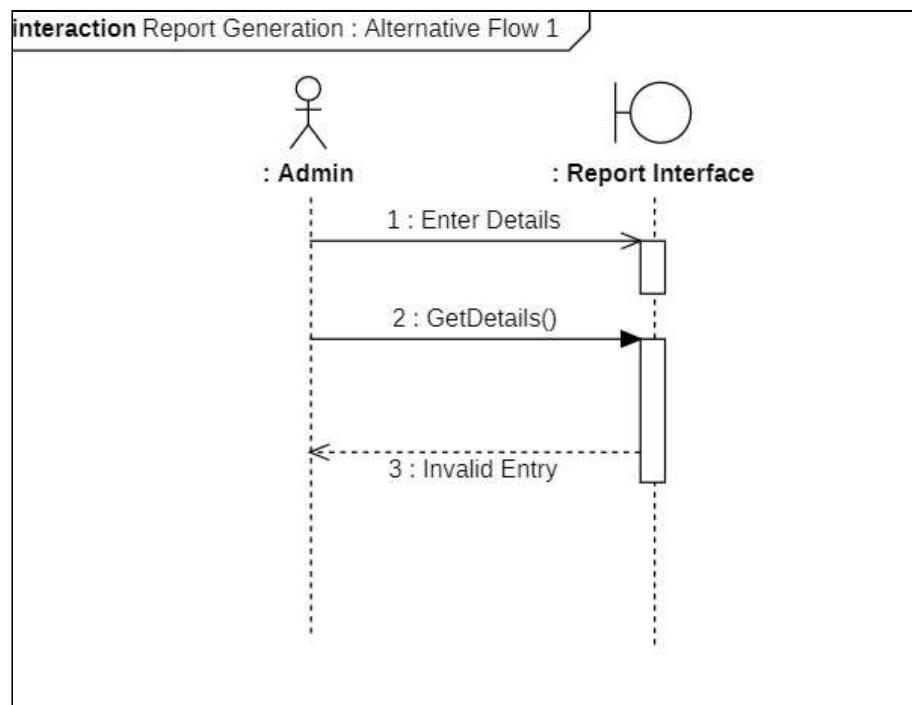


10. REPORT GENERATION

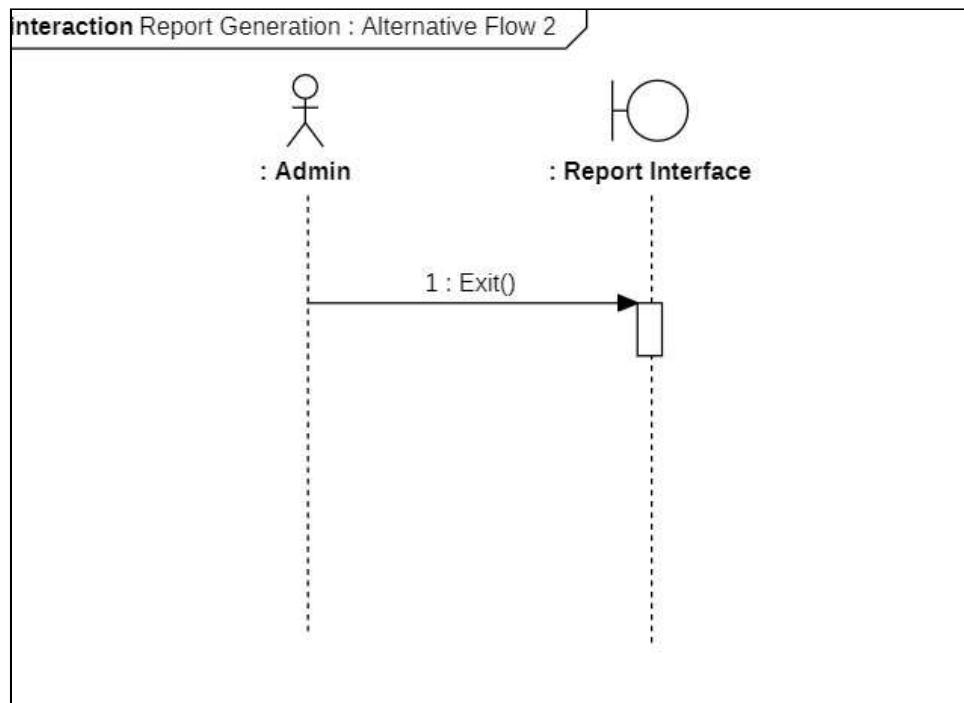
REPORT GENERATION BASIC FLOW:



