

# Martial Arts Gym Information Tracking System (MAGITS)

---

## Software Requirement Analysis and Design Document

**Version 1.0**

November 11, 2019

## TABLE OF CONTENTS

<b>1.0 Introduction</b>	3
1.1 Purpose	3
1.2 Scope	3
<b>2.0 System Overview</b>	5
Project Perspective	5
System Context	5
General Constraints	5
Assumptions	6
Internal Dependencies	6
External Dependencies	6
<b>3.1 Functional Requirements</b>	6
3.1.1 New User Functional Requirement	6
3.1.2 Member Functional Requirement	7
3.1.3 Instructor Functional Requirement	9
3.1.4 Admin Function Requirements	13
<b>3.2 Use Cases</b>	21
3.2.1 New User Use Case	21
3.2.2 Member Use Case	22
3.2.3 Instructor Use Case	23
3.2.4 Admin Use Case	26
<b>3.3 Data Modelling and Analysis</b>	32
3.3.1 Normalized Data Model Diagram	32
3.3.2 Activity Diagram	<b>Error! Bookmark not defined.</b>
3.3.3 Sequence Diagrams	44
3.3.4 UML Class Diagram	87
<b>3.4 Process Modelling</b>	89
3.4.1 Data Flow Diagram lvl 0	89
3.4.2 Data Flow Diagram lvl 1	89
3.4.3 Data Flow Diagram lvl 2	90
<b>4.0 Non-Functional Requirements</b>	95
<b>5.0 Logical Database Requirements</b>	97
<b>6.0 Other Requirements</b>	102
<b>7.0 Approval</b>	102

**1. 1.0 Introduction**

The Introduction section provides an overview of the system using software requirements analysis and design for the scope of the system.

**a. 1.1 Purpose**

This document describes the high level software requirements for the system. It describes the what, not how, of the capabilities of the system for the intended audiences.

**b. 1.2 Scope**

This explains what the proposed system will and will not do. Describe relevant benefits, objectives and goals. The description of scope should be consistent with the Project Plan.

In Scope	Objective	Scope Description
<b>1. Course Management</b>	Course Management Scope pertains to the management of the Courses in the system. It is for the Admin to be able to create a Course where Instructors can be assigned and for members to attend.	<ul style="list-style-type: none"> <li>• Admin can add, view, edit and remove courses in the system</li> <li>• Admin can add and remove an instructor to a course</li> <li>• Admin can enroll and remove a student from a course</li> <li>• Instructor can enroll and remove a member from a course</li> </ul>
<b>2. Instructor Management</b>	Instructor Management Scope pertains to the management of the Instructors in the system. It is for the Admin to be able to Add an Instructor and track Instructors of the system. An Instructor will be added to a Course in the Course Management.	<ul style="list-style-type: none"> <li>• Admin can add, edit, remove and view all instructors in the system</li> </ul>
<b>3. Member Acct. Management</b>	Member Acct. Mgmt Scope pertains to the management of the Member Accounts in the system. It is for the Admin to be able to register a non-member to the system when they don't have the means to register themselves.	<ul style="list-style-type: none"> <li>• Admin can register a member, edit, delete and view all members in the system</li> <li>• Instructor can register a member to the system.</li> <li>• Non-member can register to the system online.</li> <li>• Member can edit and view it's account</li> </ul>

<b>4. Membership Type Management</b>	Membership Type Mgmt Scope pertains to the management of Membership Types in the system. The Admin can create memberships which Members can purchase to be able to join a Course.	<ul style="list-style-type: none"> <li>● As an admin, I want to create a membership type for member's to choose which payment package they want from</li> <li>● As an admin, I want to delete a membership type</li> <li>● As an admin, I want to view a membership type</li> <li>● As an admin, I want to update a membership type</li> </ul>
<b>5. Announcements Management</b>	Announcements Mgmt. Scope pertains to the management of Announcements in the system. An Admin and Instructor will be able to create and send announcements to easily reach the members of the system or of a Course.	<ul style="list-style-type: none"> <li>● Admin can add, edit, delete and view all announcements</li> <li>● Instructor can add, edit, delete and view all announcements</li> <li>● Member can view announcements</li> <li>● Instructor can send an announcement to a member's email</li> </ul>
<b>6. Session Management</b>	Session Mgmt. Scope pertains to the management of Sessions in the system. An Instructor can create a Session to be able to start a class. When a Session starts, the Instructor can then check-in a Member for their attendance.	<ul style="list-style-type: none"> <li>● Instructor can start/open a session for the member to check-in</li> <li>● Instructor can check-in a student to a session</li> <li>● Admin can view all attendees in a session that is stored in history</li> <li>● Admin can view all the feedback of the session.</li> </ul>
<b>7. Session Feedback Feature</b>	Session Feedback Feature is where a member can leave a feedback to Session when a Session is closed.	<ul style="list-style-type: none"> <li>● A member in a session can leave a feedback on the session</li> </ul>
<b>8. Login and Logout Feature</b>	To be able to access the system, users must be able to login.	<ul style="list-style-type: none"> <li>● Admin, instructor and member can log-in to be able to use the system.</li> <li>● Users of the system can also log-out from their account.</li> </ul>

Out of Scope	Description
1. Payment System	<ul style="list-style-type: none"> <li>This will manage the transactions made by members when they make a payment.</li> </ul>
2. Inventory Management	<ul style="list-style-type: none"> <li>This will store information regarding maintenance of gym equipment.</li> </ul>
3. Employee Management	<ul style="list-style-type: none"> <li>This includes scheduling work date for the workers as well as their payment.</li> </ul>
4. Social Management	<ul style="list-style-type: none"> <li>This includes a social media platform within the system.</li> </ul>
5. Social Messaging	<ul style="list-style-type: none"> <li>This system does not include sending messages between the members.</li> </ul>
6. Payroll	<ul style="list-style-type: none"> <li>This system calculates the salary of each employee.</li> </ul>

## 2. 2.0 System Overview

The System Overview section introduces the system context and design.

a.

b.

### c. Project Perspective

The system will implement new technologies to replace an existing informal system that the martial arts gym uses. The old system involves a manual process of storing and retrieving physical profile sheets of students. It also involves manually recording student attendance by hand. There is no data tracking or any kind of technical system in place to automate any process. Thus, this is the origin of the new system.

d.

### e. System Context

A web application that automates business processes and solves common problems that small Martial Arts Gyms face in their everyday operations. The project will enable the gym to keep digital records of students' profiles, monitor the remaining sessions on their account, track their attendance and use modern scanning technologies such as a QR code to submit attendance.

f.

### g. General Constraints

These are the processes and external constraints that will hinder the system from running.

1. The system will only work with an internet connection.

2. There is a downtime on the cloud service.
3. The business should have a machine where the web application will run.
4. The machine requires a camera to scan QR codes.

**h. Assumptions**

These are the assumptions that are based on the scope, requirements, business processes and stakeholders.

1. The system is for small businesses, which is defined as privately owned corporations, partnerships, or sole proprietorships that have fewer employees and/or less annual revenue than a regular-sized business or corporation.
2. Every session will have less than 50 members participating at any given day.
3. The MAG staff are computer literate.
4. The gym member always carries a device or a card that has their QR code for the session check-in.

**i. Internal Dependencies**

1. The user must be registered to be able to see the courses in the system.
2. The member should pay the membership before they are enrolled in a course by the instructor.
3. The user must be enrolled in a course to be able to have a qr code where their course session will be scanned and thus having their attendance tracked.
4. The admin or the instructor must be logged in to be able to use the qr scanner.

**j. External Dependencies**

1. The system will be utilizing a cloud based database thus it must be up at all times.

3.

4.

**5. 3.1 Functional Requirements****a. 3.1.1 New User Functional Requirement****3.1.1.1 Register Account**

Introduction: Adding new account to the database

Inputs: User information such as name, email, and date of birth.

Processing:

1. User must go to the registration page.
2. The user inputs required information on the form.
3. The system validates the required fields.
4. The system creates an account in the database.

Outputs: User account has been added to the database and a confirmation message was sent to the user's email.

**b. 3.1.2 Member Functional Requirement**

**3.1.2.1 View Courses**

Introduction: Displays a list of courses being taught at the gym.

Inputs: Member has an account.

Processing:

1. The member logs in to the app.
2. Member must go to the course list section
3. System will generate a list of courses available.
4. The member selects a course that he wants to view its schedules and information.

Outputs: A list of all the courses has been displayed

**3.1.2.2 Enroll to Course**

Introduction: User can take part in the lesson taught in the course.

Inputs: Member has an account.

Processing:

1. The member logs in to the app.
2. Member must go to the course list section.
3. System will generate a list of courses available.
4. The member then may select a course to enroll to.
5. System prompts the user for confirmation.
6. System registers the user's account to the course in the database.

Outputs: Database has been updated, details regarding enrollment is sent to the member's email.

**3.1.2.3 Cancel Subscription to a Course**

Introduction: The member must be able to cancel subscription to a course.

Inputs: Account that is currently enrolled in a course.

Processing:

1. Member logs in to the app.
2. Member navigates to the courses subscribed section.
3. The system generates a list of all the courses the user is subscribed to.
4. The member selects a course from the list he/she wishes to unsubscribe to.
5. The system prompts the user for confirmation.

Outputs: Member removed from that course and the database has been updated.

---

**3.1.2.4 View Announcements**

Introduction: Displays a list of announcements sent by admin/instructors.

Inputs: User has an account.

Processing:

1. The user logs in to the app.
2. The system will then populate the homepage with recent announcements regarding the gym and their classes.
3. The user can access all gym announcement by navigating to the announcement section.
4. The system generates a list of all the announcements.

Outputs: List of all gym announcements.

**3.1.2.5 View Profile**

Introduction: User may view details regarding his account.

Inputs: User must have an account.

Processing:

1. User logs in to the app.
2. User navigates to the profile section.
3. The system displays the user details.

Outputs: Displays basic information of the user such as name, date of birth, contact info, and his/her QR identification code.

**3.1.2.6 Edit Profile**

Introduction: Member updates his account information.

Inputs: Member has an account.

Processing:

1. Member logs in to the app.
2. Member navigates to the profile section.
3. The system displays the Member details in a form for modification.
4. Member updates the information.
5. The system validates the changes made.

Outputs: Database has been updated.

**3.1.2.7 Check-in to Session**



Introduction: Member checks in to the session for attendance

Inputs: Member has an account registered to a course.

Processing:

1. Member logs in to the app.
2. Member navigates to the profile section.
3. The system displays the user details as well as their qr identification code.
4. The member lets the code be scanned.
5. The system updates the session's attendance to mark this student present.

Outputs: Updated database, member's attendance recorded.

### **3.1.2.8 Send Feedback**

Introduction: User sends feedback regarding a session.

Inputs: User account registered to a course.

Processing:

1. User logs in to the app.
2. User navigates to the sessions attended section.
3. The system generates a list of all sessions attended.
4. The user selects the session he/she wants to send a feedback to.
5. The system displays a form with a textfield and submit button.
6. The user fills the form with his feedback and submits it.

Outputs: Feedback for a course has been added to the database.

## **c. 3.1.3 Instructor Functional Requirement**

### **3.1.3.1 Register a new User**

Introduction: Adding a new user account to the database

Inputs: Instructor account and new user information.

Processing:

1. Instructor must log in to the app.
2. User must go to the register user page.
3. The system displays a form for input.
4. The Instructor inputs the user information and submits.
5. The system validates the input.

Outputs: User account has been added to the database and a confirmation message was sent to the user's email.

### **3.1.3.2 Enroll Members to a Course**

Introduction: Instructor adds a member to the course he/she is managing.

Inputs: Instructor has authority to add members to a course.

Processing:

1. Instructor logins to the app.
2. The instructor navigates to the course section.

3. The system displays the course information, add member form, and lists of all members enrolled in the course.
4. The instructor fills in the form.
5. The system validates that the user is allowed to enroll in a course.

Outputs: Course information and course member lists, update database, notification sent to the user added from the course.

### **3.1.3.3 Remove Member from Course**

Introduction: Instructor removes a member from a course

Inputs: Instructor has authority to remove members from a course.

Processing:

1. The Instructor logs in to the app.
2. The Instructor navigates to the course section.
3. The system displays lists of all members enrolled in the course.
4. The instructor selects the user he/she wants to remove.

Outputs: Course information and course member lists, update database, notification sent to the user removed from the course.

*i.*

### **3.1.3.4 View Instructor Course Schedule**

Introduction: Instructor views information regarding the courses their teaching.

Inputs: Instructor account assigned to a course.

Processing:

1. The Instructor logs in to the app.
2. The Instructor navigates to the course section.
3. The system displays lists of all courses assigned to the instructor.
4. The Instructor selects which course he/she wants to view.
5. The system displays the information of the course as well as its schedule.

Outputs: Course information displayed.

### **3.1.3.5 Start Session**

Introduction: Instructor starts a session for attendance tracking

Inputs: Instructor has the ability to start a session.

Processing:

1. Instructor logs in to the app.
2. The instructor navigates to the manage sessions section.

3. The system displays the info of the session such as the course being taught, the date and time of session, a member attendance list, and session start button.
4. The instruction press the session start button to start the session and allow member id to be scanned.

Outputs: Session started, notifies the members enrolled in the course of that session.

### **3.1.3.6 Scan/Check-in member identification**

Introduction: Instructor sets up a scanner to scan members identification code for attendance.

Inputs: Instructor has the authority to start session and scan for id code.

Processing:

6. Instructor logins to the app.
7. Instructor navigates to the manage session section.
8. The system displays the info of the session such as the course being taught, the date and time of session, a member attendance list, and session start button.
9. The instruction press the session start button to start the session and allow member id to be scanned.

Outputs: Session started, allows scanner to scan ID code, list of attendees displayed.

### **3.1.3.7 View Session Feedback**

Introduction: Instructor views feedback regarding a session he/she taught.

Inputs: Instructor has access to the sessions page.

Processing:

1. Instructor logins to the app.
2. Instructor navigates to the session page.
3. The system displays the session information such as course being taught at that session, schedule and user feedbacks.

Outputs: List of session feedback from course members.

### **3.1.3.8 Send Announcement**

Introduction: Instructor creates and sends announcements to every member.

Inputs: Instructor has access to manage announcements page.

Processing:

1. Instructor logins to the app.
2. Instructor navigates to the manage announcements page.
3. The system displays a list of all announcements and a form with a textfield and submit button for announcement creation.

4. The Instructor fills out the announcement creation form.
5. The system validates the input fields.

Outputs: Announcement has been added to the database and all users has been notified.

#### **3.1.3.9 View Announcements**

Introduction: Displays a list of announcements.

Inputs: Instructor has an account.

Processing:

1. The instructor logs in to the app.
2. The system will then populate the homepage with recent announcements regarding the gym and their classes.
3. The instructor can access all gym announcement by navigating to the manage announcements section.
4. The system generates a list of all the announcements.

Outputs: List of all gym announcements.

#### **3.1.3.10 Edit Announcement**

Introduction: Edit announcements information.

Inputs: Instructor has access to the manage announcements page.

Processing:

1. The Instructor logs in to the app.
2. Instructor navigates to the manage announcements page
3. The system generates a list of all the announcements with edit/delete button only on the announcements that this instructor has sent.
4. The Instructor selects the announcement he/she wants to edit and click the edit button.
5. The system will display the announcement info in a form for modification.
6. The Instructor updates the information.

Outputs: List of all gym announcements, updates the database and notifies users.

#### **3.1.3.11 Delete Announcement**

Introduction: Deletes the announcements sent by this Instructor.

Inputs: Instructor has access to the manage announcements page.

Processing:

1. The Instructor logs in to the app.
2. Instructor navigates to the manage announcements page
3. The system generates a list of all the announcements with edit/delete button only on the announcements that this instructor has sent.
4. The Instructor selects the announcement he/she wants to delete.
5. The system prompts the Instructor for confirmation.

6. The instructor confirms the action.

Outputs: List of the announcements and announcement removed from the database.

### **3.1.3.12 View Session Attendance Summary**

Introduction: Instructor views the attendance summary of the sessions taught.

Inputs: Instructor has authority to access the session summary page.

Processing:

1. The instructor logs in to the app.
2. The instructor navigates to the sessions summary page.
3. The system displays a list of sessions he/she has completed.
4. The instructor selects a session to view its attendance summary
5. The system displays the session details such as the date and course taught as well as the session attendance summary.

Outputs: Session summaries for each session.

## **a. 3.1.4 Admin Function Requirements**

### **3.1.4.1 View Session Attendance Summary**

Introduction: Admin views the attendance summary of all the sessions.

Inputs: Admin has authority to access the session summary page.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the sessions page.
3. The system displays a list of all the sessions completed.
4. The admin selects a session to view its attendance summary
5. The system displays the session details such as the date and course taught as well as the session attendance summary.

Outputs: Session summaries for each session.

### **3.1.4.2 Creating a new course**

Introduction: Adding a new course to the database

Inputs: Admin account with access to course manage section.

Processing:

1. Admin must log in to the app.
2. Admin must go to the manage course page.
3. The system displays a list of all the courses in the system, as well as a form field for course creation.
4. The admin inputs the course information in the form field and submits.
5. The system validates the input.

Outputs: Course has been added to the database.

**3.1.4.3 View Course Information**

Introduction: The admin views the details of all the courses.

Inputs: Admin has authority to the manage course section.

Processing:

1. Admin logs in to the app.
2. The admin navigates to the course section
3. The system displays a list of all the courses in the system, as well as a form field for course creation.
4. The admin selects which course he wants to view from the list.
5. The system displays the course information.

Outputs: Course information displayed

**3.1.4.4 Edit Course Information**

Introduction: The Admin edits the details of all the courses.

Inputs: Admin has authority to the manage course section.

Processing:

1. Admin logs in to the app.
2. The admin navigates to the course section
3. The system displays a list of all the courses in the system with an edit and delete button appended, as well as a form field for course creation.
4. The admin selects which course he wants to edit and press the edit button.
5. The system displays the course details in a form field as well as a submit button.
6. The admin inputs the changes and clicks the submit button.

Outputs: Course information updated in the database.

**3.1.4.5 Delete a Course**

Introduction: Admin deletes a course from the database

Inputs: Has access to the manage course section.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the manage course section.
3. The system displays a list of all the courses in the system with an edit and delete button appended, as well as a form field for course creation.
4. The admin selects the course he wants to delete and press delete.
5. The system prompts the user to confirm action.

Outputs: The course is removed from the database.

**3.1.4.6 Add an Instructor**

Introduction: Admin adds an instructor to the database

Inputs: Has access to the instructor manager page.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the instructor manager page.
3. The system generates a form to input the instructor details as well as a list of all the instructors with a view/edit/delete button appended to each one..
4. Admin inputs the instructor details in the form fields.
5. System validates the input fields.

Outputs: Instructor added to the database.

---

#### **3.1.4.7 View Instructor Information**

Introduction: Admin views the details of the instructor.

Inputs: Has access to the instructor manager page.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the instructor manager page.
3. The system generates a form to input the instructor details as well as a list of all the instructors with a view/edit/delete button appended to each one.
4. The user selects the instructor he/she wants to view and clicks view.
5. The system displays the details regarding the instructor.

Outputs: List of all the instructors and their info.

---

#### **3.1.4.8 Edit Instructor Information**

Introduction: The admin edits the instructor details.

Inputs: Has authority to manage instructor page.

Processing:

1. The admin logs in to the app.
2. Admin navigates to the manage instructor page.
3. The system generates a form to input the instructor details as well as a list of all the instructors with a view/edit/delete button appended to each one.
4. Admin selects an instructor he/she chooses to edit and press edit.
5. The system displays a form field field with the current instructor details.
6. Admin inputs the changes he/she wishes to make.
7. The system validates the changes.

Outputs: Updated instructor details in the database.

---

#### **3.1.4.9 Delete an Instructor**

Introduction: The admin deletes an instructor from the database

Inputs: Has authority to manage instructor page.

Processing:

1. The admin logs in to the app.
2. Admin navigates to the manage instructor page.
3. The system generates a form to input the instructor details as well as a list of all the instructors with a view/edit/delete button appended to each one.
4. Admin selects an instructor he/she chooses to delete and clicks delete.
5. The system prompts the admin for confirmation.
6. The admin confirms.

Outputs: Removed instructor from the database.

---

#### **3.1.4.10 Add a member**

Introduction: Adding a new user account to the database

Inputs: Has access to the member manager page.

Processing:

1. Admin must log in to the app.
2. Admin navigates to the member manager page.
3. The system generates a form to input the user details and submit button as well as a list of all the members with a view/edit/delete button appended to each one.
4. The Admin inputs the user information in the form and submits.
5. The system validates the input.

Outputs: Member account has been added to the database and a confirmation message was sent to the user's email.

#### **3.1.4.11 View Member Information**

Introduction: Admin views the details of the member.

Inputs: Has access to the member manager page.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the member manager page.
3. The system generates a form to input the user details and submit button as well as a list of all the members with a view/edit/delete button appended to each one.
4. The user selects the member he/she wants to view and press view.
6. The system displays the details regarding the member.

Outputs: List of all the members and their info.

---

#### **3.1.4.12 Edit Member Information**

Introduction: Admin edits the details of the member.

Inputs: Has access to the member manager page.

Processing:

---



1. The admin logs in to the app.
2. The admin navigates to the member manager page.
3. The system generates a form to input the user details and submit button as well as a list of all the members with a view/edit/delete button appended to each one.
4. The user selects the member he/she wants to edit and press edit.
5. The system displays the details regarding the member in a form field.
6. The admin inputs the changes.
7. The system validates the changes.

Outputs: List of all the members and their info, member information updated in the database.

#### **3.1.4.13 Delete a Member**

Introduction: Admin deletes a member from the database.

Inputs: Has access to the member manager page.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the member manager page.
3. The system generates a form to input the user details and submit button as well as a list of all the members with a view/edit/delete button appended to each one.
4. The user selects the member he/she wants to delete and press delete.

Outputs: List of all the members and their info, member removed from the database.

#### **3.1.4.14 Send Announcement**

Introduction: Admin creates and sends announcements to every user.

Inputs: Admin has access to manage announcements page.

Processing:

1. Admin logs in to the app.
2. Admin navigates to the announcement manage announcements page.
3. The system displays a list of all announcements and a form with a textfield and submit button for announcement creation.
4. The Admin fills out the announcement creation form.
5. The system validates the input fields.

Outputs: Announcement has been added to the database and all users has been notified.

#### **3.1.4.15 View Announcements**

Introduction: Displays a list of all announcements.

Inputs: Admin account with access to manage announcements page.

Processing:

1. The admin logs in to the app.
2. The system will then populate the homepage with recent announcements regarding the gym and their classes.
3. The Admin can access all gym announcement by navigating to the manage announcement page.
4. The system generates a list of all the announcements as well as an view/edit/delete button appended to each one.

Outputs: List of all gym announcements and their details.

#### **3.1.4.16 Edit Announcement**

Introduction: Edit announcements information.

Inputs: Admin has access to the manage announcements page.

Processing:

1. The Instructor logs in to the app.
2. Admin navigates to the manage announcements page
3. The system generates a list of all the announcements as well as an view/edit/delete button appended to each one.
4. The Admin selects the announcement he/she wants to edit and click the edit button.
5. The system will display the announcement info in a form for modification.
7. The Admin updates the information and submits the changes.
8. The system validates the input.

Outputs: List of all gym announcements, updates the announcement from the database and notifies users.

#### **3.1.4.17 Delete Announcement**

Introduction: Deletes the announcements.

Inputs: Admin has access to the manage announcements page.

Processing:

1. The admin logs in to the app.
2. Admin navigates to the manage announcements page
3. The system generates a list of all the announcements as well as an view/edit/delete button appended to each one.
7. The Admin selects the announcement he/she wants to delete and press delete.
8. The system prompts the admin for confirmation.
9. The admin confirms the action.

Outputs: List of the announcements and announcement removed from the database.

#### **3.1.4.18 Add Membership Type**

Introduction: Creates a new membership type for the gym.

Inputs: Admin has access to the manage membership section, membership information.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the manage membership section.
3. The system generates a list of all the memberships with view/edit/delete button appended to each one as well as a form for membership creation.
4. The admin inputs the membership information and submits.
5. The system validates the inputs.

Outputs: Membership type added to the database.

#### **3.1.4.19 View Membership Type Information**

Introduction: View a list of all membership types.

Inputs: Admin has access to the manage membership section.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the manage membership section.
3. The system generates a list of all the memberships with view/edit/delete button appended to each one as well as a form for membership creation.
4. The admin selects which membership to view and press the view button.
6. The system displays the membership type information.

Outputs: List of all the membership type and their corresponding details.

#### **3.1.4.20 Edit Membership Type Information**

Introduction: Edit membership type details

Inputs: Admin has access to the manage membership section.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the manage membership section.
3. The system generates a list of all the memberships with view/edit/delete button appended to each one as well as a form for membership creation.
4. The admin chooses which membership type to edit and press the edit button.
5. The system will display the membership details in a form field for editing.
6. The admin inputs the changes.
7. The system validates the inputs.

Outputs: Edited membership type information in the database.

#### **3.1.4.21 Delete a Membership Type**

Introduction: Delete membership type from the database.

Inputs: Admin has access to the manage membership section.

Processing:

1. The admin logs in to the app.
2. The admin navigates to the manage membership section.
3. The system generates a list of all the memberships with view/edit/delete button appended to each one as well as a form for membership creation.
4. The admin selects which membership type to delete and press the delete button.
5. The system prompts for confirmation.
6. The admin confirms their action.

Outputs: Removed membership type from the database.

#### **3.1.4.22 Assign Instructor to a Course**

Introduction: Assigns an instructor to a course.

Inputs: Admin has access to the manage course section.

Processing:

1. The admin logs in to the app.
2. Admin navigates to the manage courses page.
3. The system generates a list of all the courses.
4. The admin selects a course.
5. The system shows the course information with the assigned instructor and an option to add/change/remove this instructor.
7. The admin press the add button to add an instructor.
8. The system will show a list of available instructor.
9. The admin selects the instructor he/she wishes to assign to the course.

Outputs: Course and Instructor information updated in the database.

#### **3.1.4.23 Remove Instructor from course**

Introduction: The Admin removes an instructor from the course

Inputs: Has access to the manage courses page.

Processing:

1. The admin logs in to the app.
2. Admin navigates to the manage courses page.
3. The system generates a list of all the courses.
4. The admin selects a course.
5. The system shows the course information with the assigned instructor and an option to change/remove this instructor.
6. The admin submits the change.

Outputs: Course details updated in the database as well as instructor details.

#### **3.1.4.24 Enroll Members to a Course**

Introduction: Admin adds a member to a course.

Inputs: Admin has authority to add members to a course.

Processing:

1. Admin logs in to the app.
2. The admin navigates to the course section.
3. The system generates a list of all the courses.
4. The admin selects a course.
5. The system displays the course information, add member form, and lists of all members enrolled in the course with view/remove button appended to each one.
6. The admin fills in the form.
7. The system validates that the user is allowed to enroll in a course.

Outputs: Course information and course member lists, update database, notification sent to the user added from the course.

#### 3.1.4.25 Remove Member from Course

Introduction: Admin removes a member from a course

Inputs: Admin has authority to remove members from a course.

Processing:

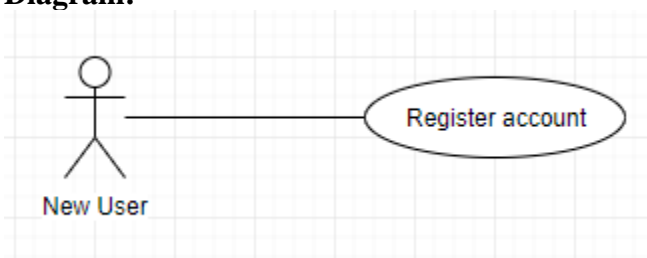
1. The Admin logs in to the app.
2. The Admin navigates to the course section.
3. The system generates a list of all the courses.
4. The admin selects a course.
5. The system shows the course information add member form, and list of all members enrolled with view/remove button appended to each one.
6. The Admin selects the member he/she wants to remove and press remove.
7. The system prompts for confirmation.
8. The admin confirms their action.

Outputs: Course information and course member lists, update database, notification sent to the user removed from the course.

## 10. 3.2 Use Cases

### a. 3.2.1 New User Use Case

**Diagram:**



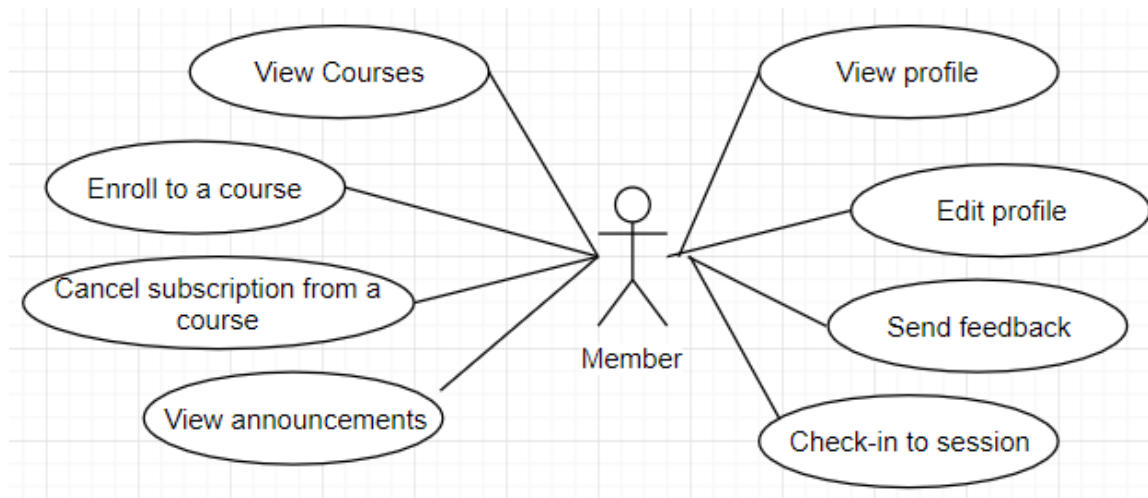
**3.2.1.1 Use Case: Register account**

Brief Description: A new user will be able to access the website, input user information and submit the form to register a new account to the system. (Note: A user that is successfully able to register an account is now a member of the system)

Detailed Description: A new user selects to create a new member account from the main homepage of the website. The user completes the registration form and submits. System verifies the form and creates the member account.

**b. 3.2.2 Member Use Case**

**Diagram:**

**3.2.2.1 Use Case: View Courses**

Brief Description: A member must be able to view all courses available with the appropriate course details.

Detailed Description: A member must be logged in to their account. The member chooses to view all courses available to join in.

**3.2.2.2 Use Case: Enroll to a Course**

Brief Description: A member must be able to enroll in a course available from the course list.

Detailed Description: A member chooses a course from the course list. The member selects a course and the system enrolls the member to the course.

**3.2.2.3 Use Case: Cancel subscription from a course**

Brief Description: A member must be able to cancel subscription from a course.

Detailed Description: A member must be enrolled in a course. When member views a list of their course subscriptions, they can choose to cancel any subscription to any course they are enrolled in.

#### **3.2.2.4 Use Case: View announcements**

Brief Description: A member must be able to view announcements sent by instructors or admin.

Detailed Description: A member must be logged in to view announcements from their dashboard.

#### **3.2.2.5 Use Case: View Profile**

Brief Description: A member must be able to view their member profile.

Detailed Description: A member must be logged in to their account. The member chooses to view their profile from the member account page.

#### **3.2.2.6 Use Case: Edit Profile**

Brief Description: A member must be able to edit their member profile.

Detailed Description: A member chooses to edit their profile after viewing their profile in the member account page.

#### **3.2.2.7 Use Case: Check-in to session**

Brief Description: A member must be able to check in to a session that an instructor has started, so that they will be recorded in the attendance.

Detailed Description: A member must be logged in to their account. The member selects to view their member QR code. The member uses to QR code to be scanned by the scanner. The system accepts the member id and checks the user in the session, marking their attendance.

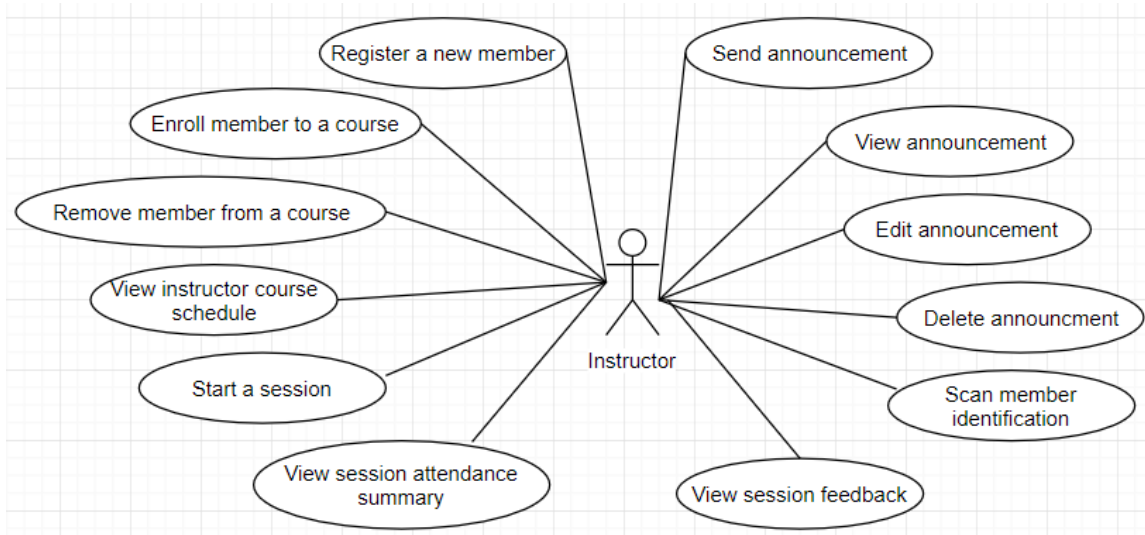
#### **3.2.2.8 Use Case: Send feedback**

Brief Description: A member must be able to send feedback after each session has concluded.

Detailed Description: A member must be logged in to their account. The member must also attend a session. After the session, the user chooses to send feedback about the session. The system accepts and stores the feedback.

### **c. 3.2.3 Instructor Use Case**

**Diagram:**



### 3.2.3.1 Use Case: Register a new member

**Brief Description:** An instructor must be able to register a new member to the system.

**Detailed Description:** An instructor must be logged in to the system. The instructor chooses to register a new member to the system. After inputting the details, the instructor submits and the system creates the new member account.

### 3.2.3.2 Use Case: Enroll member to a course

**Brief Description:** An instructor must be able to enroll in an existing member to a course.

**Detailed Description:** An instructor chooses a course from the available course list. The instructor then chooses to enroll a student by selecting a member to enroll in the course selected.

### 3.2.3.3 Use Case: Remove member from a course

**Brief Description:** An instructor must be able to remove a member from a course.

**Detailed Description:** An instructor chooses a course from the available course list. The instructor then chooses to remove a student by selecting a member to remove from the course selected.

### 3.2.3.4 Use Case: View instructor course schedule

**Brief Description:** An instructor must be able to view the course schedule relating to their account.

**Detailed Description:** An instructor must be logged in to the system. The instructor chooses to view the schedule for the courses they are assigned to for the week.

### 3.2.3.5 Use Case: Start session

**Brief Description:** An instructor must be able to start a new session.

**Detailed Description:** An instructor must be logged in to the system. The instructor chooses to start the session for the day.



**3.2.3.6 Use Case: Scan/Check-in member identification**

Brief Description: An instructor must be able to scan a member identification code or input a member identification code to a session.

Detailed Description: An instructor chooses to start a session. Then, the instructor may check in a student to the session attendance, either by scanning their QR ID code or by manually entering their information in the system.

**3.2.3.7 Use Case: View session feedback**

Brief Description: An instructor must be able to view session feedback from members who attended the session.

Detailed Description: An instructor chooses to start a session and end a session. The members who attended the session has the option to submit a feedback for the concluded session. The instructor chooses to view session feedback by selecting a completed session and viewing feedback for the session.

**3.2.3.8 Use Case: Send announcement**

Brief Description: An instructor must be able to send an announcement.

Detailed Description: An instructor chooses from a list of their courses. The instructor chooses to send an announcement, adds announcement details and submits the announcement. The members registered to a course receives the announcement.

**3.2.3.9 Use Case: View announcement**

Brief Description: An instructor must be able to view an announcement.

Detailed Description: An instructor chooses from a list of their courses. The instructor chooses to view all announcements for the course.

**3.2.3.10 Use Case: Edit announcement**

Brief Description: An instructor must be able to edit an announcement.

Detailed Description: An instructor chooses from a list of their courses. The instructor chooses an announcement to edit. The instructor edits announcement details and submits the announcement. The members registered to a course receives the announcement.

**3.2.3.11 Use Case: Delete announcement**

Brief Description: An instructor must be able to delete an announcement.

Detailed Description: An instructor chooses from a list of their courses. The instructor chooses to view all announcements. The instructor deletes the announcement.

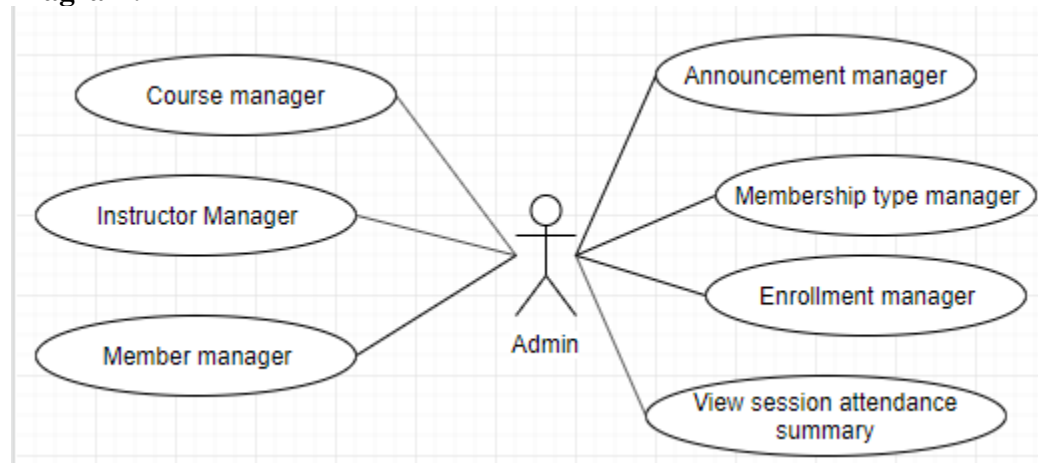
**3.2.3.12 Use Case: View session attendance summary**

Brief Description: An instructor must be able to view session attendance summary..

Detailed Description: An instructor chooses from a list of completed sessions. The instructor chooses a session and a list of members who attended will be displayed.

### d. 3.2.4 Admin Use Case

Diagram:

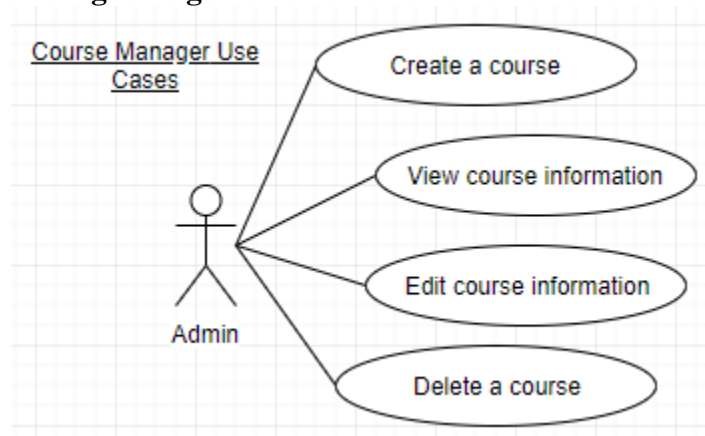


#### 3.2.4.1 Use Case: View Session attendance summary

**Brief Description:** Admin must be able to view attendance summary for every session

**Detailed Description:** An Admin chooses from a list of completed sessions. The instructor chooses a session and a list of members who attended will be displayed.

Manager diagrams:



#### 3.2.4.2 Use Case: Create a course

**Brief Description:** Admin must be able to create a course.

**Detailed Description:** An admin chooses to create a new course. The admin enters course details and submits. The system creates the course.

#### 3.2.4.3 Use Case: View course information

**Brief Description:** Admin must be able to view course information.

**Detailed Description:** An admin chooses to view a list of courses. The admin selects a course to view course information.

#### 3.2.4.4 Use Case: Edit course information

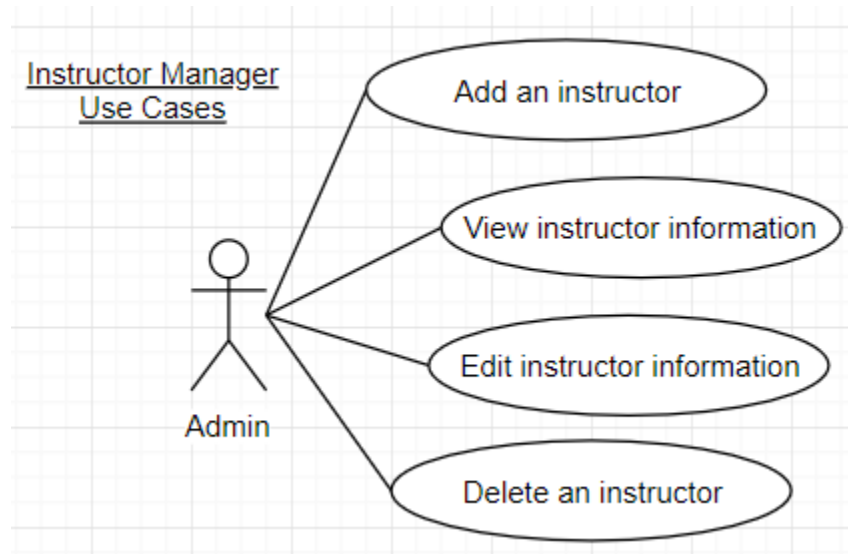
**Brief Description:** Admin must be able to edit course information.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course. Admin edits course details and submits the changes to the system.