REPORT GENERATION ALTERNATIVE FLOW 1:INVALID ENTRY

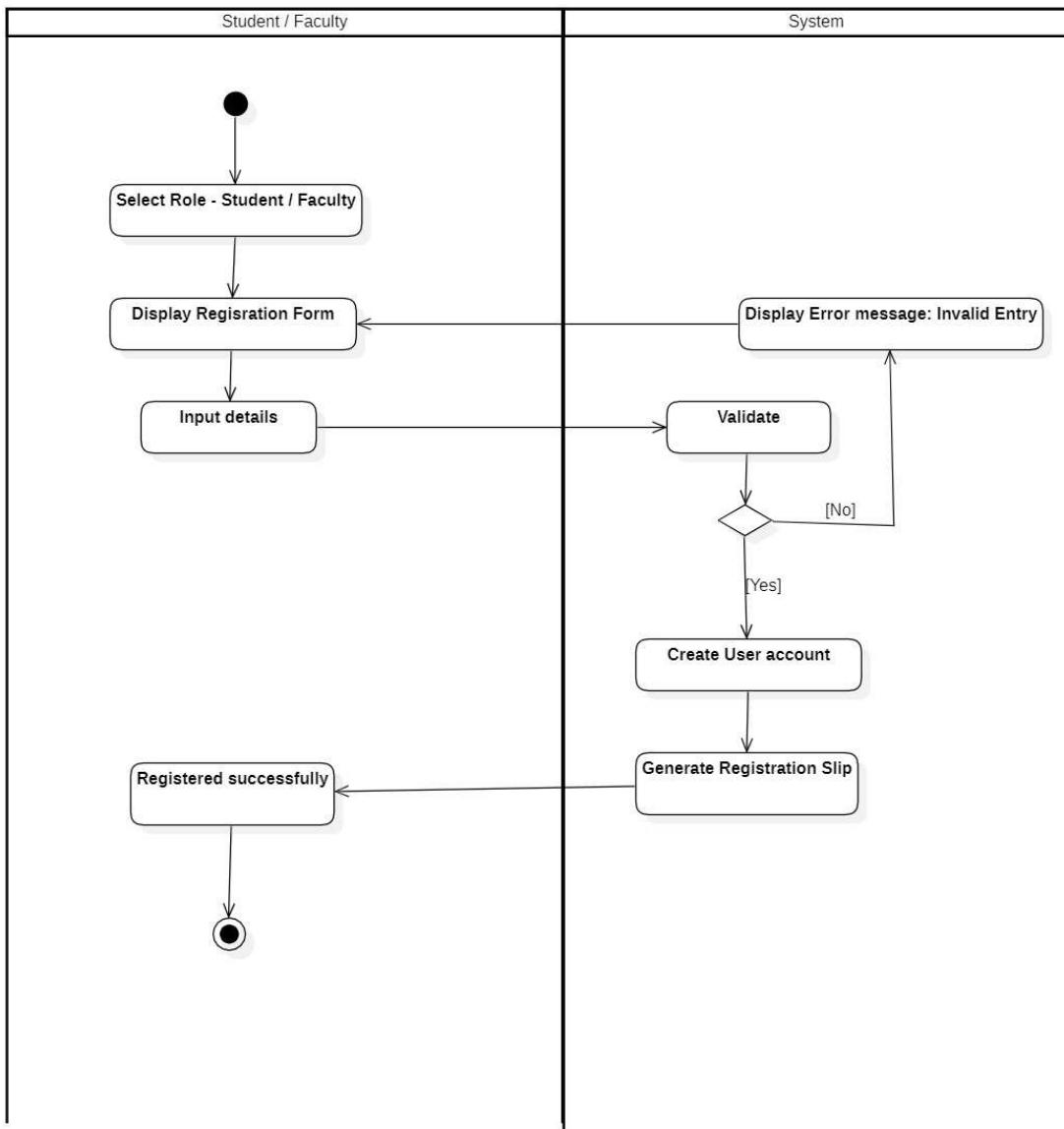


REPORT GENERATION ALTERNATIVE FLOW 2:USER EXITS

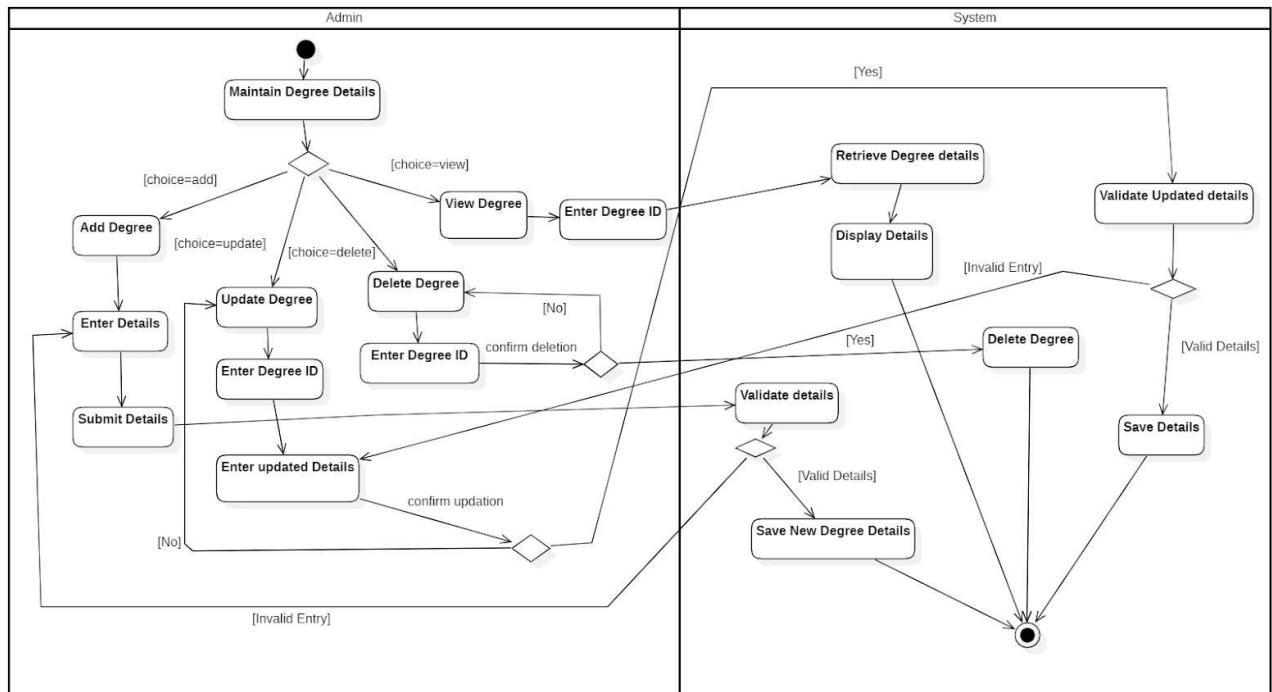


ACTIVITY DIAGRAM

1. INITIAL REGISTRATION

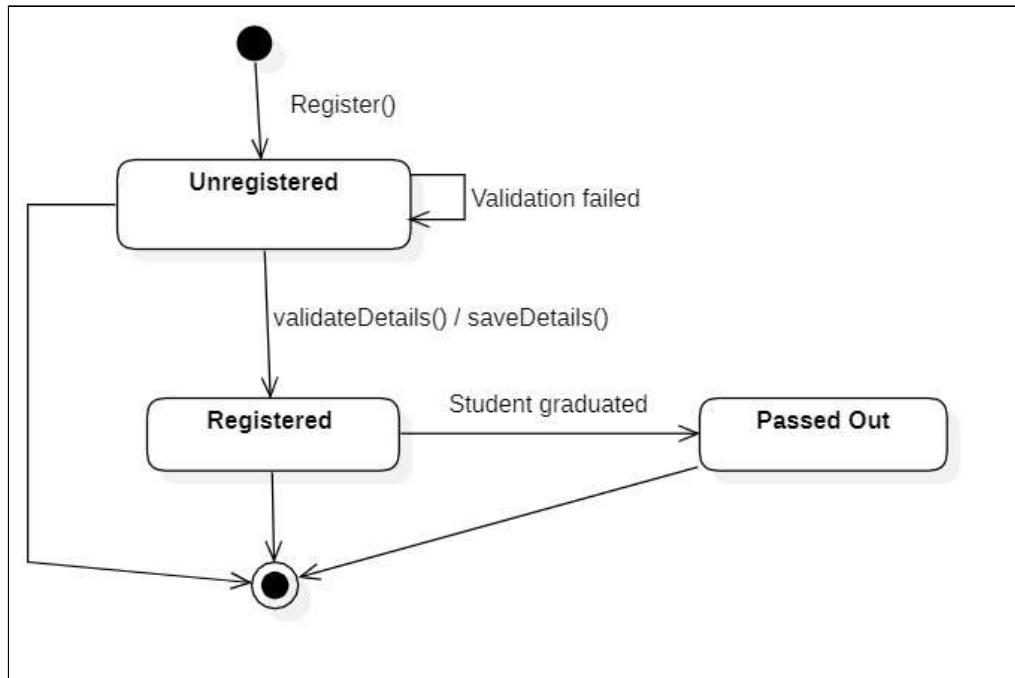


2. MAINTAIN DEGREE DETAILS

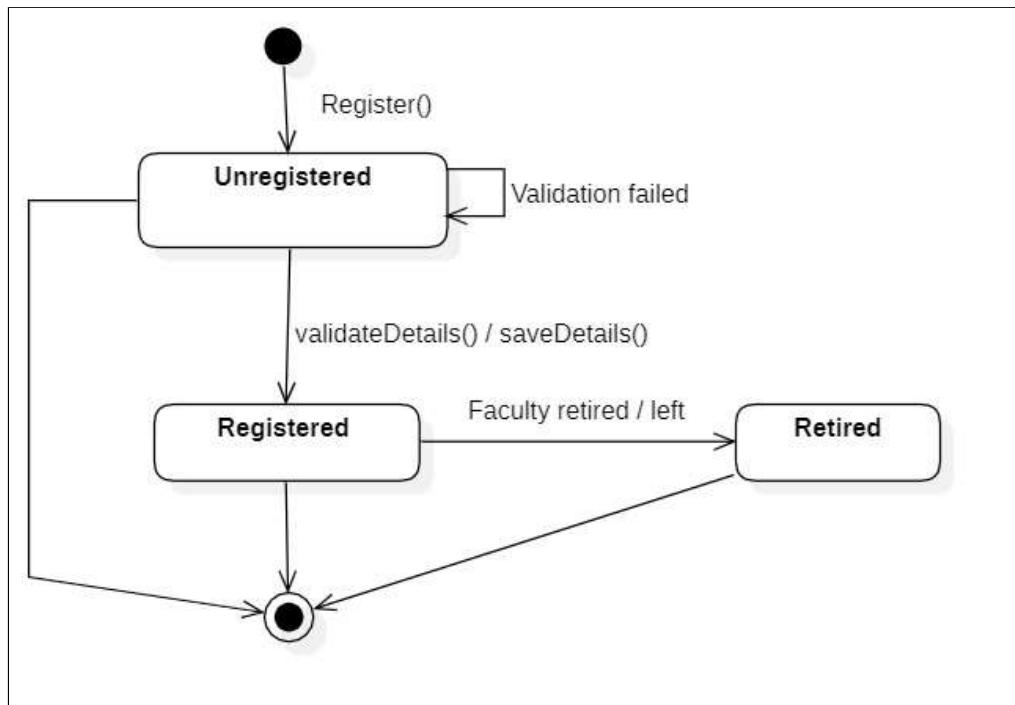


STATECHART DIAGRAM

1. Statechart diagram for Student class:



2. Statechart diagram for Faculty class:



TEST CASE MATRIX

1. MAINTAIN DEGREE DETAILS TEST CASES:

TEST CASE ID	SCENARIO AND DESCRIPTION	INPUT 1- DEGREE ID	INPUT 2- DEGREE NAME	INPUT 3 - STUDENT IDs	UPDATE CONFIRMED	DELETION CONFIRMED	EXPECTED RESULT	REMARKS (IF ANY)
TC1	Scenario 1- Add a degree	101	Computer Science	401, 402	n/a	n/a	Degree is added successfully	
TC2	Scenario 2- Add a degree alternative flow: Invalid entry	AB12	*	*	n/a	n/a	Invalid Degree ID	Degree ID is not in the specified format. Degree Name become 'do not care'