#### **3.2.4.5 Use Case: Delete a course**

Brief Description: Admin must be able to delete a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course and chooses to delete the course.



#### **3.2.4.6 Use Case: Add an instructor**

Brief Description: Admin must be able to add an instructor.

Detailed Description: An admin chooses to add a new instructor. The admin enters instructor details and submits. The system creates and adds the instructor.

#### **3.2.4.7 Use Case: View instructor information**

Brief Description: Admin must be able to view instructor information.

Detailed Description: An admin chooses to view a list of instructors. The admin selects an instructor to view information.

#### **3.2.4.8 Use Case: Edit instructor information**

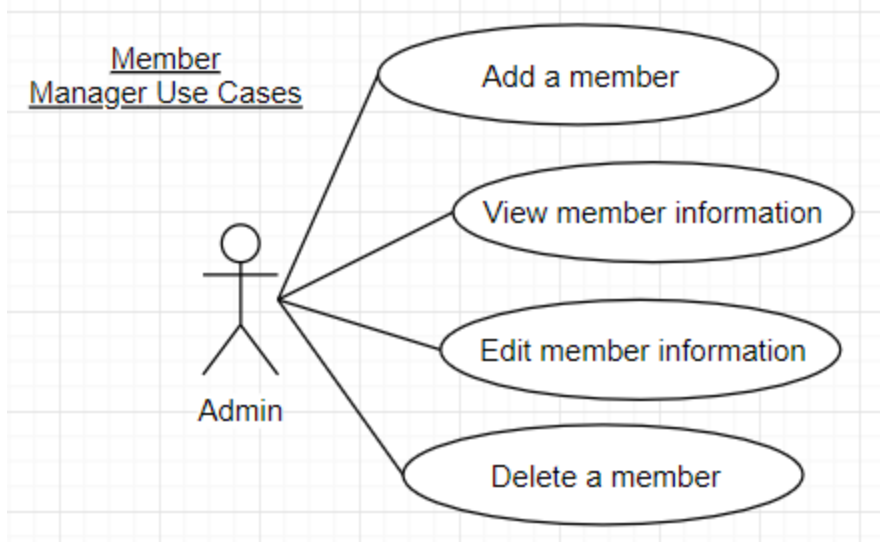
Brief Description: Admin must be able to edit course information.

Detailed Description: An admin chooses to view a list of courses. The admin selects an announcement. Admin edits course details and submits the changes to the system.

#### **3.2.4.9 Use Case: Delete an instructor**

Brief Description: Admin must be able to delete an instructor.

Detailed Description: An admin chooses to view a list of instructors. The admin selects an instructor and chooses to delete the instructor.

**3.2.4.10 Use Case: Add a member**

Brief Description: Admin must be able to add a member.

Detailed Description: An admin chooses to add a new member. The admin enters member details and submits. The system creates and adds the member.

**3.2.4.11 Use Case: View member information**

Brief Description: Admin must be able to view member information.

Detailed Description: An admin chooses to view a list of members. The admin selects a member to view information.

**3.2.4.12 Use Case: Edit member information**

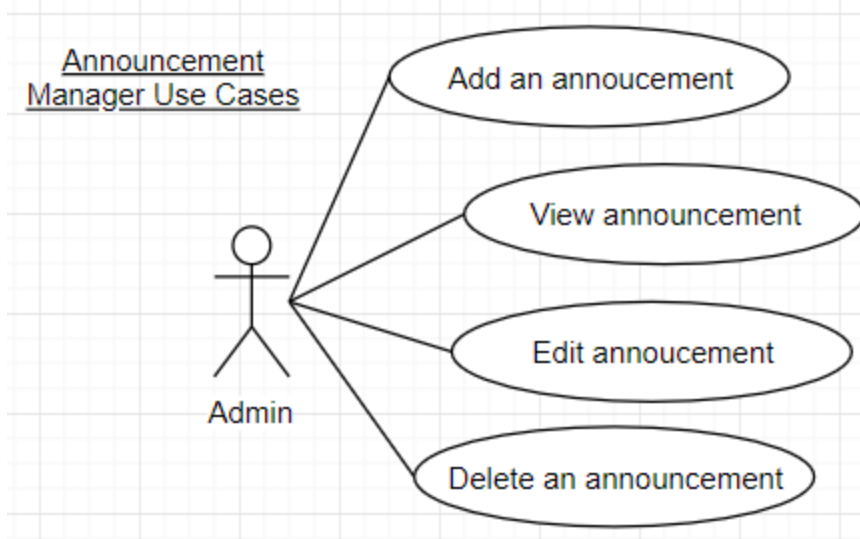
Brief Description: Admin must be able to edit member information.

Detailed Description: An admin chooses to view a list of members. The admin selects a member. Admin edits member details and submits the changes to the system.

**3.2.4.13 Use Case: Delete a member**

Brief Description: Admin must be able to delete a member.

Detailed Description: An admin chooses to view a list of members. The admin selects a member and chooses to delete the member.

**3.2.4.14 Use Case: Send an announcement**

Brief Description: Admin must be able to send an announcement.

Detailed Description: An admin chooses to send a new announcement. The admin enters announcement details and submits. The system creates and sends the announcement.

**3.2.4.15 Use Case: View announcements**

Brief Description: Admin must be able to view announcements.

Detailed Description: An admin chooses to view a list of announcements.

**3.2.4.16 Use Case: Edit announcement**

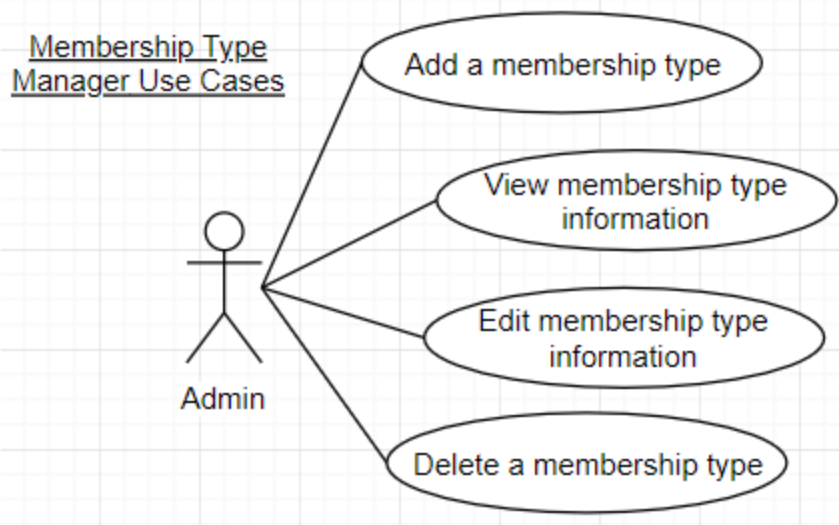
Brief Description: Admin must be able to edit announcements.

Detailed Description: An admin chooses to view a list of announcements. The admin selects an announcement. Admin edits announcement details and submits the changes to the system.

**3.2.4.17 Use Case: Delete an announcement**

Brief Description: Admin must be able to delete an announcement.

Detailed Description: An admin chooses to view a list of instructors. The admin selects an announcement and chooses to delete the announcement.

**3.2.4.18 Use Case: Add a membership type**

Brief Description: Admin must be able to add a membership type.

Detailed Description: An admin chooses to add a new course. The admin enters course details then proceeds to add membership type details. The system creates and adds the membership type details to the system. Another way: An admin selects a course from a list of courses. The admin chooses to add a new membership type for the selected course. Admin enters membership type details then submits changes to the system.

**3.2.4.19 Use Case: View membership type information**

Brief Description: Admin must be able to view membership type information from a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course to view information. The membership type information is displayed alongside course information.

**3.2.4.20 Use Case: Edit membership type information**

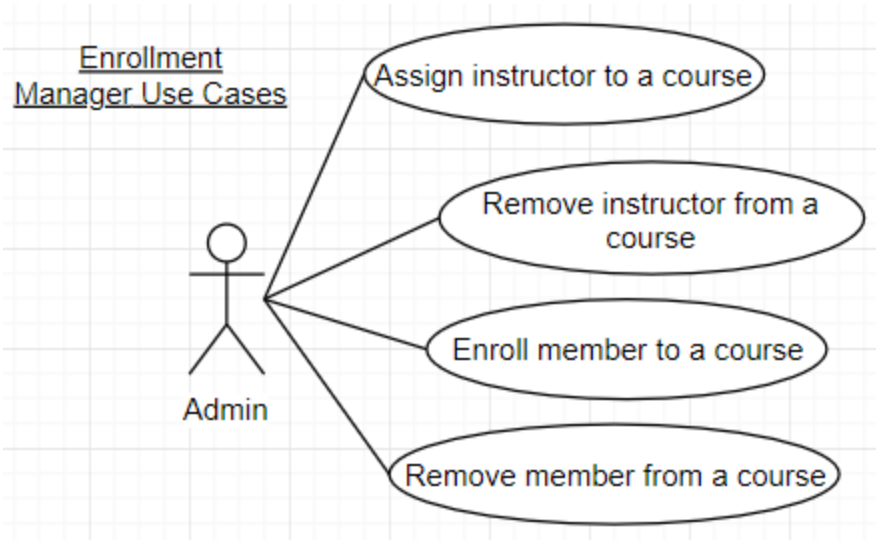
Brief Description: Admin must be able to edit membership type details from a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course. Admin chooses to edit current membership type details for the selected course. Admin submits the changes to the system.

**3.2.4.21 Use Case: Delete a membership type**

Brief Description: Admin must be able to delete a membership type for a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course. Admin can choose to delete the membership type if there is more than 1 membership type. Admin cannot delete a single membership type since all courses must have a membership type.

**3.2.4.22 Use Case: Assign instructor to a course**

Brief Description: Admin must be able to assign an instructor to a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course and chooses the option to assign an instructor. Existing instructors are presented, and can be selected to be assigned to a course.

**3.2.4.23 Use Case: Remove instructor from a course**

Brief Description: Admin must be able to remove an instructor from a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course. The admin chooses to remove an instructor from the course selected, and submits the changes.

**3.2.4.24 Use Case: Enrol member to a course**

Brief Description: Admin must be able to enrol a member to a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course and chooses the option to enroll members. Existing members are presented and can be selected to enrol the member to selected course.

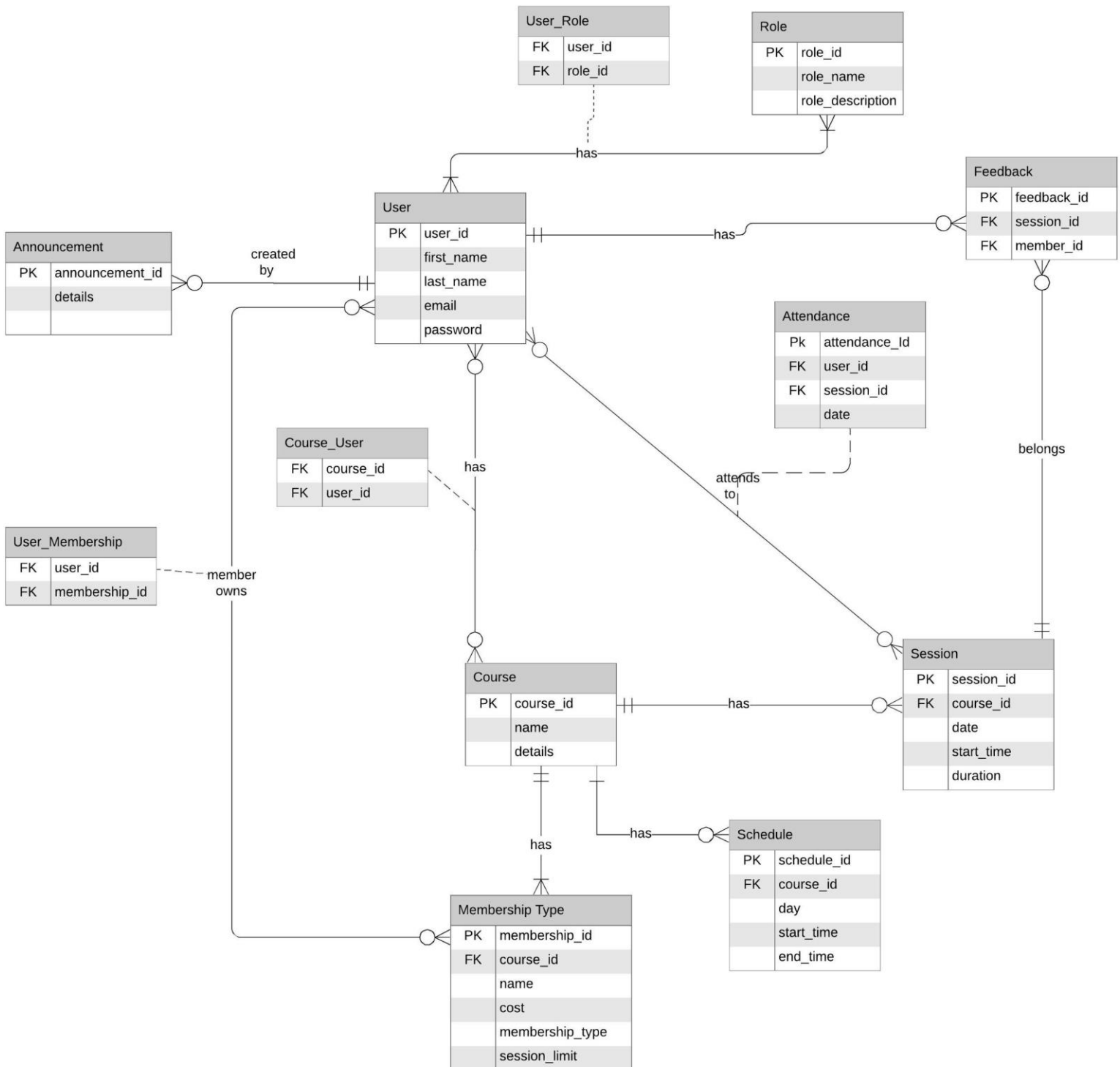
**3.2.4.25 Use Case: Remove member from a course**

Brief Description: Admin must be able to remove an active member from a course.

Detailed Description: An admin chooses to view a list of courses. The admin selects a course and chooses to view a list of enrolled members. The admin then selects members to remove from the course.

## 12. 3.3 Data Modelling and Analysis

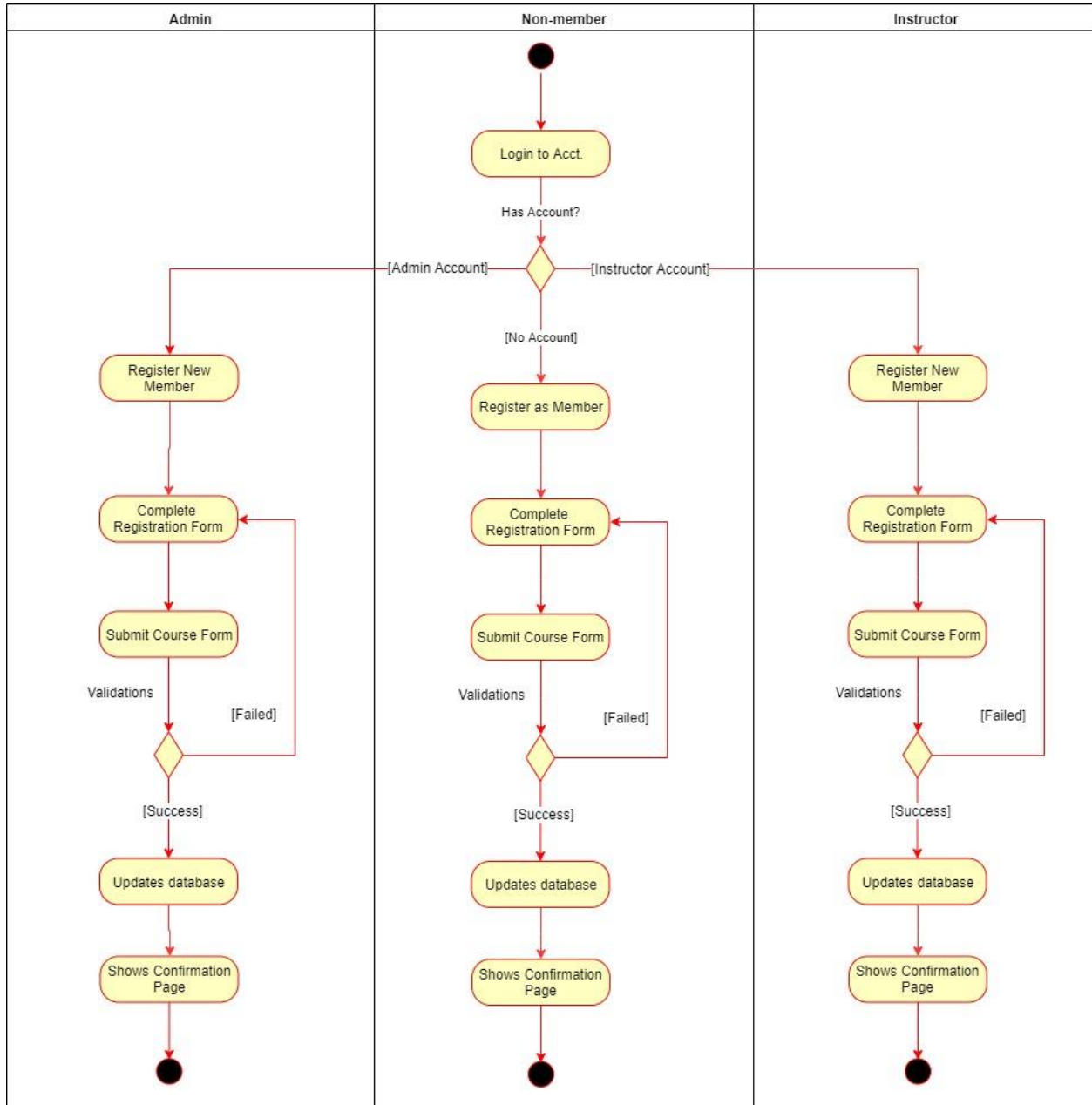
### a. 3.3.1 Normalized Data Model Diagram



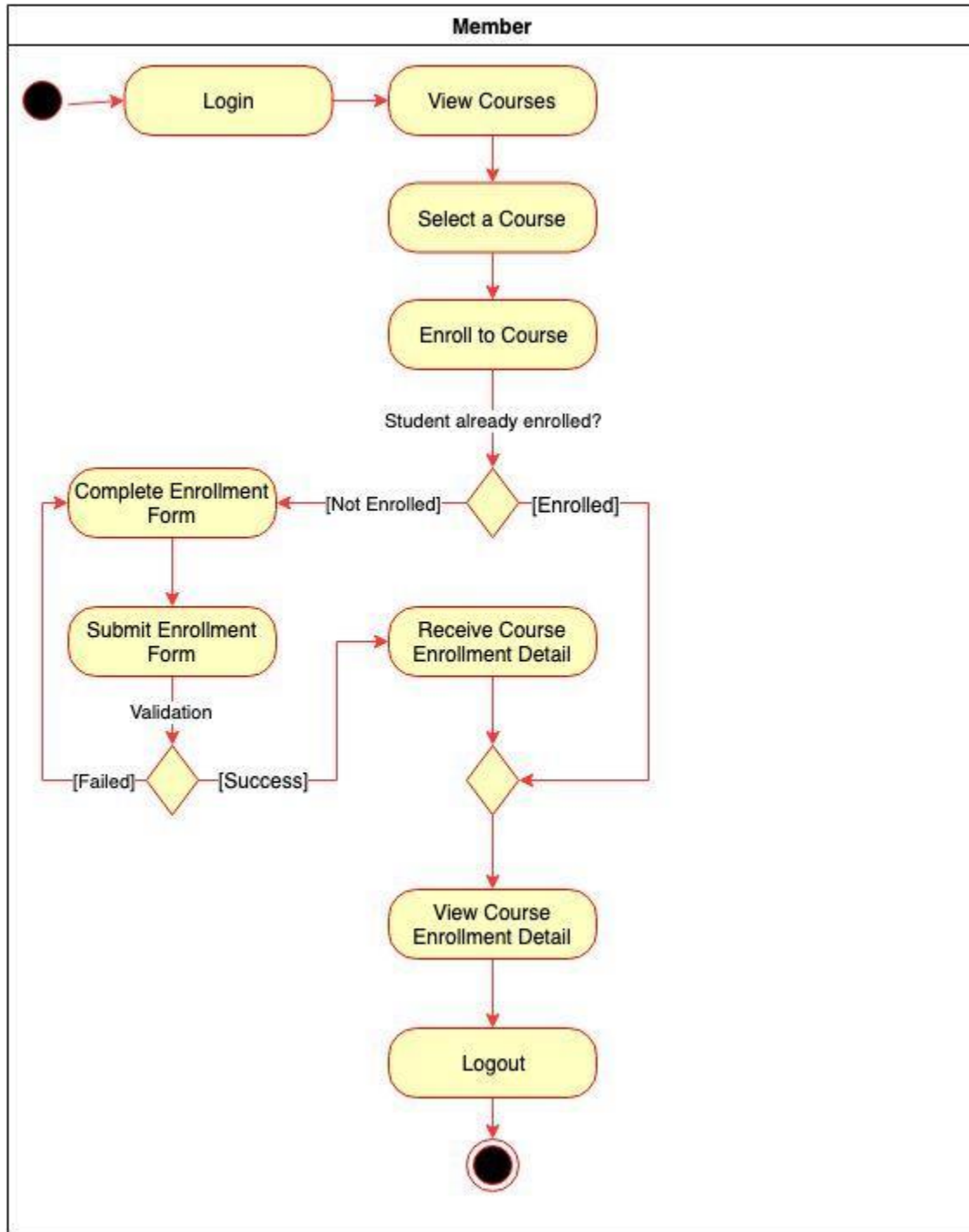
### b. 3.3.2 Activity Diagram



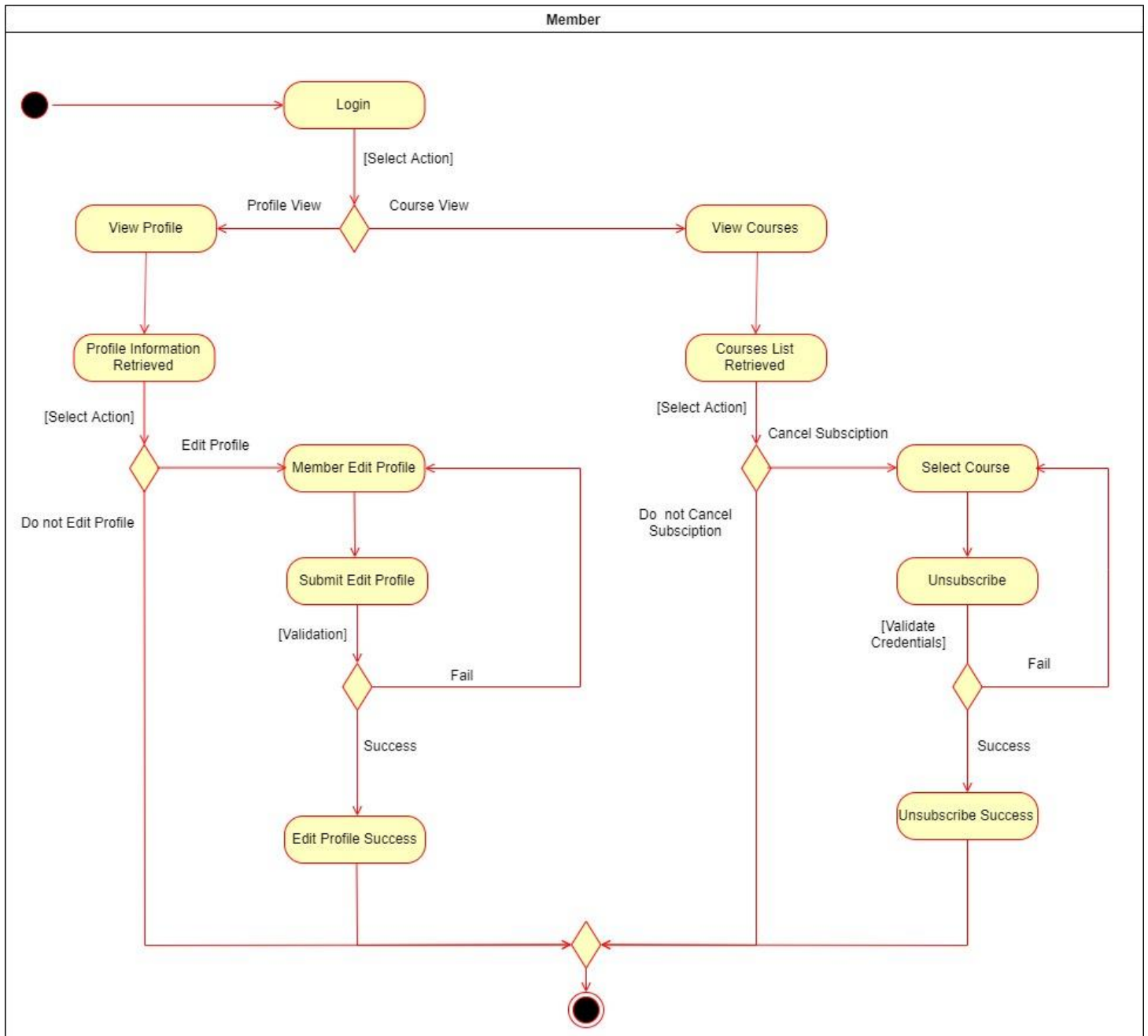
## 3.3.2.1 Register New User



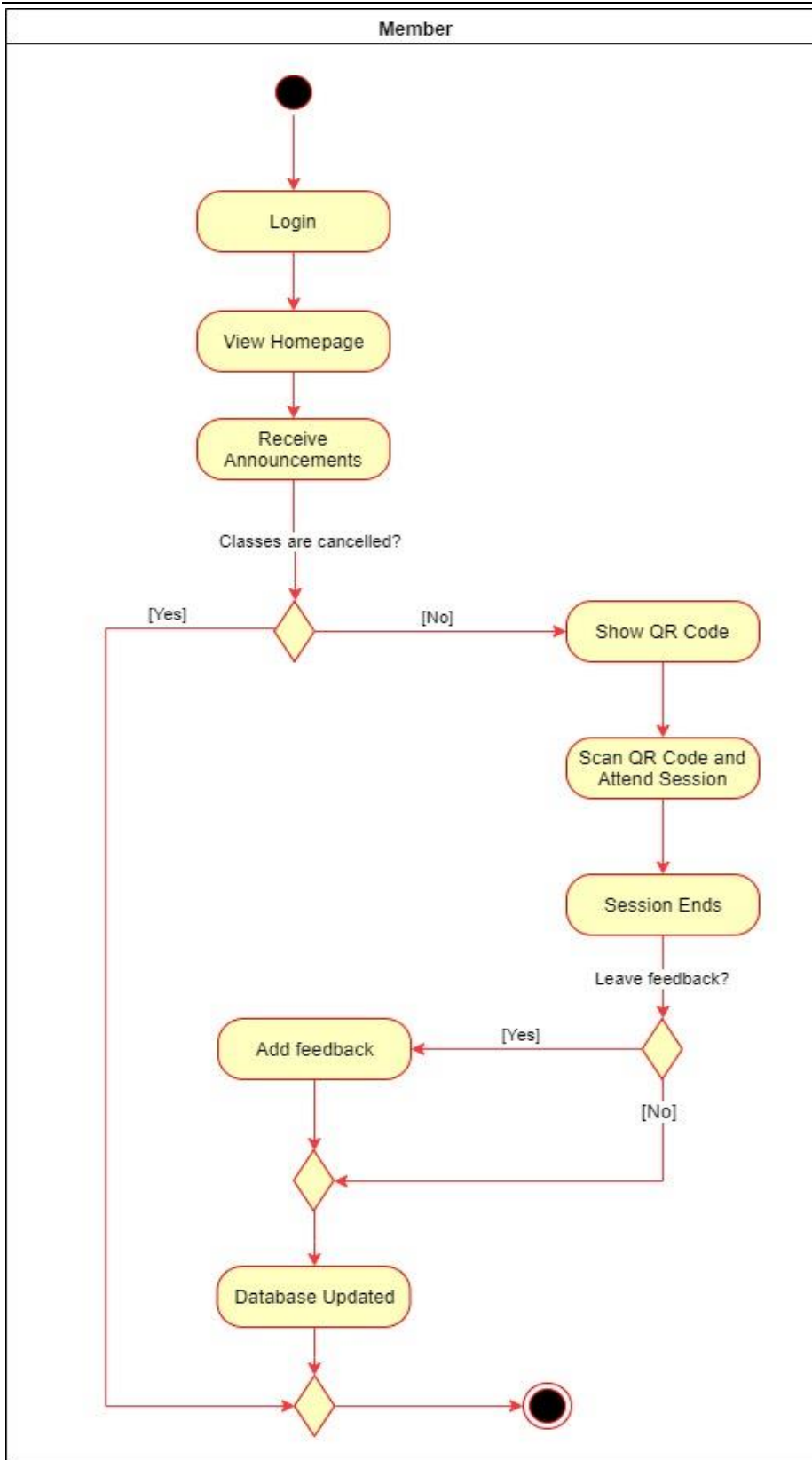
## 3.3.2.2 Member Course Enrollment



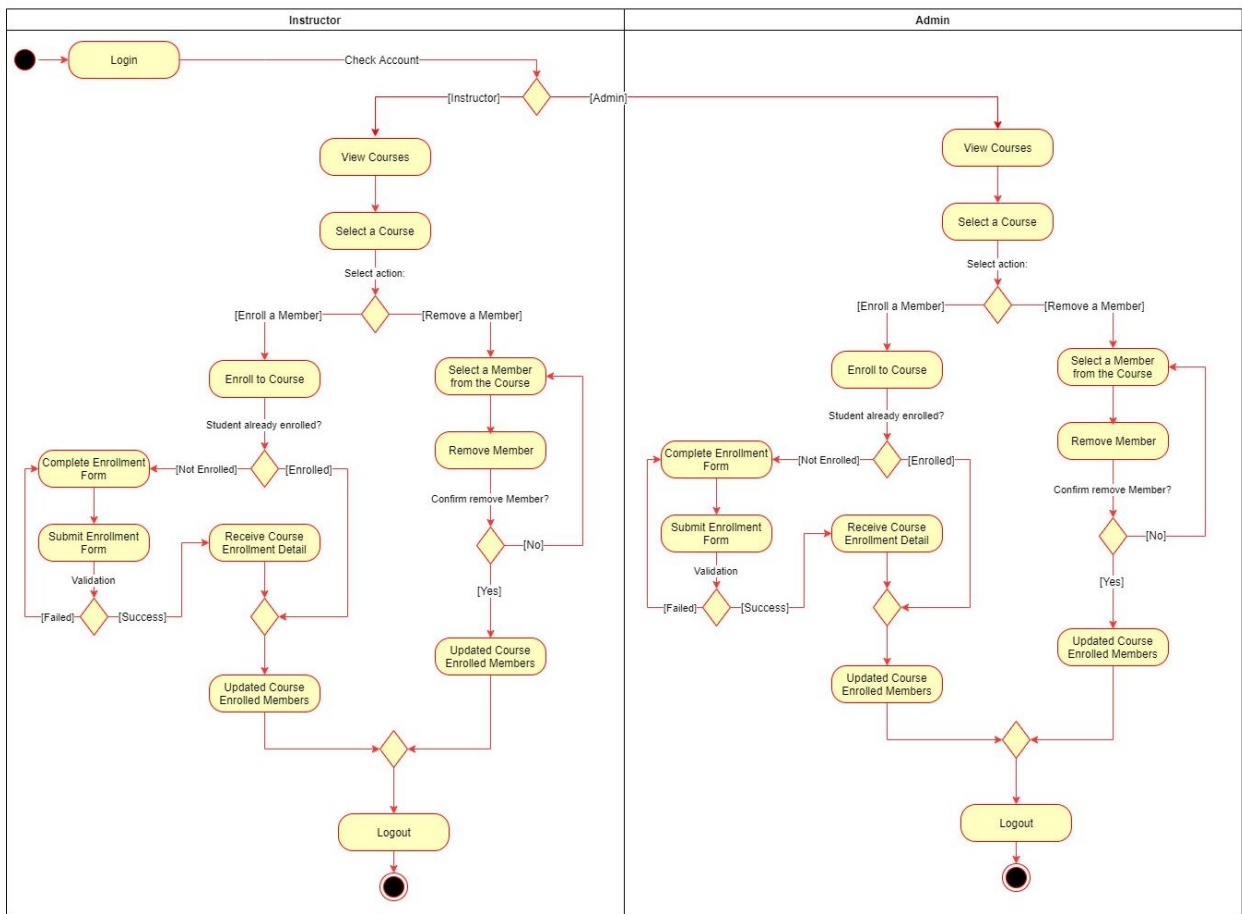
### 3.3.2.3 Managing Profile



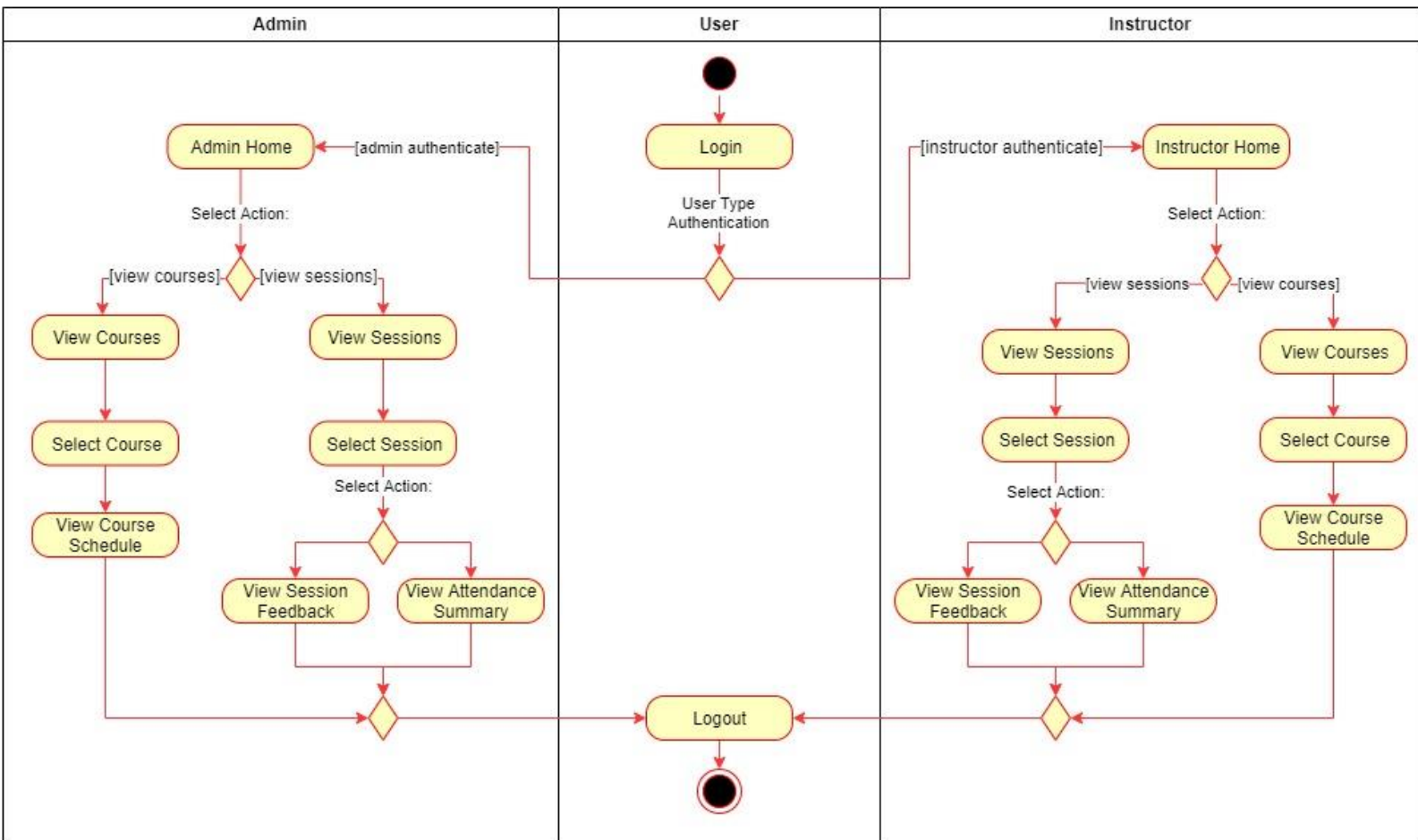
### 3.3.2.4 Member Attending a Session



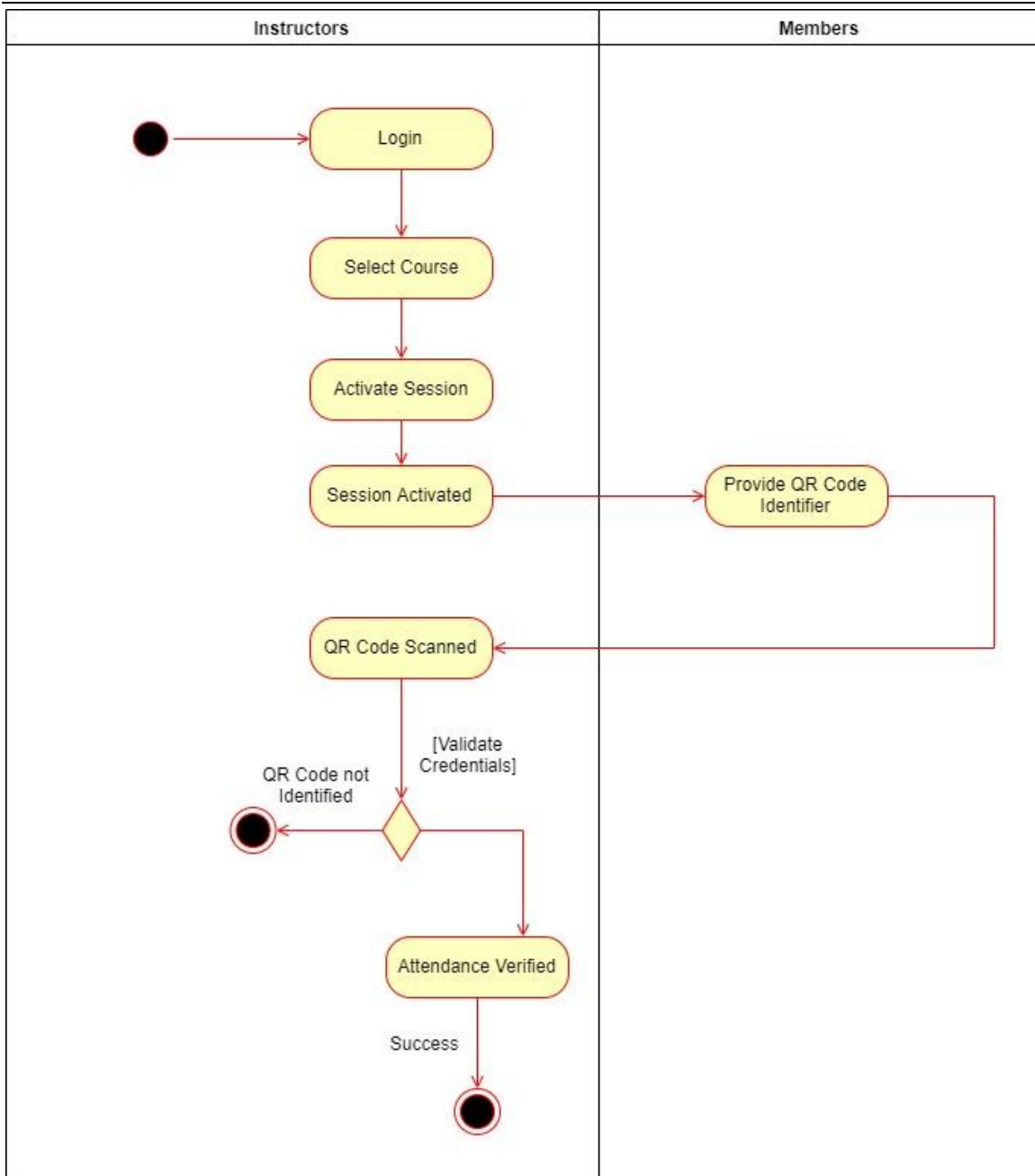
### 3.3.2.5 Managing Course Members



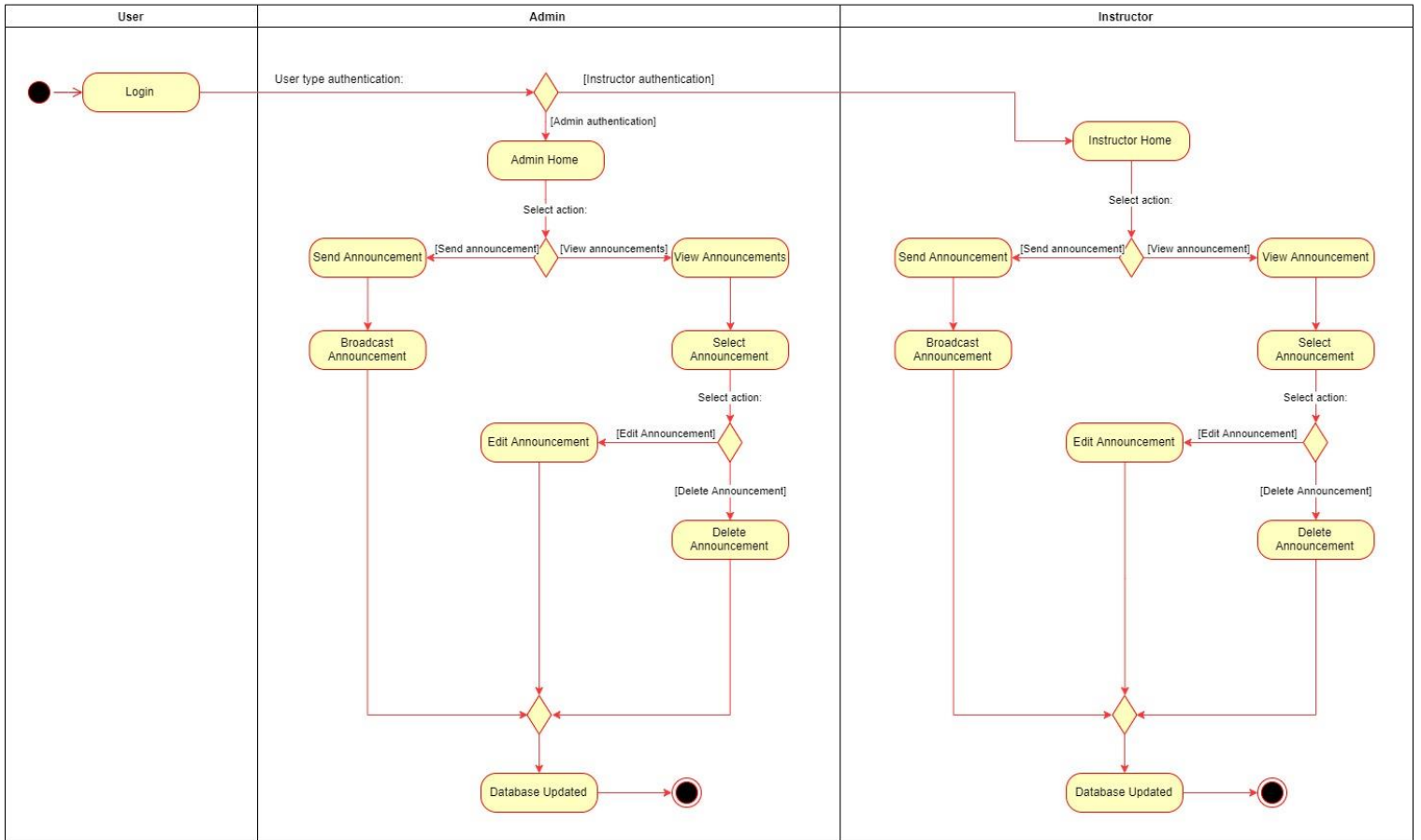
### 3.3.2.6 Viewing Course and Session Details