TC3		101	CS64@	*	n/a	n/a	Invalid Degree Name	Degree Name is not in the specified format
TC4		101	Computer Science	672,987	n/a	n/a	Invalid Student IDs	Student ID of students in the degree are not in the specified format
TC5	Scenario 3- Add a degree alternative flow: Student already enrolled in another degree	101	Computer Science	401, 402	n/a	n/a	Student already enrolled in another degree	One of the Student ID of a student already exists in the degree database
TC6	Scenario 4- Add a degree alternative flow : Degree already exists	101	Computer Science	401, 402	n/a	n/a	Degree ID already exists	The degree with the same Degree ID is already present in the database
TC7	Scenario 5- Add a degree alternative flow : User exits	Valid/ Invalid input	*	*	n/a	n/a	User is allowed to exit and returns to Main Menu	
TC8	Scenario 6- Update a degree	101	Computer Science	401, 402	Yes	n/a	Degree is updated successfully	
TC9	Scenario 7- Update a degree alternative flow: Invalid entry	Invalid input	n/a	n/a	n/a	n/a	Invalid Degree ID	Degree ID is not in the specified format. Degree Name become 'do not care'
TC10		101	CS64@	*	n/a	n/a	Invalid Degree Name	Degree Name is not in the specified format.

TC11		101	Computer Science	672,987	n/a	n/a	Invalid Student IDs	Student ID of students in the degree are not in the specified format
TC12	Scenario 8- Update a degree alternative flow: Student already enrolled in another degree	101	Computer Science	401, 402	n/a	n/a	Student already enrolled in another degree	One of the Student ID of a student already exists in the degree database
TC13	Scenario 9- Update a degree alternative flow: Degree Not Found	101	n/a	n/a	n/a	n/a	Degree Not Found	Degree with specified ID does not exist in the database
TC14	Scenario 10- Update a degree alternative flow: Update cancelled	101	Computer Science	401, 402	No	n/a	Main screen of degree appears	
TC15	Scenario 11- Update a degree alternative flow : User exits	*	*	*	n/a	n/a	User is allowed to exit and returns to Main Menu	
TC16	Scenario 12- Delete a degree	101	n/a	n/a	n/a	Yes	Degree is deleted successfully	
TC17	Scenario 13- Delete a degree alternative flow: Degree Not Found	110	n/a	n/a	n/a	n/a	Degree Not Found	Degree with specified ID does not exist in the database
TC18	Scenario 14- Delete a degree alternative flow: Delete cancelled	101	n/a	n/a	n/a	No	Main screen of degree appears	
TC19	Scenario 15- Delete a degree alternative flow : User exits	*	n/a	n/a	n/a	n/a	User is allowed to exit and returns to Main Menu	
TC20	Scenario 16- View a degree	101	n/a	n/a	n/a	n/a	Degree is displayed successfully	The degree details are displayed
TC21	Scenario 17- View a degree alternative flow: Degree Not Found	110	n/a	n/a	n/a	n/a	Degree Not Found	Degree with specified ID does not exist in the database
TC22	Scenario 18- View a degree alternative flow : User exits	*	n/a	n/a	n/a	n/a	User is allowed to exit and returns to Main Menu	

2. MAINTAIN COURSE DETAILS:

TEST CASE ID	SCENARIO AND DESCRIPTION	INPUT 1-COURSE ID	INPUT 2-COURSE NAME	INPUT 3-FACULTY ID	INPUT 4-NO. OF CREDITS	INPUT 5-NO. OF SEATS	INPUT 6-STUDENT IDs	UPDATE CONFIRMED	DELETION CONFIRMED	EXPECTED RESULT	REMARKS (IF ANY)
TC1	Scenario 1-Add a course	201	Data Structures	301	4	5	401,402	n/a	n/a	Course is added successfully	
TC2	Scenario 2-Add a course alternative flow: Invalid entry	64AB	*	*	*	*	*	n/a	n/a	Invalid Course ID	Course ID is not in the specified format.
TC3		201	DS67@	*	*	*	*	n/a	n/a	Invalid Course Name	Course Name is not in the specified format.
TC4		201	Data Structures	76S	*	*	*	n/a	n/a	Invalid Faculty ID	Faculty ID is not in the specified format
TC5		201	Data Structures	301	Ad	*	*	n/a	n/a	Invalid No. of credits	No. of credits is not in the specified format i.e. not a numeral
TC6		201	Data Structures	301	4	Kj	*	n/a	n/a	Invalid No. of seats	No. of seats is not in the specified format i.e. not a numeral
TC7		201	Data Structures	301	4	5	672, 873	n/a	n/a	Invalid Student IDs	Student IDs of students enrolled in the course are not in the specified format
TC8	Scenario 3-Add a course alternative flow : No. of students exceeds no. of seats	201	Data Structures	301	4	5	401,402	n/a	n/a	No. of Students exceeds no. of seats	
TC9	Scenario 4-Add a course alternative flow : Course already exists	201	Data Structures	301	4	5	401,402	n/a	n/a	Course ID already exists	The course with the same Course ID is already present in the database

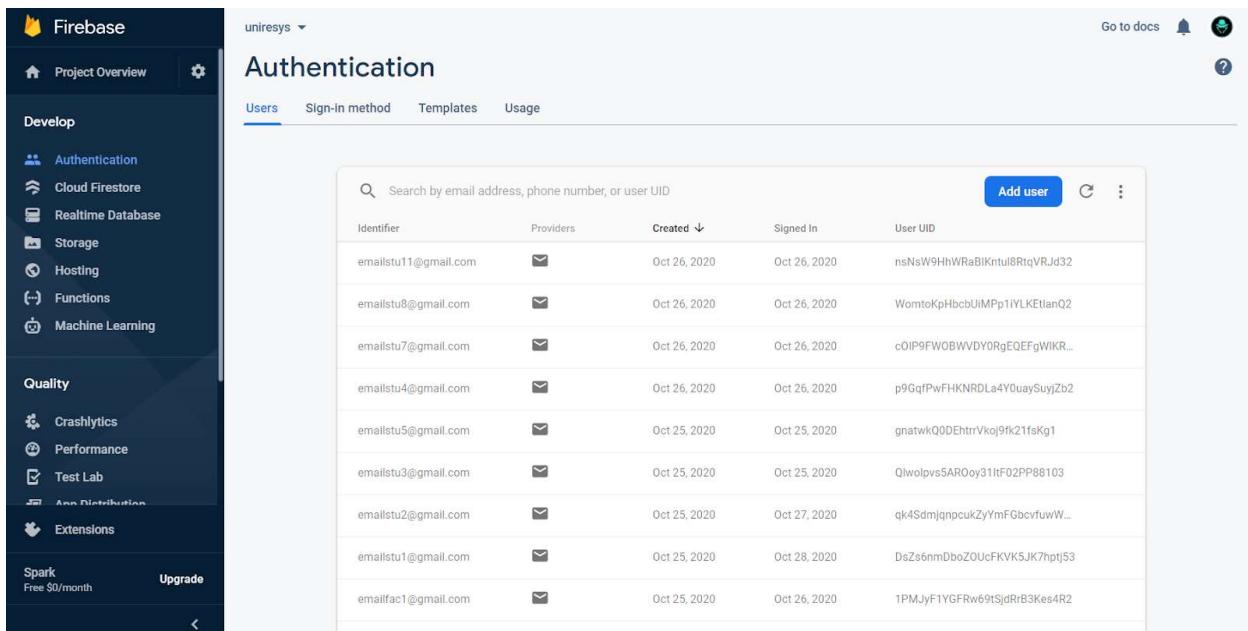
TC1 0	Scenario 5- Add a course alternative flow : User exits	*	*	*	*	*	*	n/a	n/a	User is allowed to exit and returns to Main Menu	
TC1 1	Scenario 6- Update a course	201	Data Structur es	301	4	5	401,402	Yes	n/a	Course is updated successfull y	
TC1 2	Scenario 7- Update a course alternative flow: Invalid entry	201	*	*	*	*	*	n/a	n/a	Invalid Course ID	Course ID is not in the specified format
TC1 3		201	DS67@	*	*	*	*	n/a	n/a	Invalid Course Name	Course Name is not in the specified format.
TC1 4		201	Data Structur es	76S	*	*	*	n/a	n/a	Invalid Faculty ID	Faculty ID is not in the specified format
TC1 5		201	Data Structur es	301	Ad	*	*	n/a	n/a	Invalid No. of credits	No. of credits is not in the specified format i.e. not a numeral
TC1 6		201	Data Structur es	301	4	Kj	*	n/a	n/a	Invalid No. of seats	No. of seats is not in the specified format i.e. not a numeral
TC1 7		201	Data Structur es	301	4	5	672, 873	n/a	n/a	Invalid Student IDs	Student IDs of students enrolled in the course are not in the specified format
TC1 8	Scenario 8- Add a course alternative flow : No. of students exceeds no. of seats	201	Data Structur es	301	4	5	401,402	n/a	n/a	No. of Students exceeds no. of seats	
TC1	Scenario 9-	201	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Course Not	Course with