### 3.3.2.7 Qr Usage

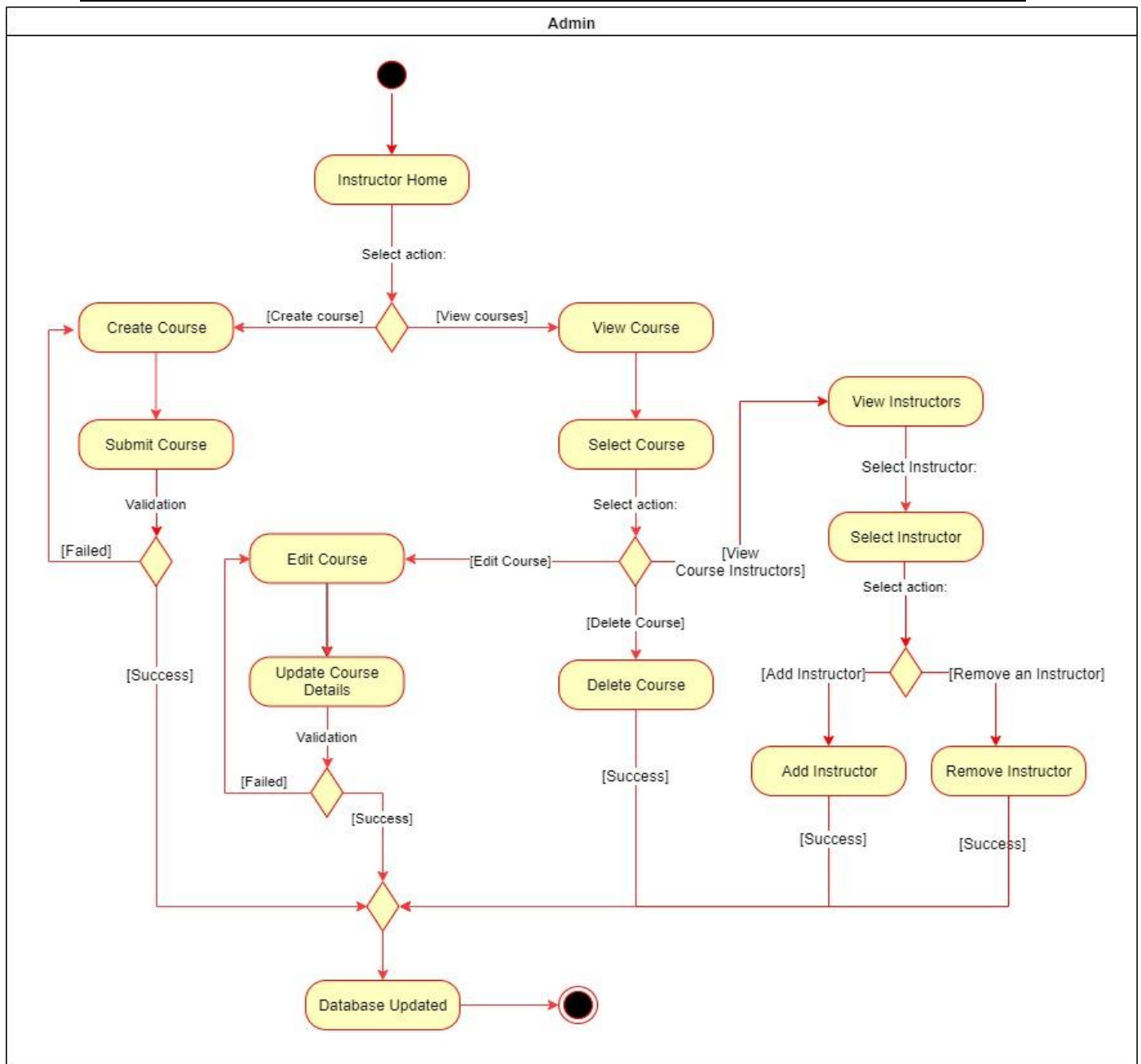


### 3.3.2.8 Managing Announcements



### 3.3.2.9 Managing Courses



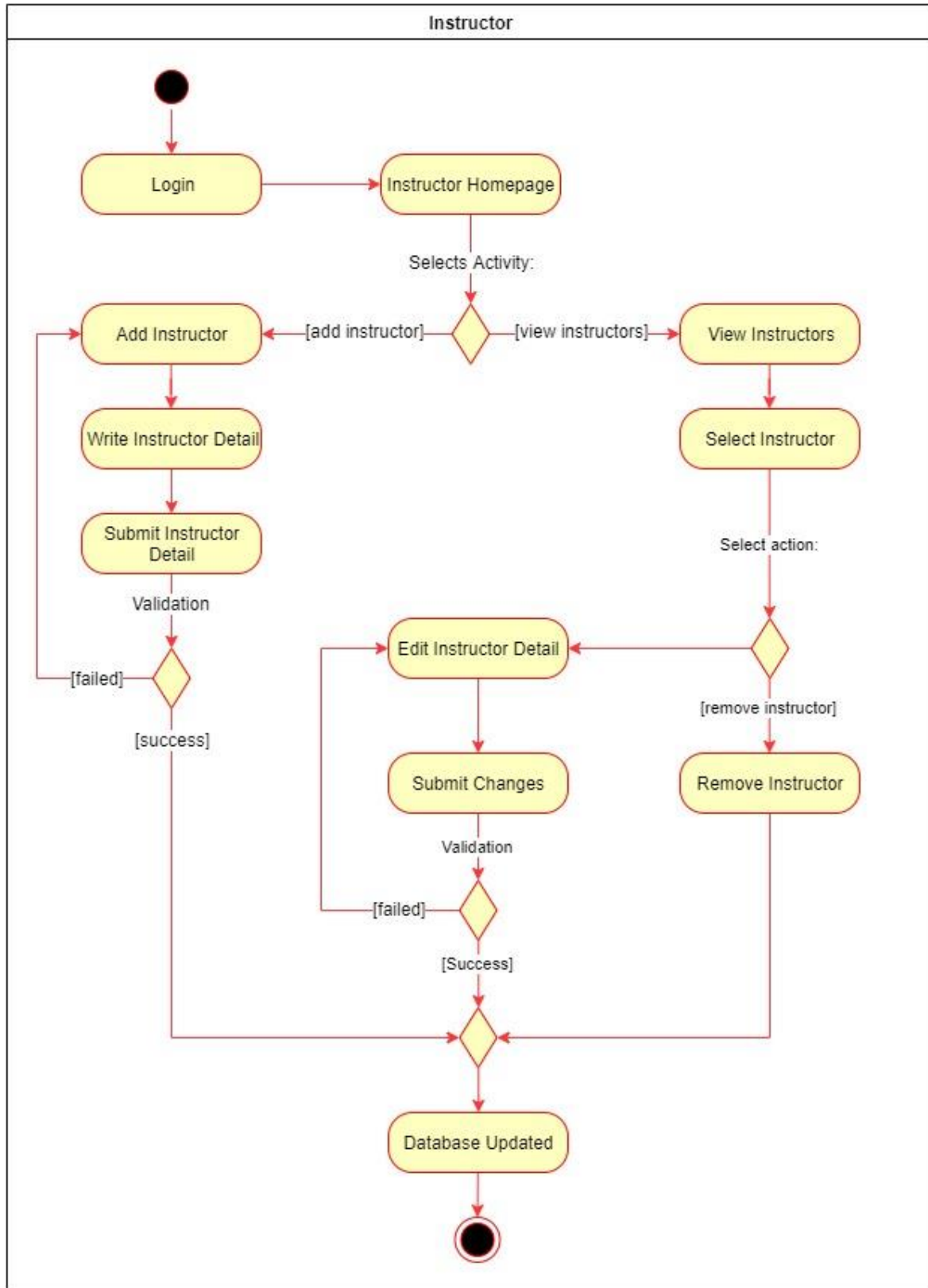


i.

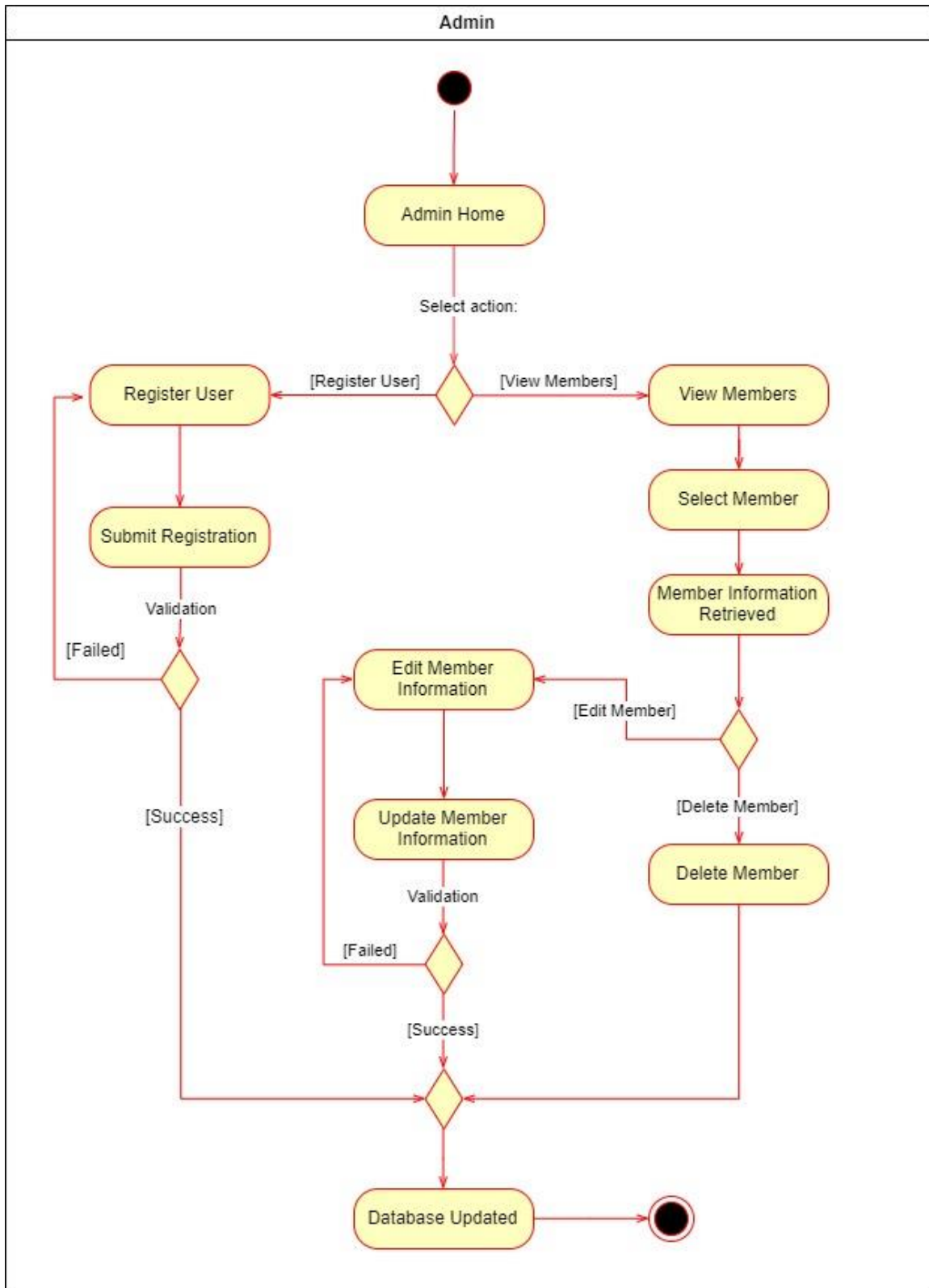
ii.

iii.

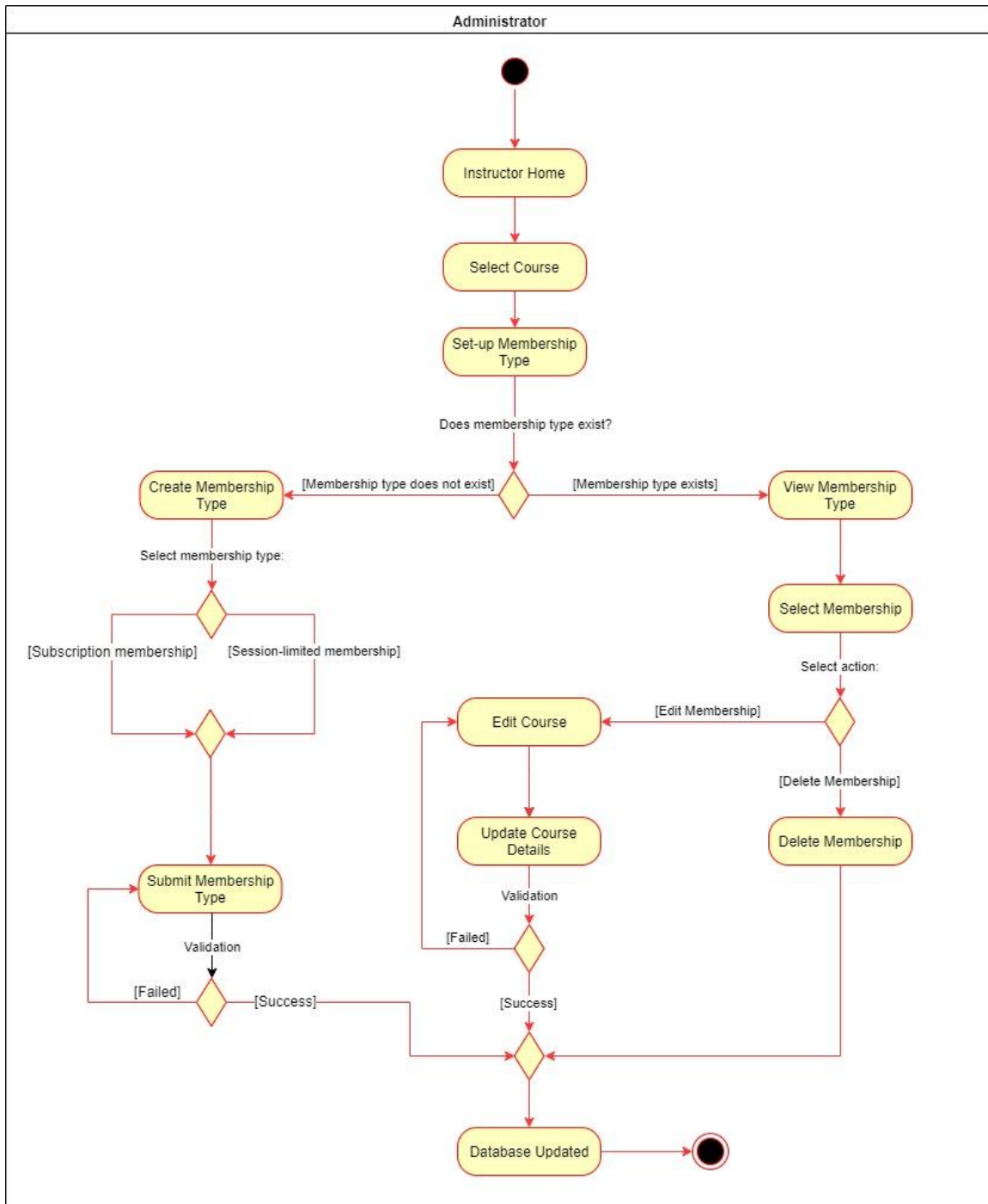
### 3.3.2.10 Managing Instructors



### 3.3.2.11 Managing member accounts



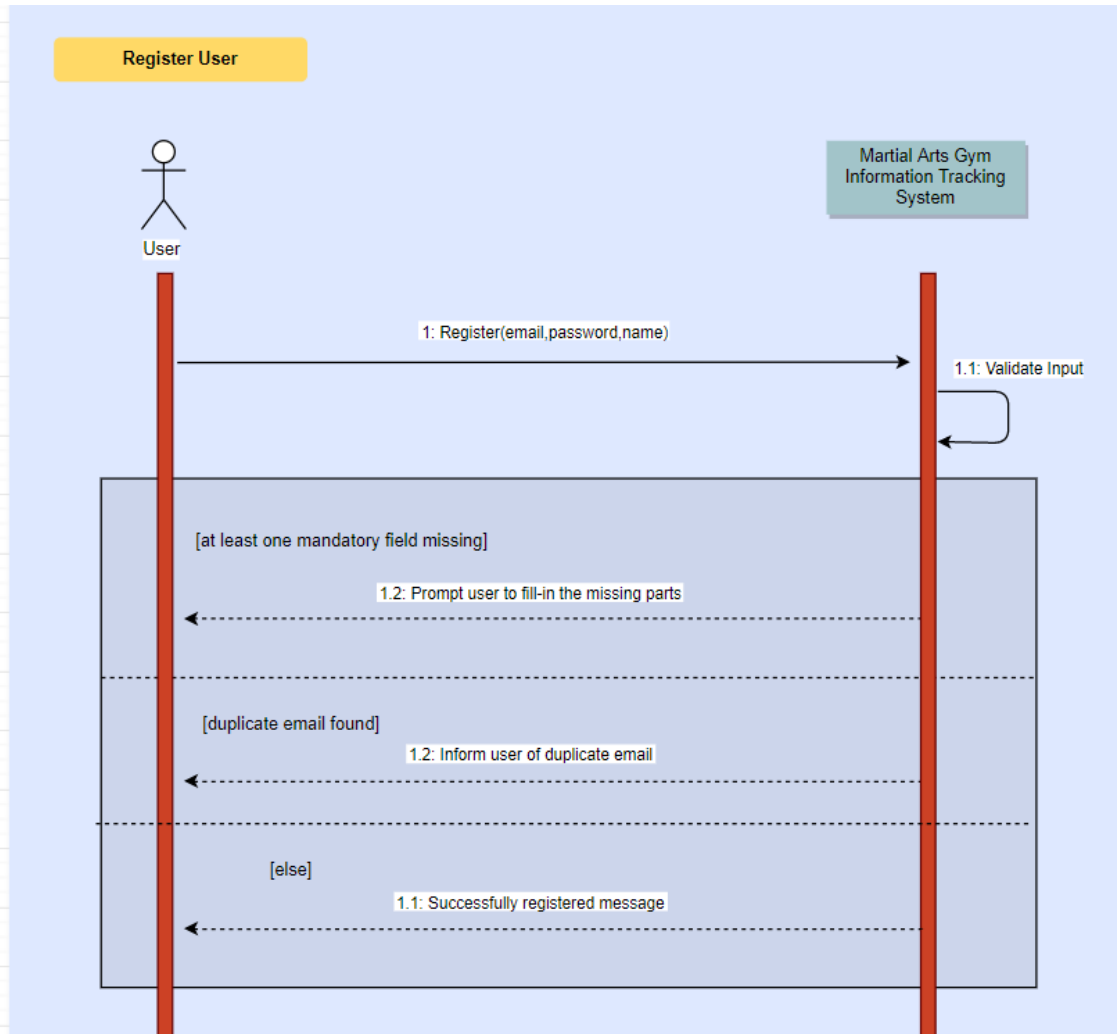
### 3.3.2.12 Managing membership



### c. 3.3.3 Sequence Diagrams

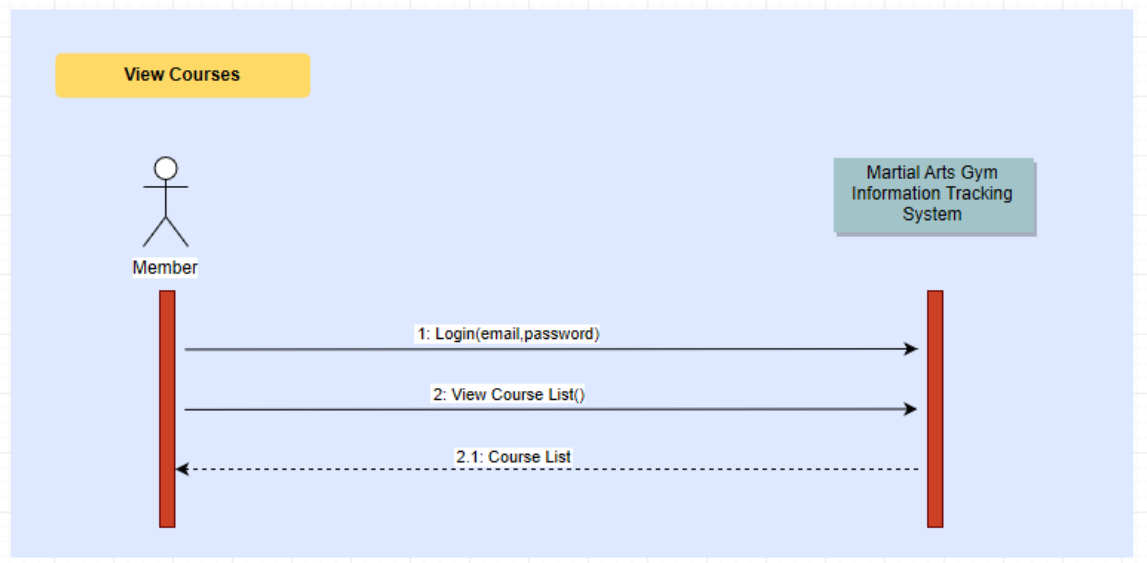
### 3.3.3.1 New User Sequence Diagram

**Register User** (refers to Use case 3.2.1.1)

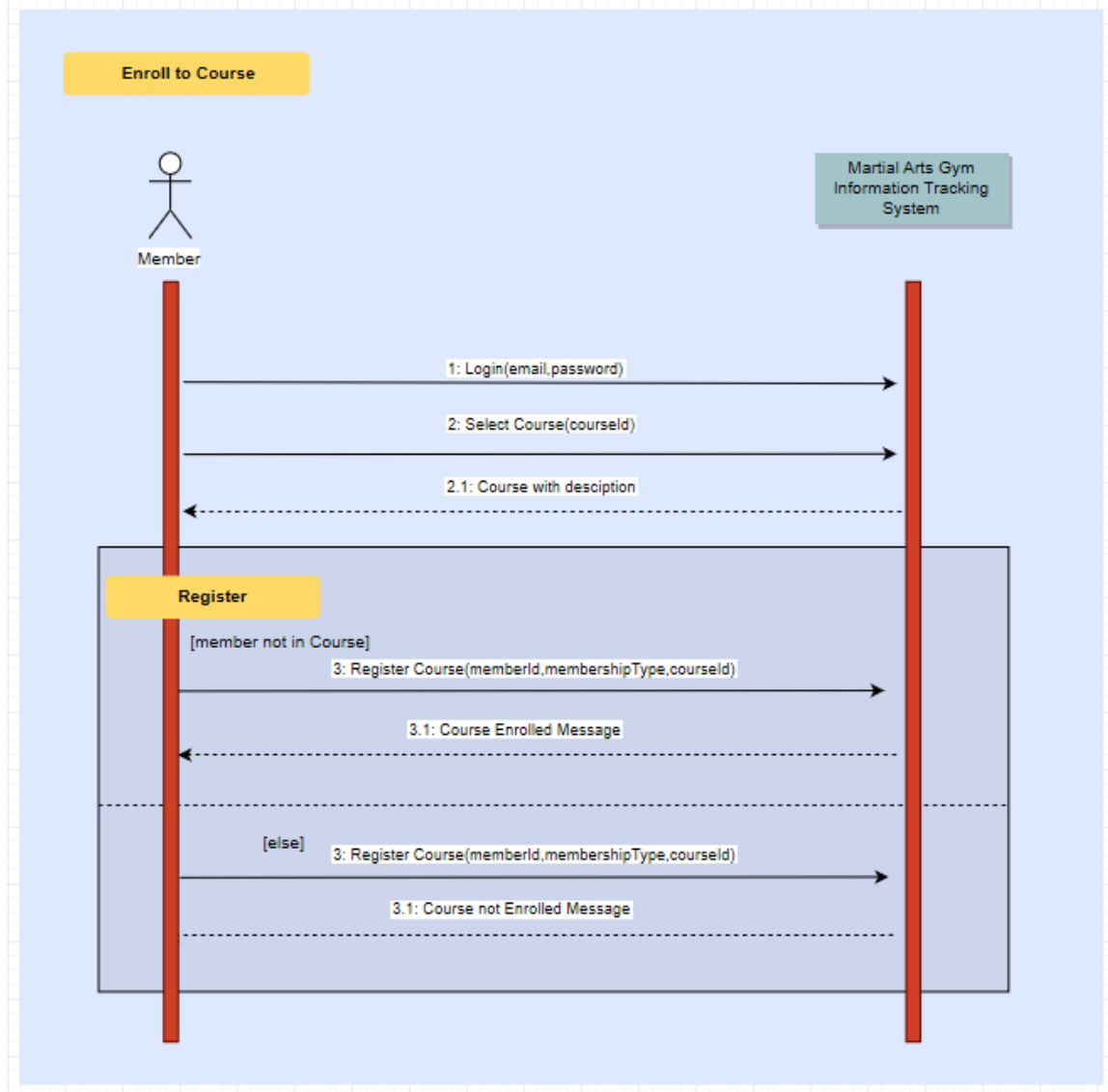


### 3.3.3.2 Member Sequence Diagram

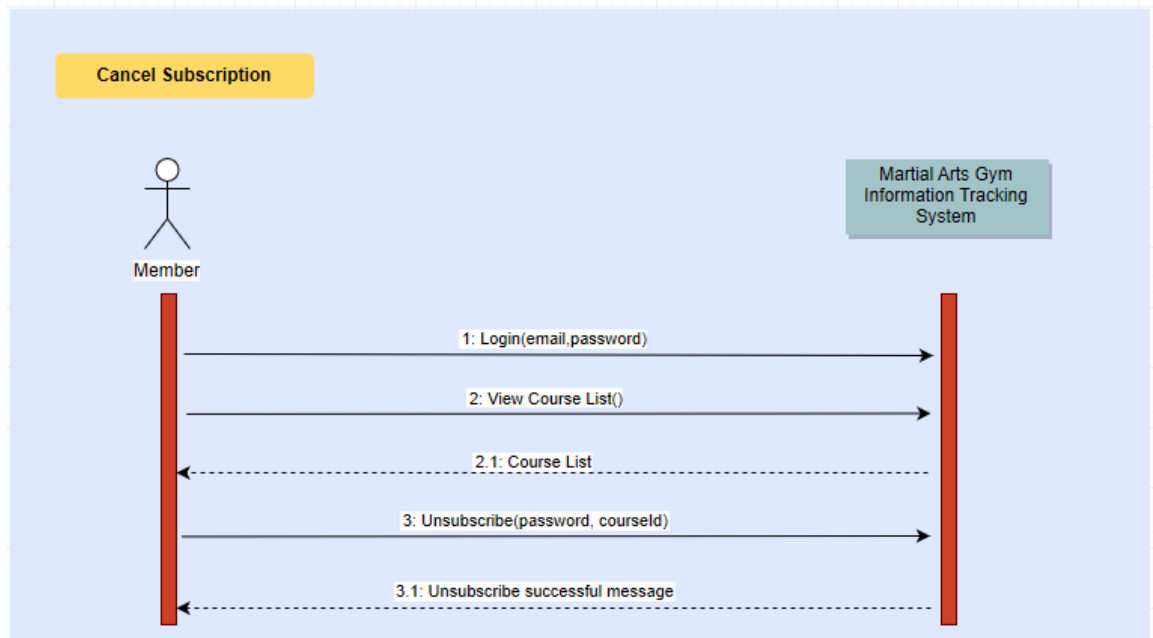
**View Courses** (refers to Use case 3.2.2.1)



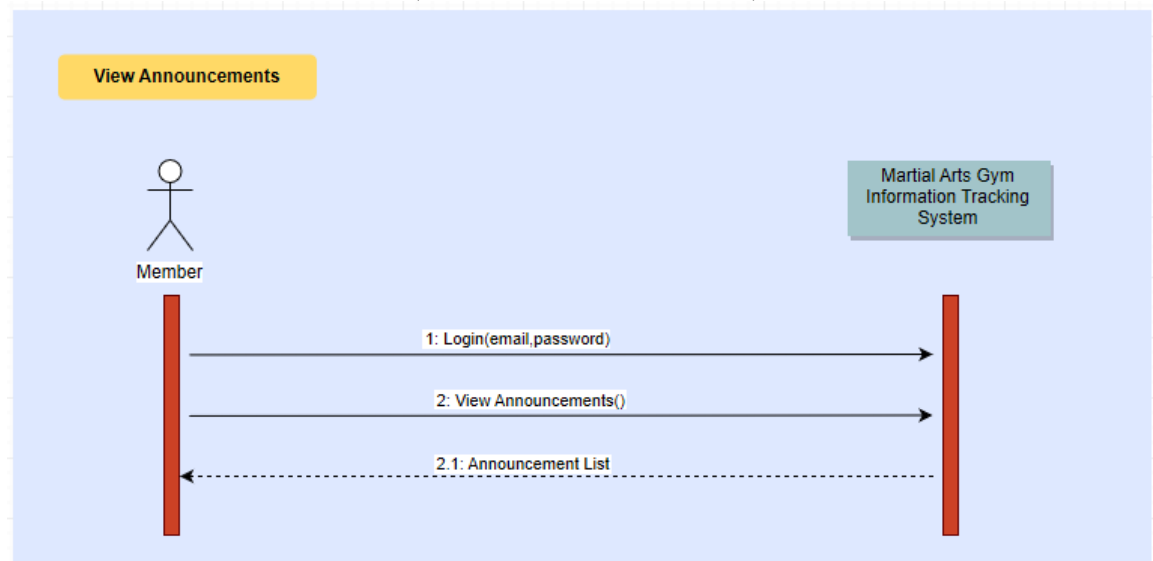
**Enroll to a Course** (refers to Use case 3.2.2.2)



**Cancel Subscription** (refers to Use case 3.2.2.3)

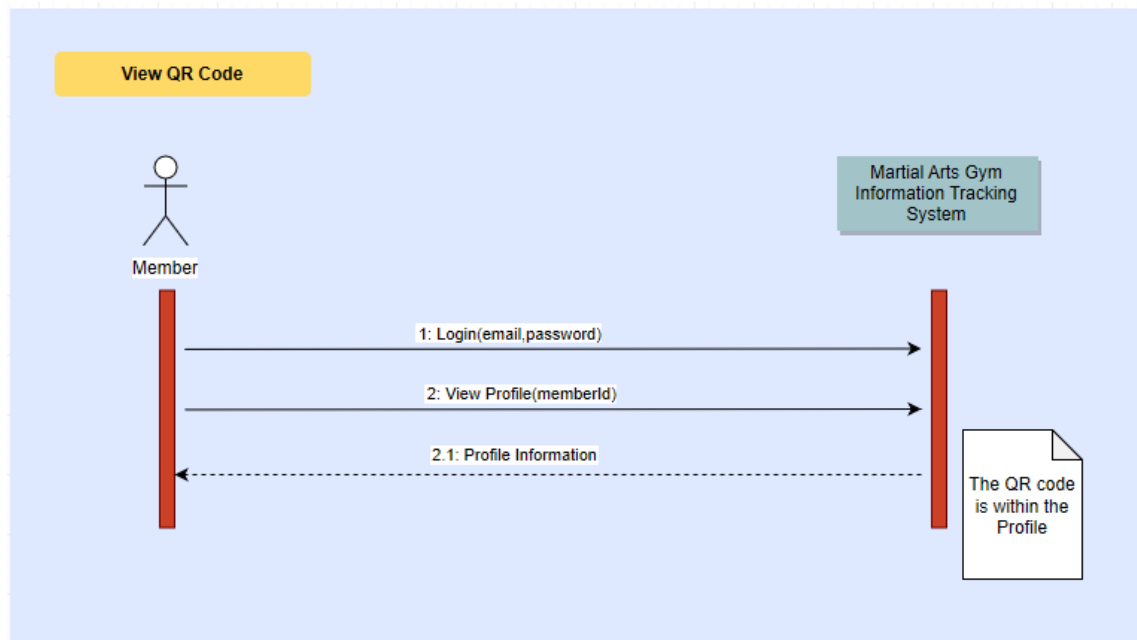


### View Announcements (refers to Use case 3.2.2.4)

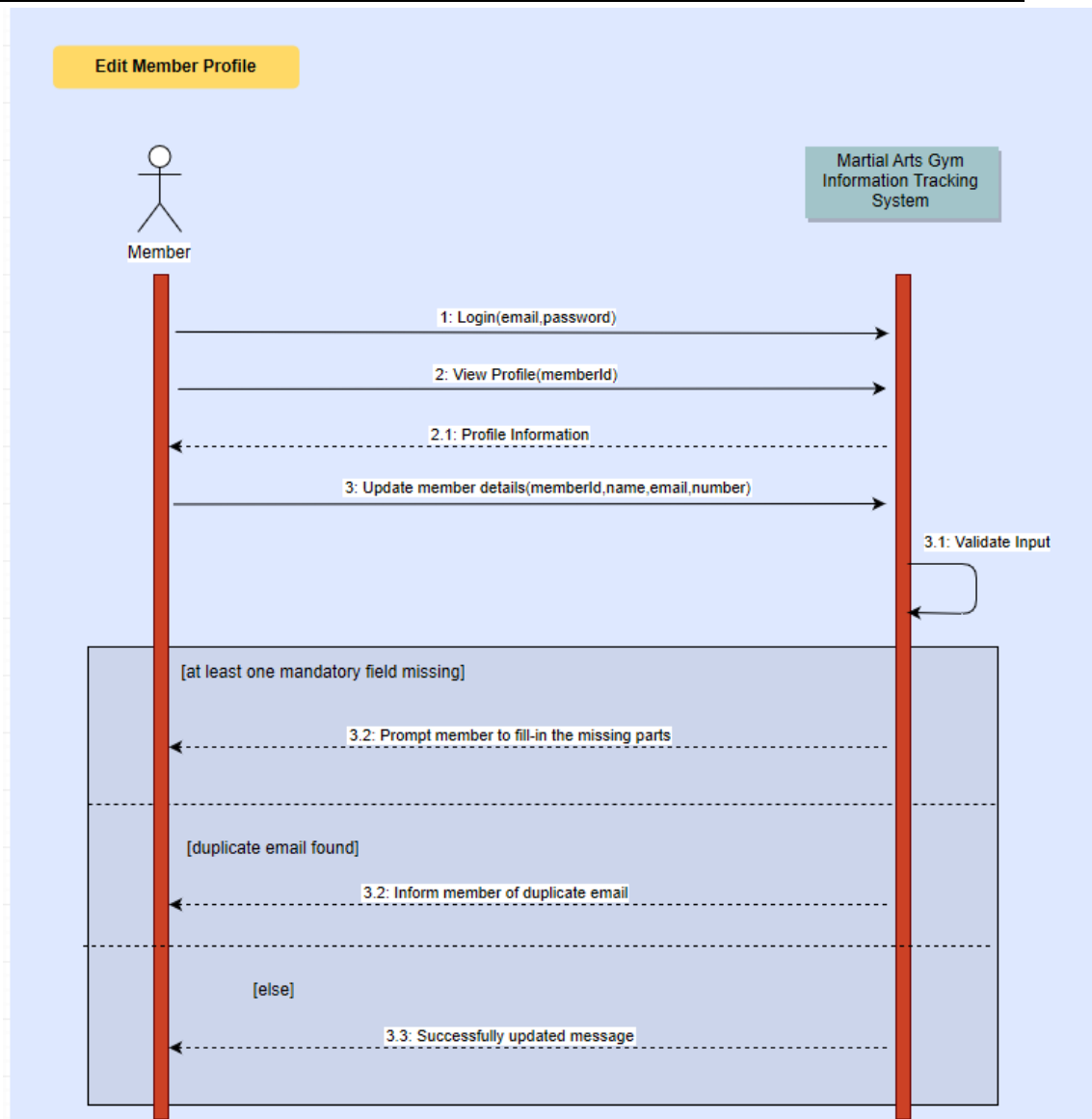


### View QR code (refers to Use case 3.2.2.5)

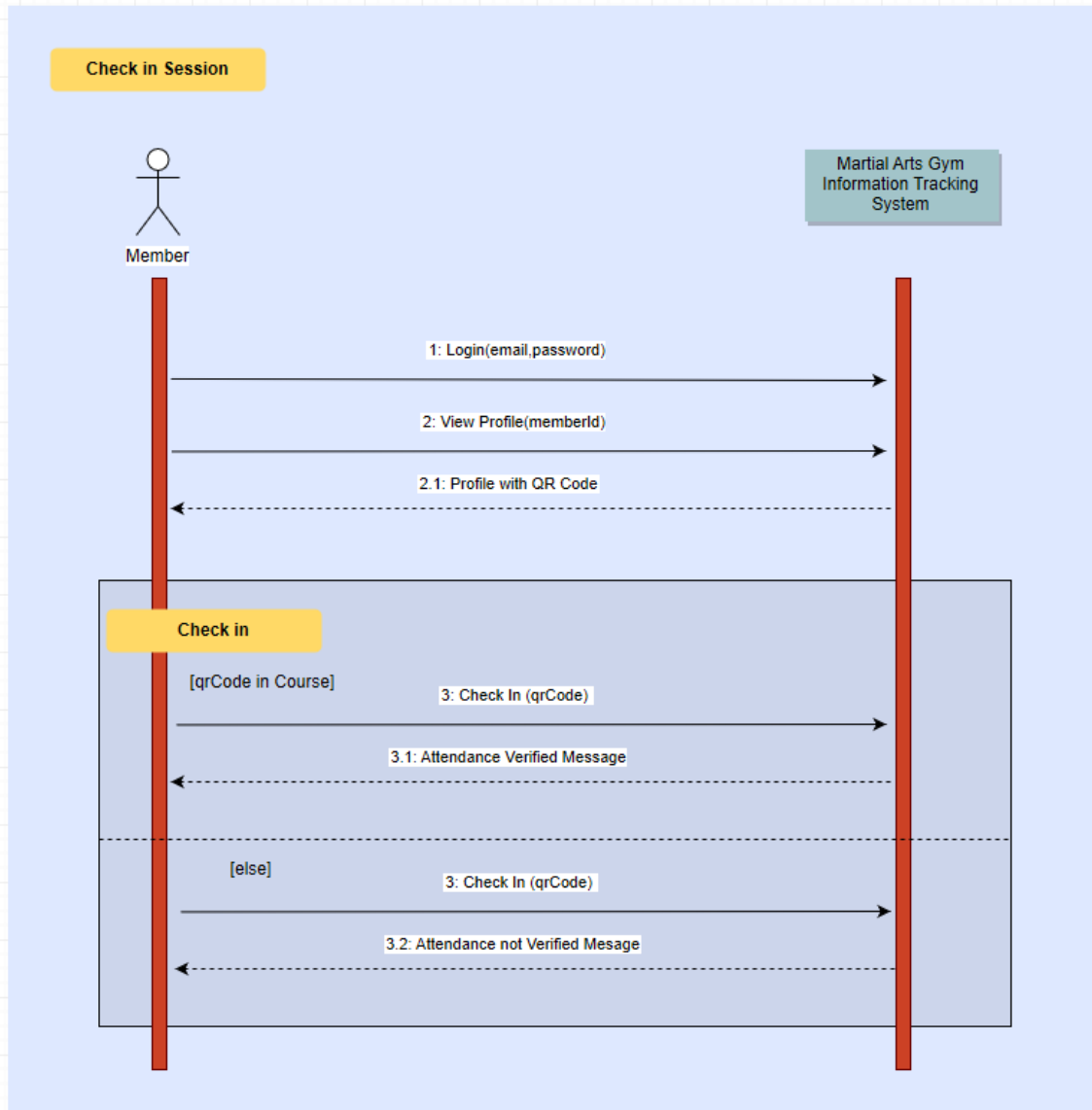




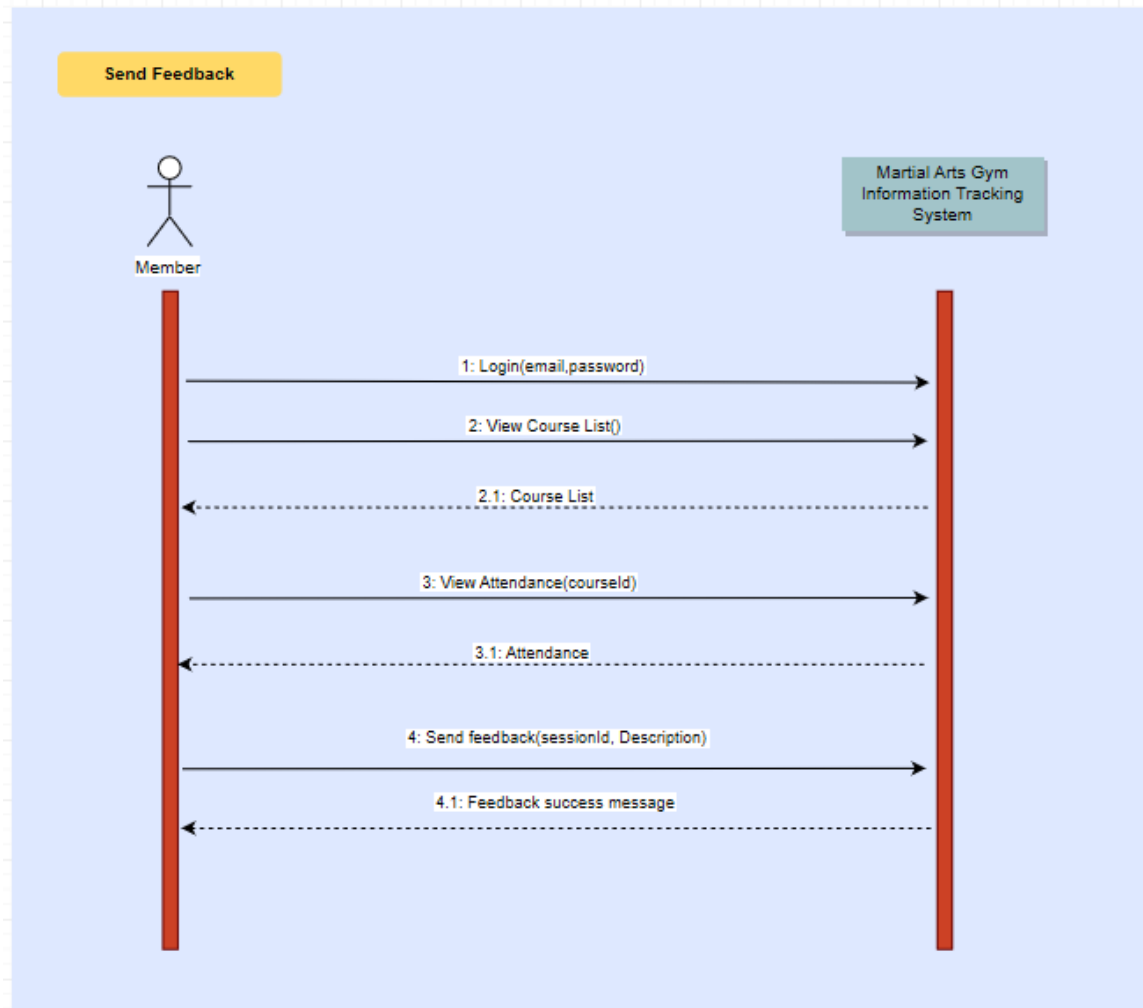
**Edit Member Profile** (refers to Use case 3.2.2.6)