9	Update a course alternative flow: Course Not Found								Found	specified ID does not exist in the database
TC2-0	Scenario 10- Update a course alternative flow: Update cancelled	201	Data Structures	301	4t	5	401,402	n/a	n/a	Main screen of course appears
TC2-1	Scenario 11- Update a course alternative flow : User exits	*	*	*	*	*	*	n/a	n/a	User is allowed to exit and returns to Main Menu
TC2-2	Scenario 12- Delete a course	201	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Course is deleted successfully
TC2-3	Scenario 13- Delete a course alternative flow: Course Not Found	201	n/a	n/a	n/a	n/a	n/a	n/a	Course Not Found	Course with specified ID does not exist in the database
TC2-4	Scenario 14- Delete a course alternative flow: Delete cancelled	201	n/a	n/a	n/a	n/a	n/a	n/a	No	Main screen of course appears
TC2-5	Scenario 15- Delete a course alternative flow : User exits	*	n/a	n/a	n/a	n/a	n/a	n/a	n/a	User is allowed to exit and returns to Main Menu
TC2-6	Scenario 16- View a course	201	n/a	n/a	n/a	n/a	n/a	n/a	n/a	Course is displayed successfully
TC2-7	Scenario 17- View a course alternative flow: Course Not Found	201	n/a	n/a	n/a	n/a	n/a	n/a	CourseNot Found	Course with specified ID does not exist in the database
TC2-8	Scenario 18- View a course alternative flow : User exits	*	n/a	n/a	n/a	n/a	n/a	n/a	n/a	User is allowed to exit and returns to Main Menu

IMPLEMENTATION DETAILS

Front end of the University Registration System (URS) has been developed using Google Flutter Software Development Kit. Back end of URS has been developed using Google Firebase.
(Screenshots of the front end of the system has been attached in SRS)

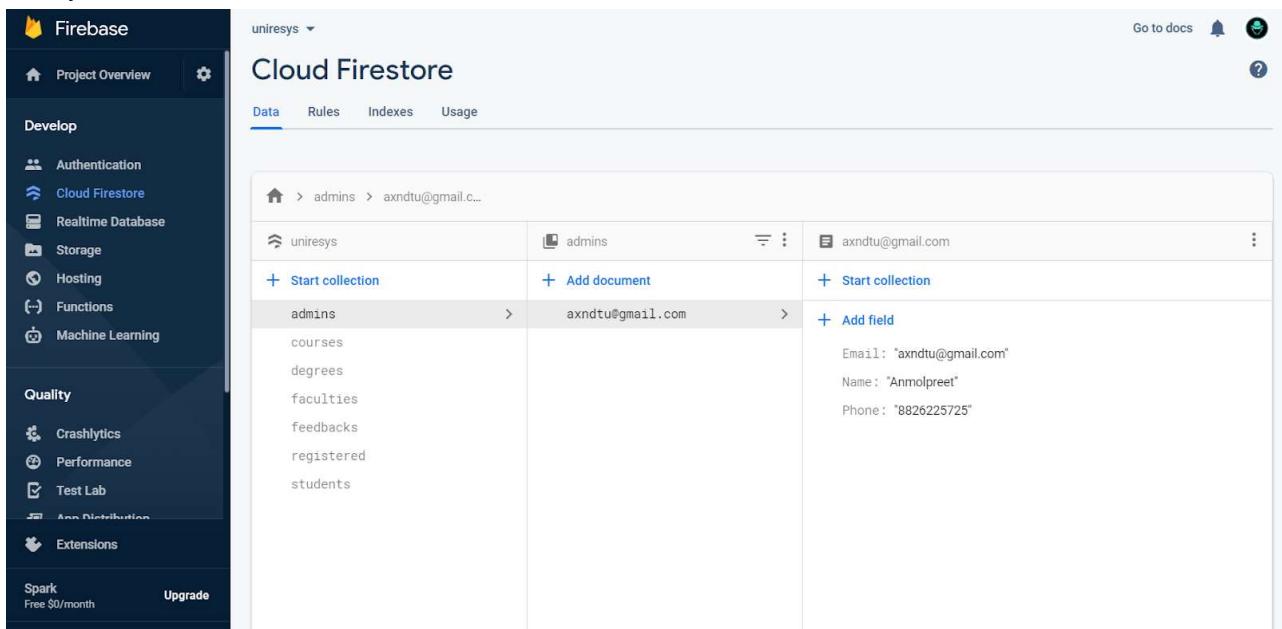
Using FirebaseAuth in Google Firebase for authentication of users of the system to provide authorised access to the system and maintain security.



The screenshot shows the Firebase Authentication console for a project named "uniressys". The left sidebar includes options for Project Overview, Develop (Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Quality (Crashlytics, Performance, Test Lab, App Distribution, Extensions), and Spark (Free \$0/month, Upgrade). The main panel is titled "Authentication" and shows a table of users. The columns are Identifier, Providers, Created, Signed In, and User UID. The table lists nine users, all of whom signed in on October 26, 2020, and have different User UIDs.

Identifier	Providers	Created	Signed In	User UID
emailstu11@gmail.com	[Email]	Oct 26, 2020	Oct 26, 2020	nsNsW9HhWRaBkIntui8RtqVRJd32
emailstu8@gmail.com	[Email]	Oct 26, 2020	Oct 26, 2020	WomtoKpHbcbUiMPp1YLKEtianQ2
emailstu7@gmail.com	[Email]	Oct 26, 2020	Oct 26, 2020	cOIP9FWOBWVVDY0RgEQEFgWIKR...
emailstu4@gmail.com	[Email]	Oct 26, 2020	Oct 26, 2020	p9GqfPwFHKNRDLa4Y0uaySuyjZb2
emailstu5@gmail.com	[Email]	Oct 25, 2020	Oct 25, 2020	gnatwkQ0DEhtrrVkoj9fk21fsKg1
emailstu3@gmail.com	[Email]	Oct 25, 2020	Oct 25, 2020	Qlwolpv5AROoy31ltF02PP88103
emailstu2@gmail.com	[Email]	Oct 25, 2020	Oct 27, 2020	qk4SdmjqnpckzYmFGbcvfuwW...
emailstu1@gmail.com	[Email]	Oct 25, 2020	Oct 28, 2020	DsZs6nmDboZOUcFKVK5JK7hptj53
emailfac1@gmail.com	[Email]	Oct 25, 2020	Oct 26, 2020	1PMJyF1YGFw69tSjdRrB3Kes4R2

The main database of the system has been developed using Cloud Firestore of Google Firebase to maintain and store entity classes



The screenshot shows the Firebase Cloud Firestore console for the same project "uniressys". The left sidebar is identical to the previous screenshot. The main panel is titled "Cloud Firestore" and shows a hierarchical document structure under the "admins" collection. The "admins" collection contains documents for "axndtu@gmail.com" and "axndtu@gmail.com". The "axndtu@gmail.com" document contains subcollections for "courses", "degrees", "faculties", "feedbacks", "registered", and "students". To the right of the document structure, there is a detailed view of the "axndtu@gmail.com" document, showing fields: Email: "axndtu@gmail.com", Name: "Anmolpreet", and Phone: "8826225725".

GIT REPOSITORY LINK: <https://github.com/axnmol/Uniresys>

VIDEO LINK: <https://drive.google.com/file/d/1eu4v6kAOzvpGOKuZYs7PZXctO0csTpVO/view?usp=sharing>