**Check-in to session** (refers to Use case 3.2.2.7)

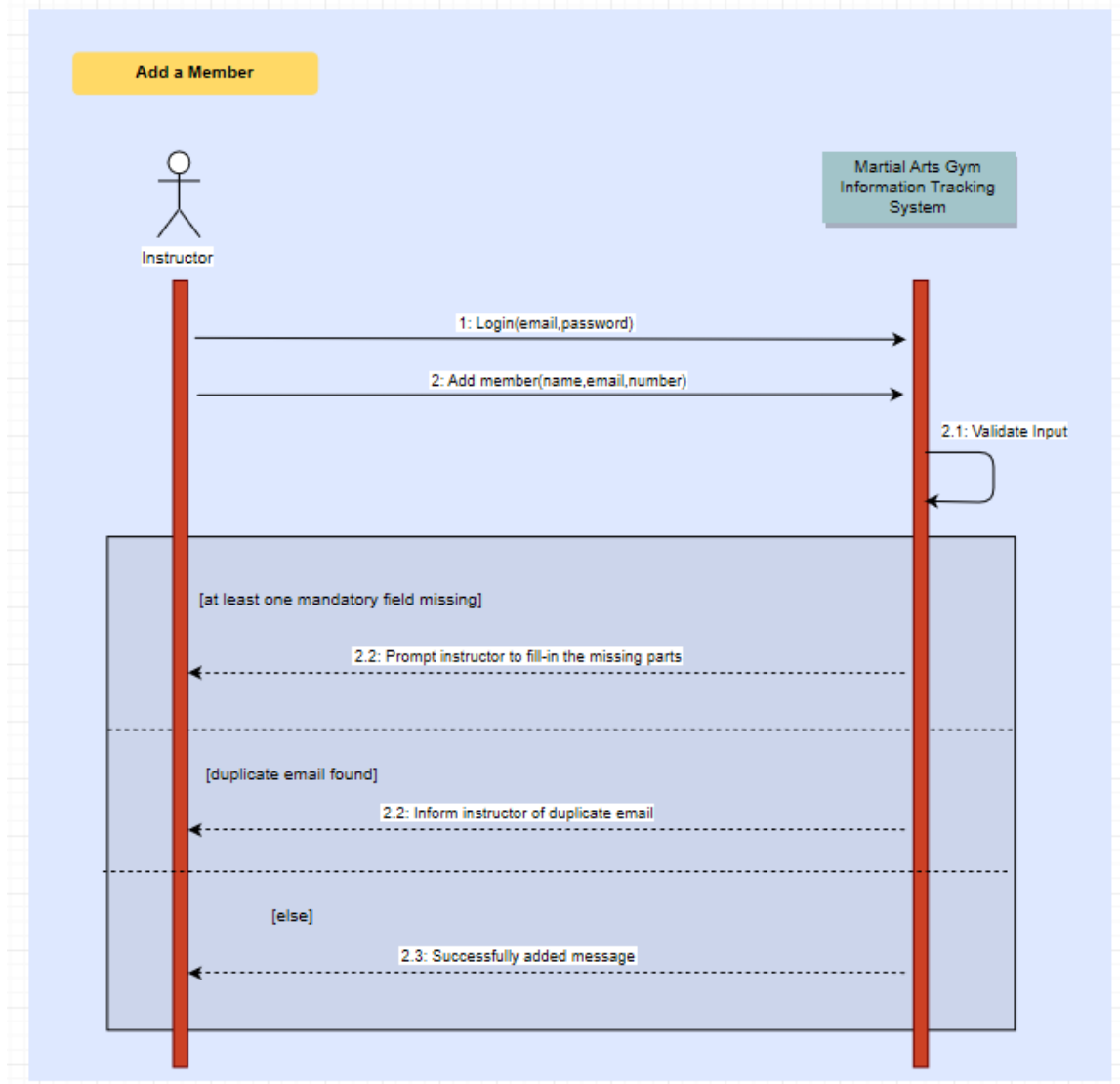


**Send Feedback** (refers to Use case 3.2.2.8)

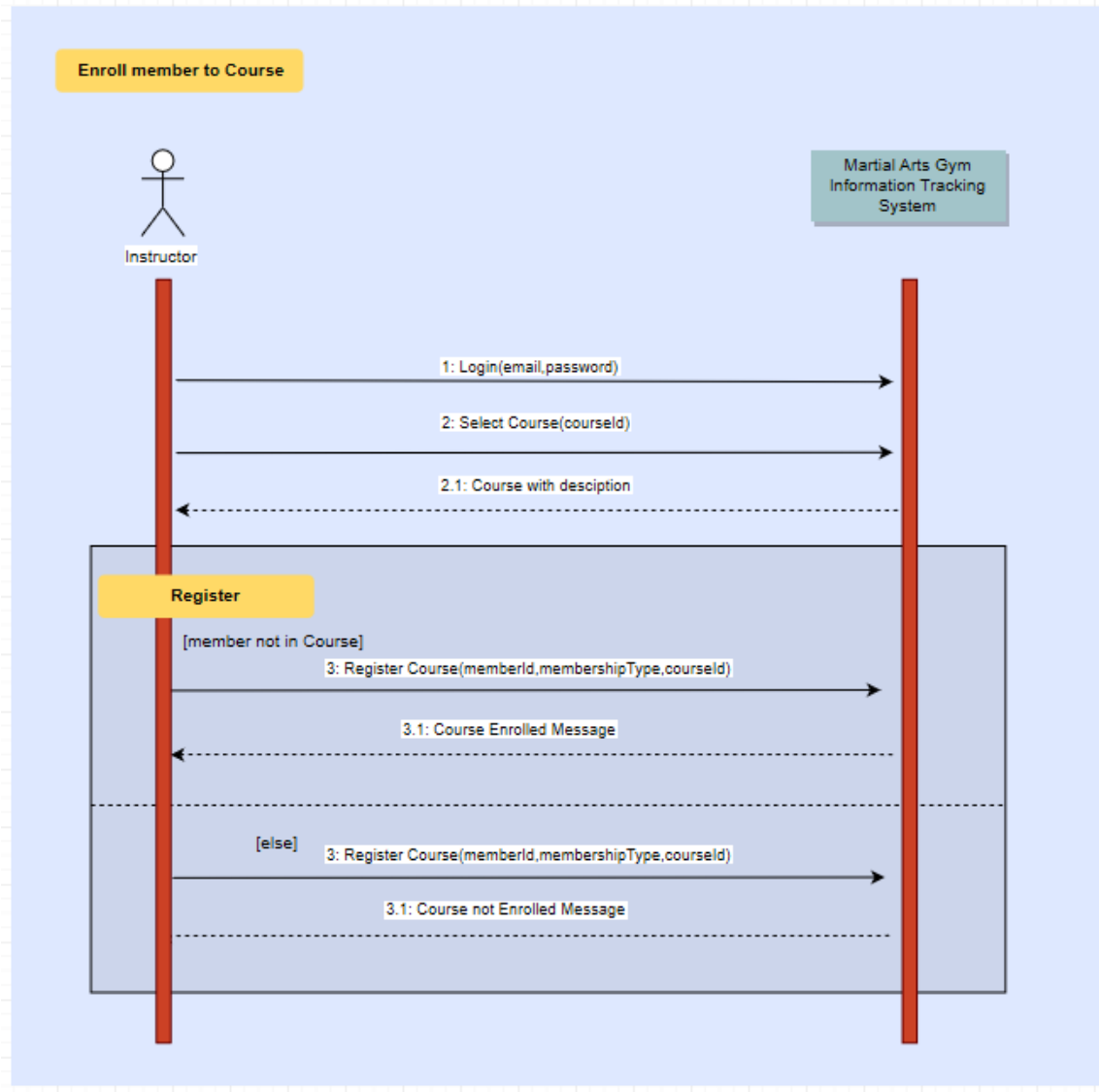


### 3.3.3.3 Instructor Sequence Diagram

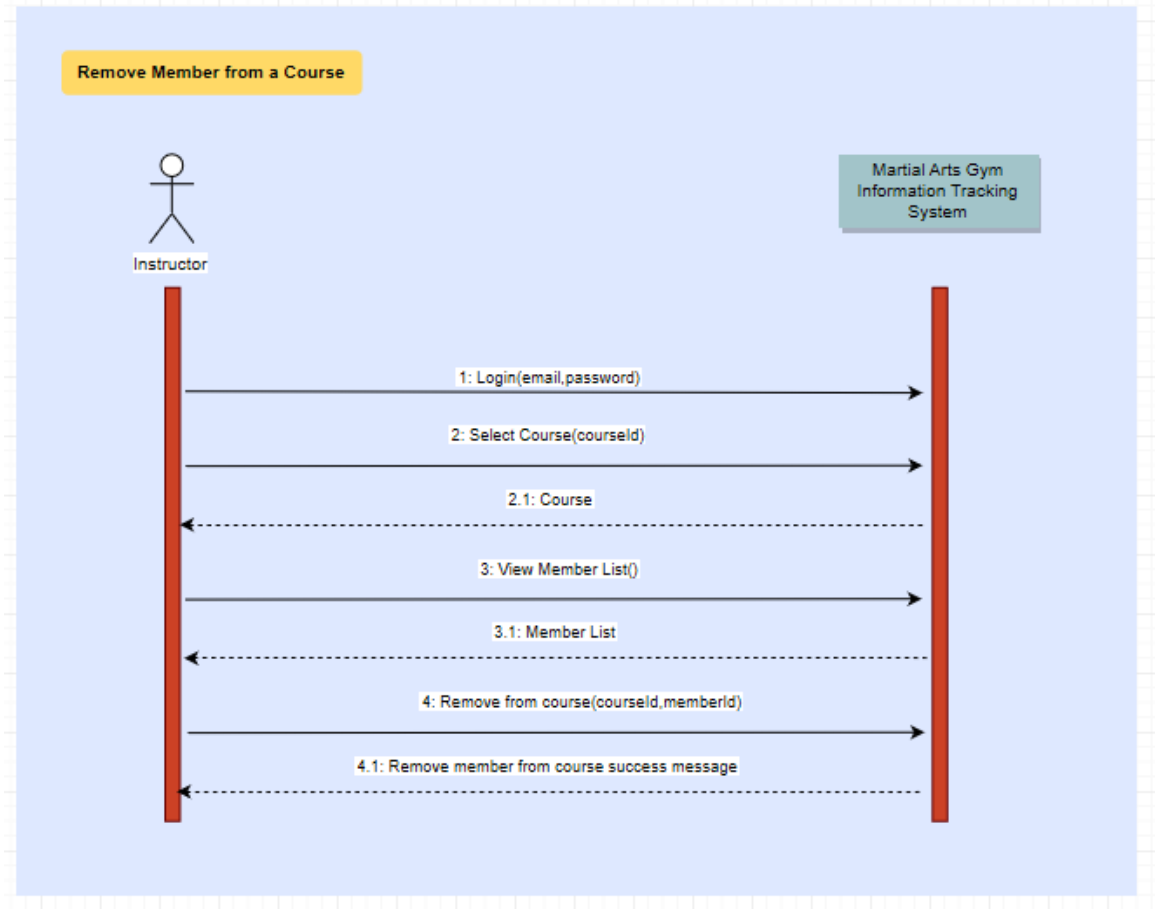
**Add a Member** (refers to Use case 3.2.3.1)



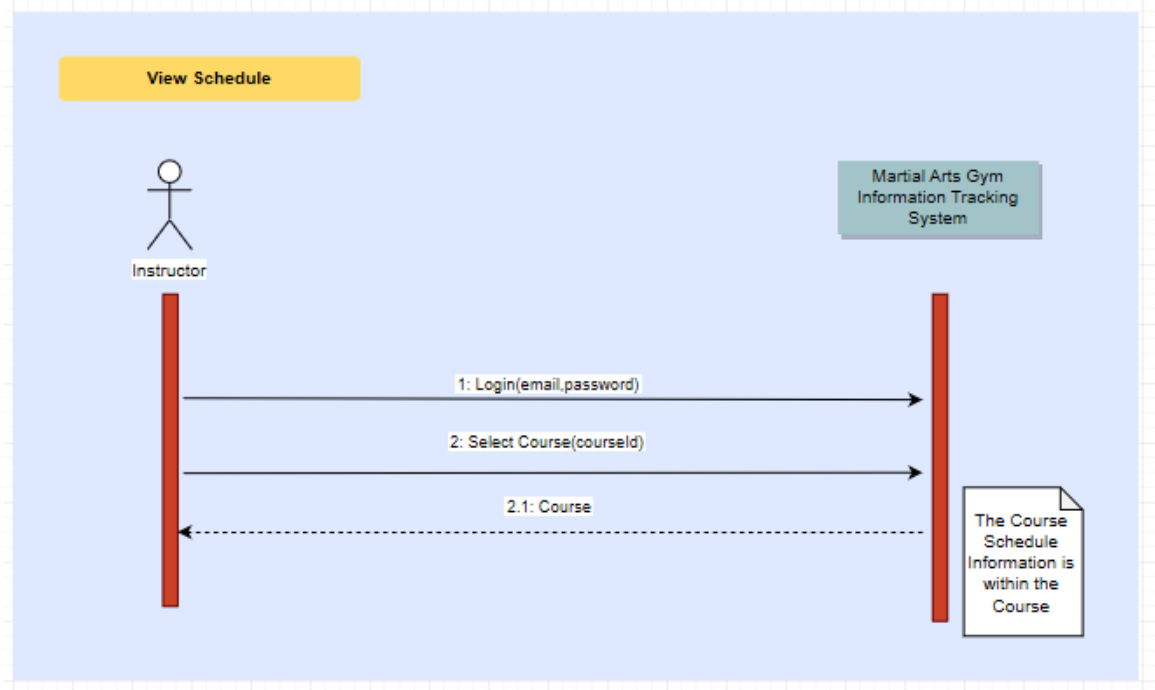
**Enroll Member to course** (refers to Use case 3.2.3.2)



**Remove member from course** (refers to Use case 3.2.3.3)

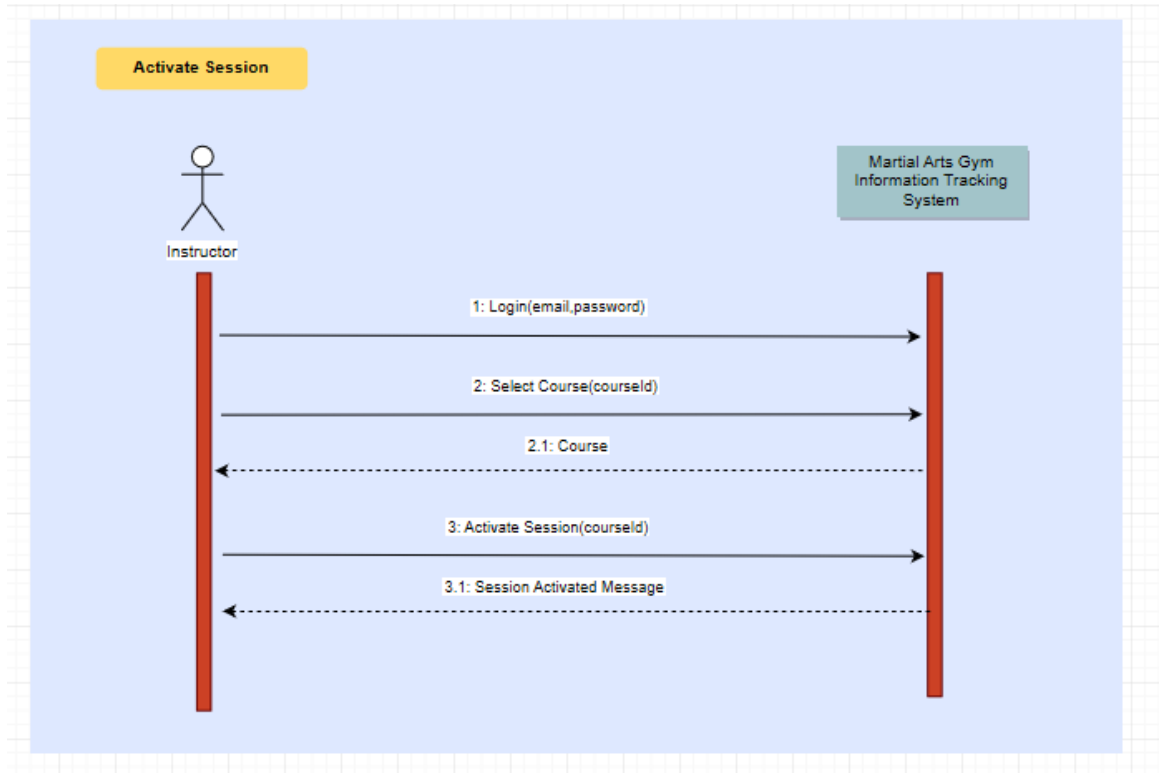


**View Schedule** (refers to Use case 3.2.3.4)

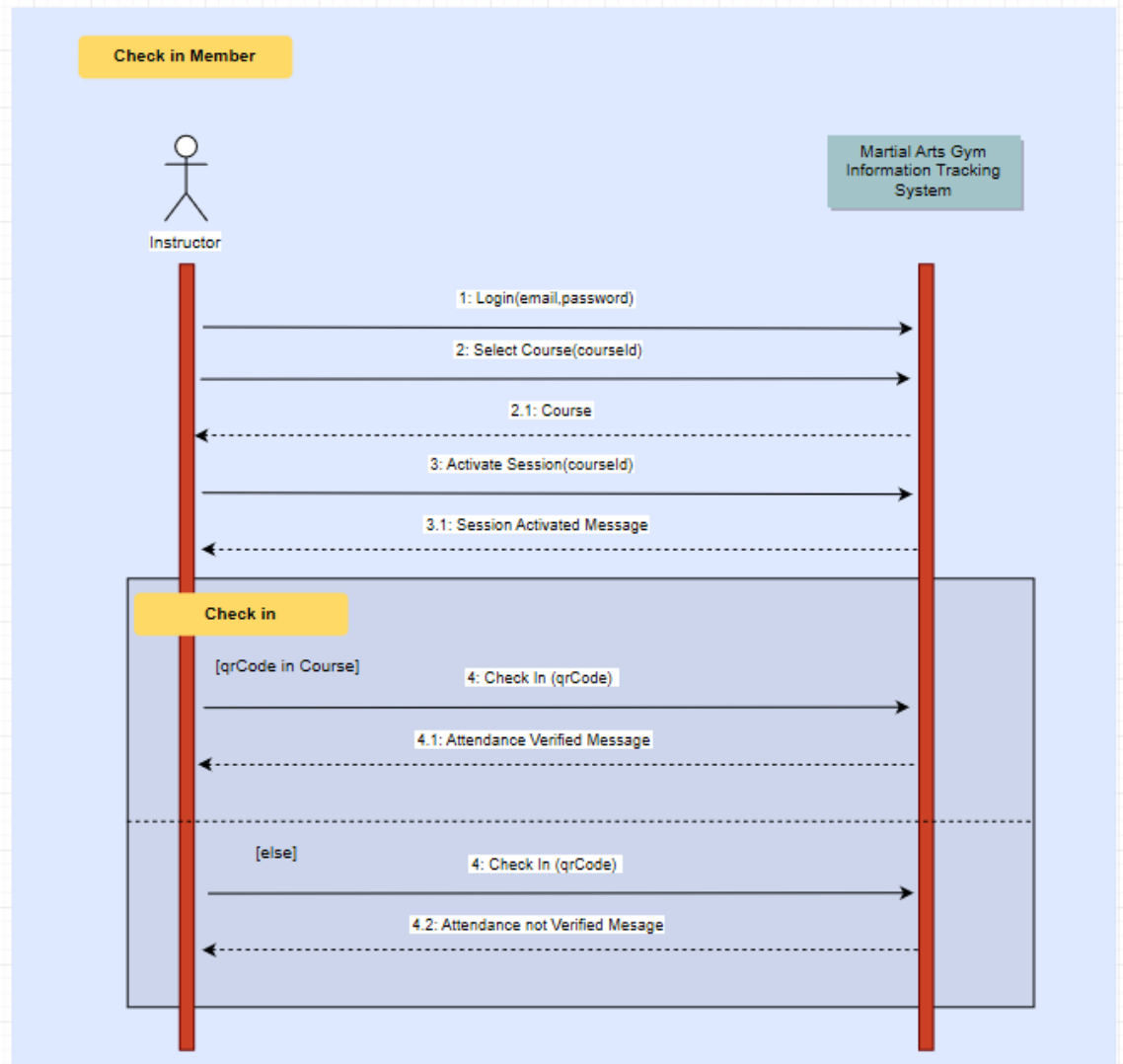


**Activate Session** (refers to Use case 3.2.3.5)

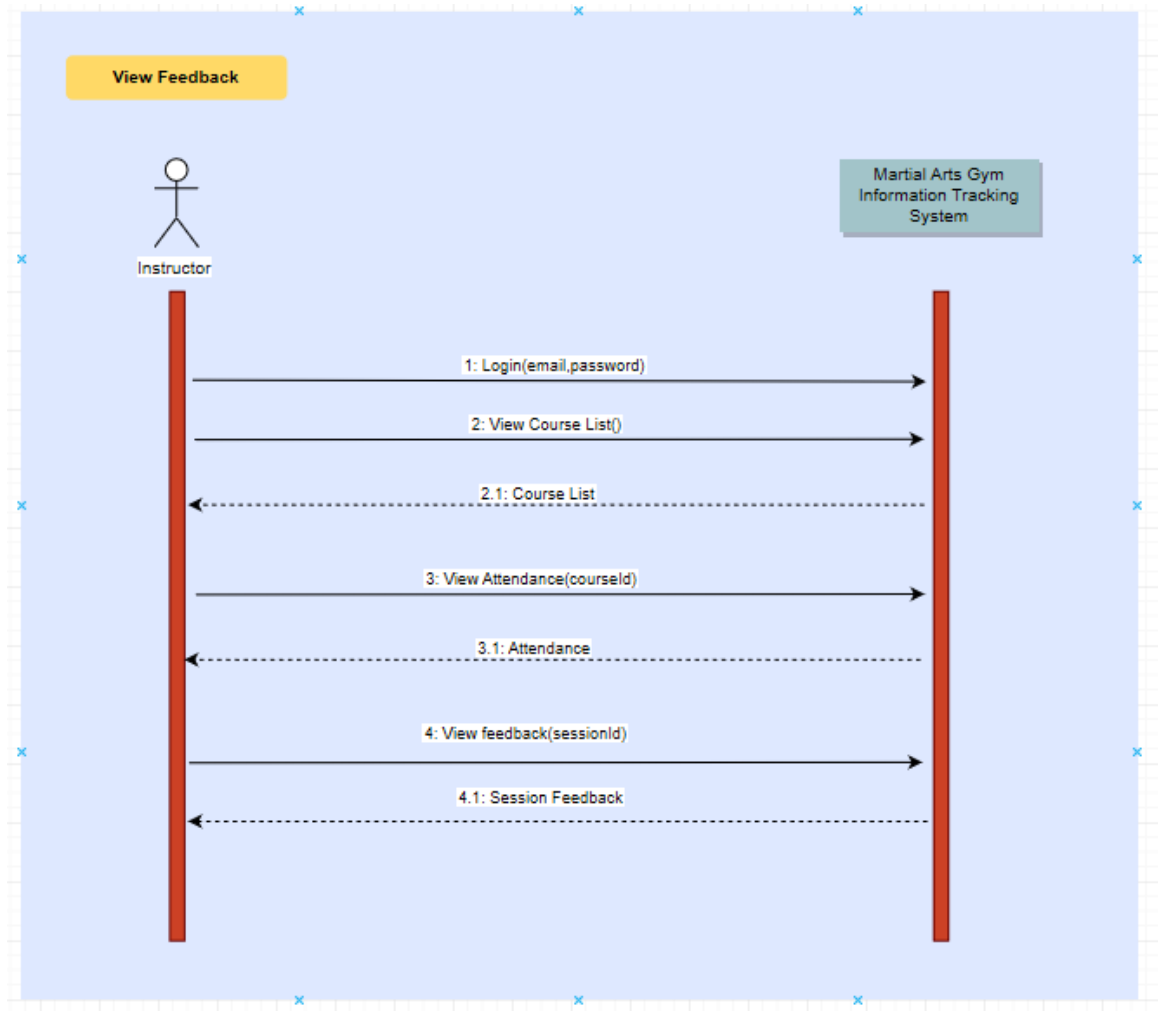




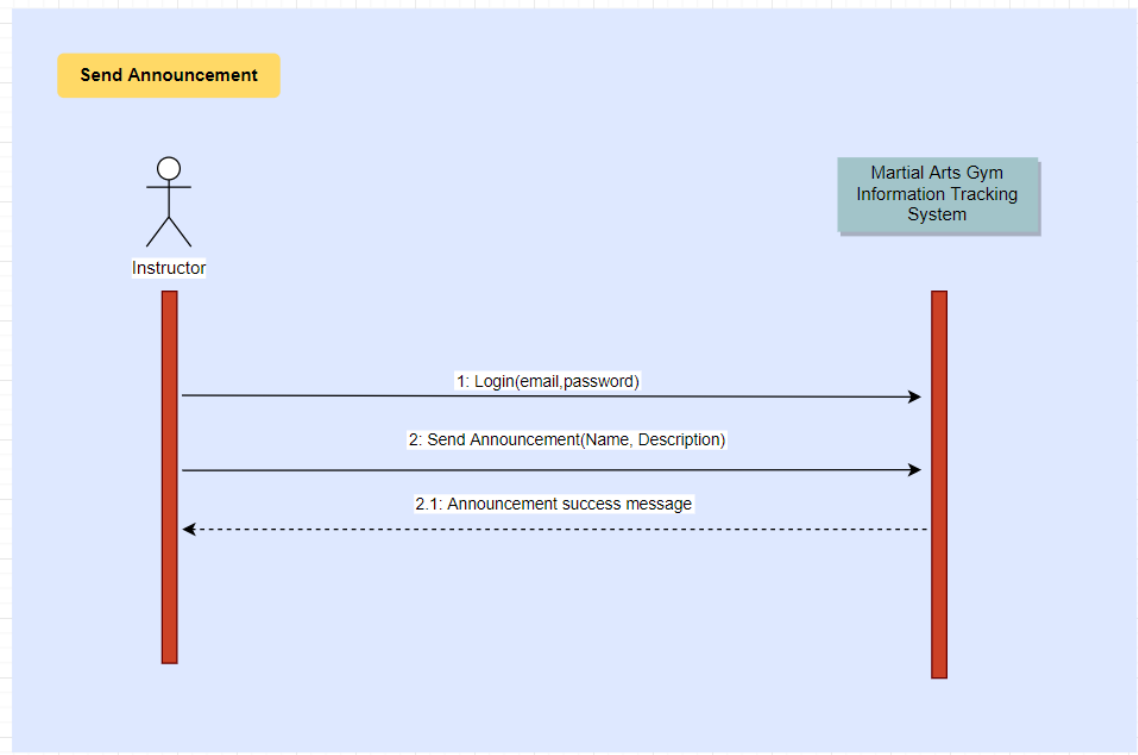
**Check in member to Session** (refers to Use case 3.2.3.6)



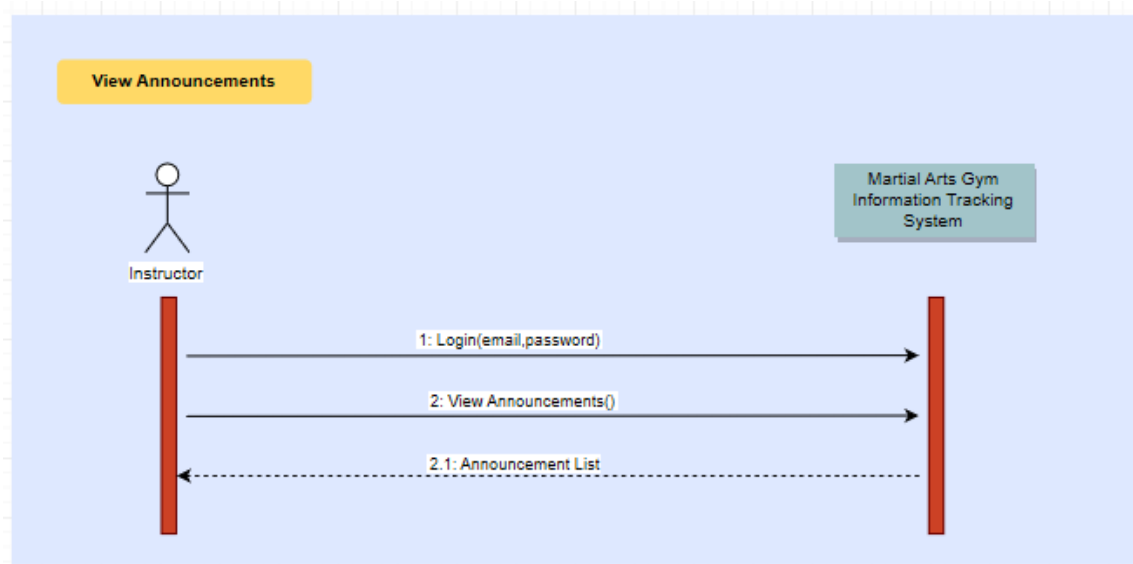
**View Session Feedback** (refers to Use case 3.2.3.7)



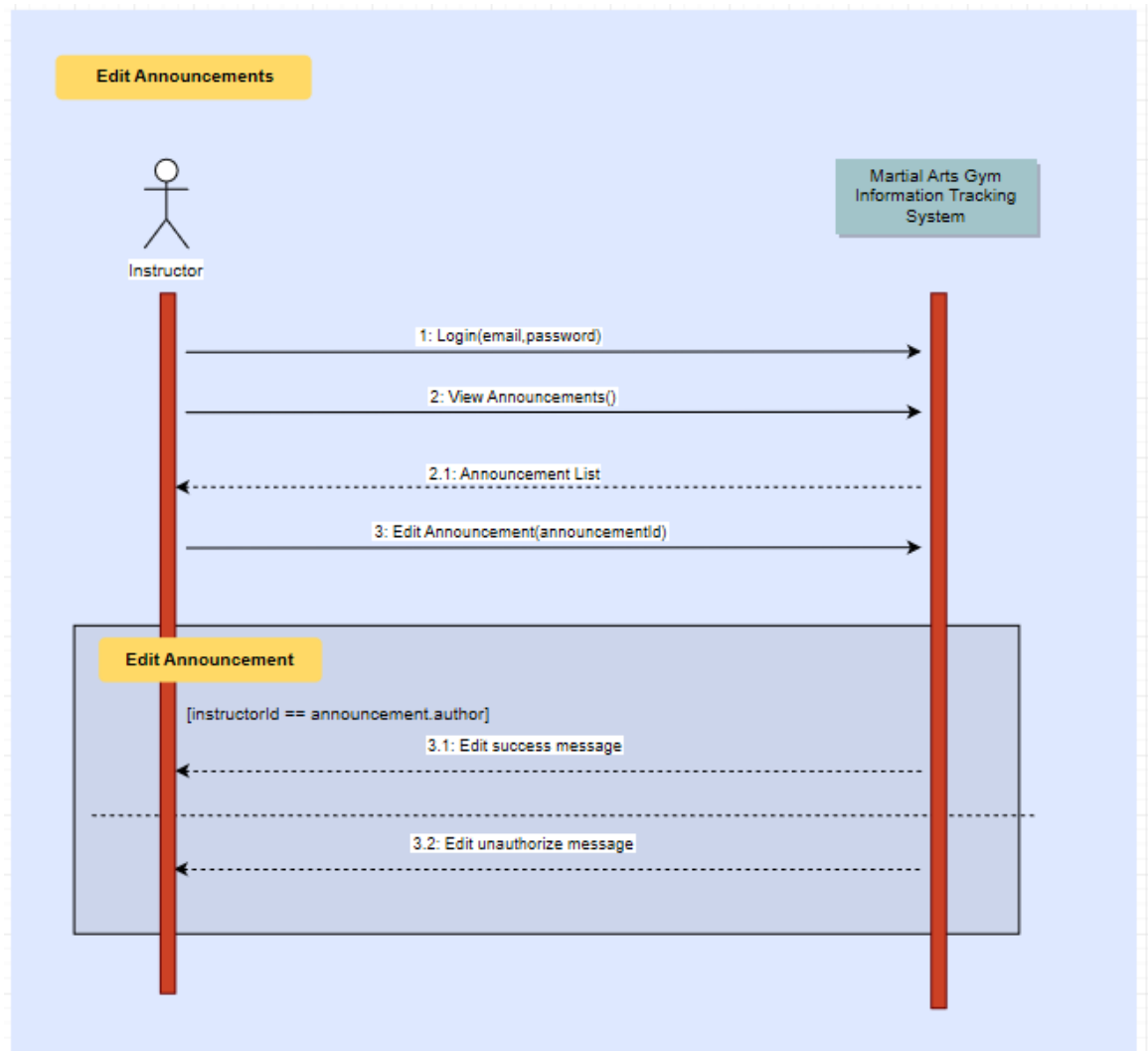
**Send announcement** (refers to Use case 3.2.3.8)



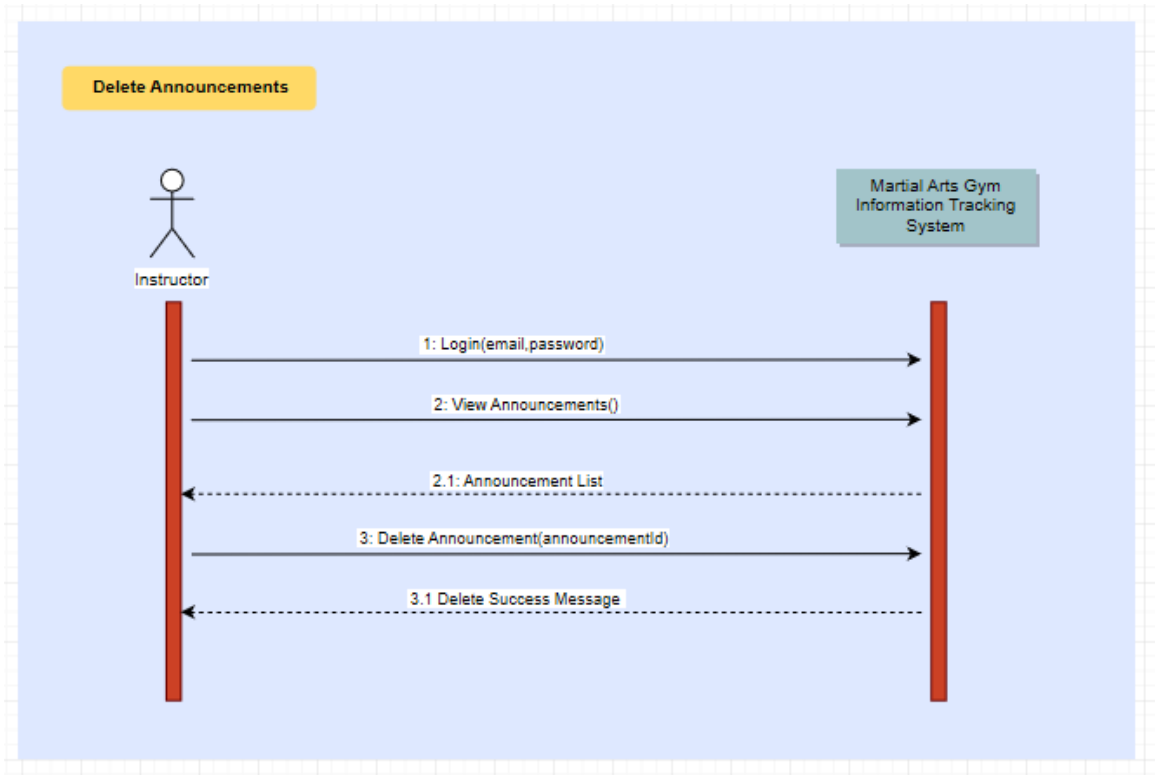
**View announcement** (refers to Use case 3.2.3.9)



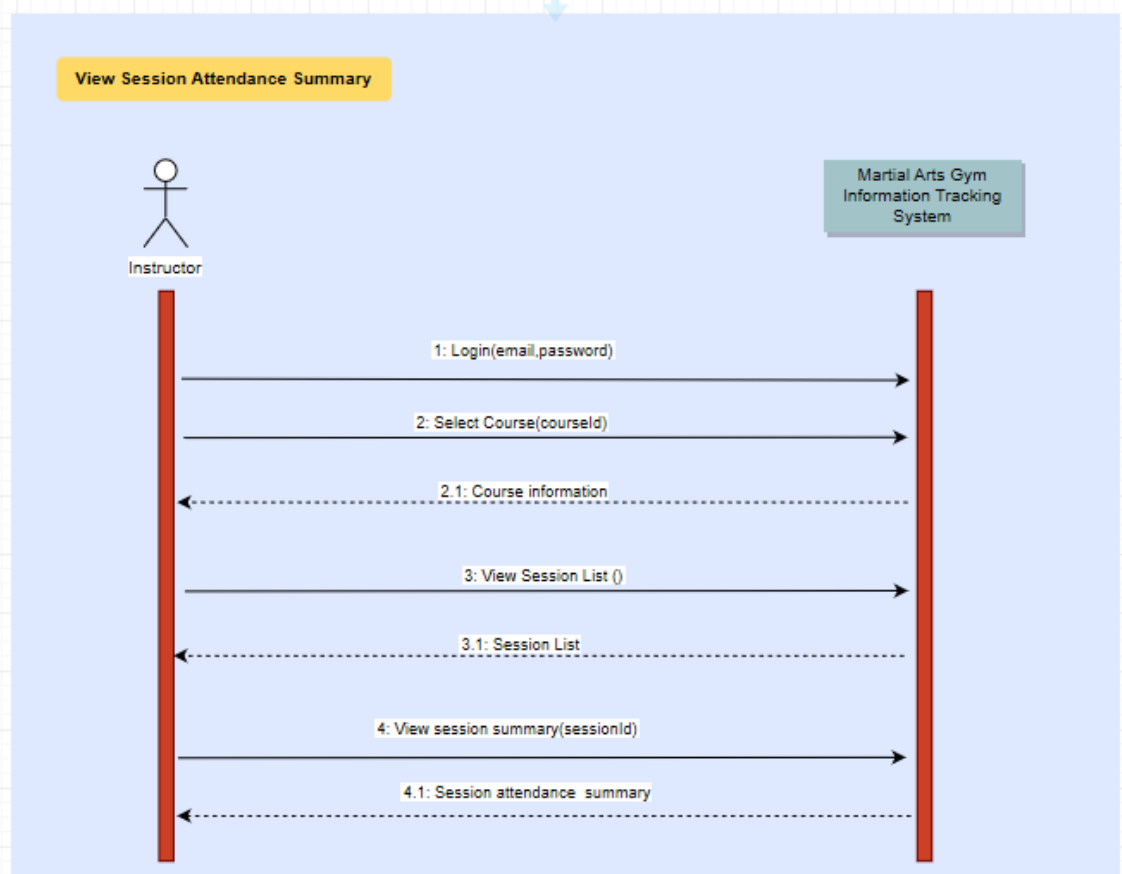
**Edit announcement** (refers to Use case 3.2.3.10)



**Delete announcement** (refers to Use case 3.2.3.11)

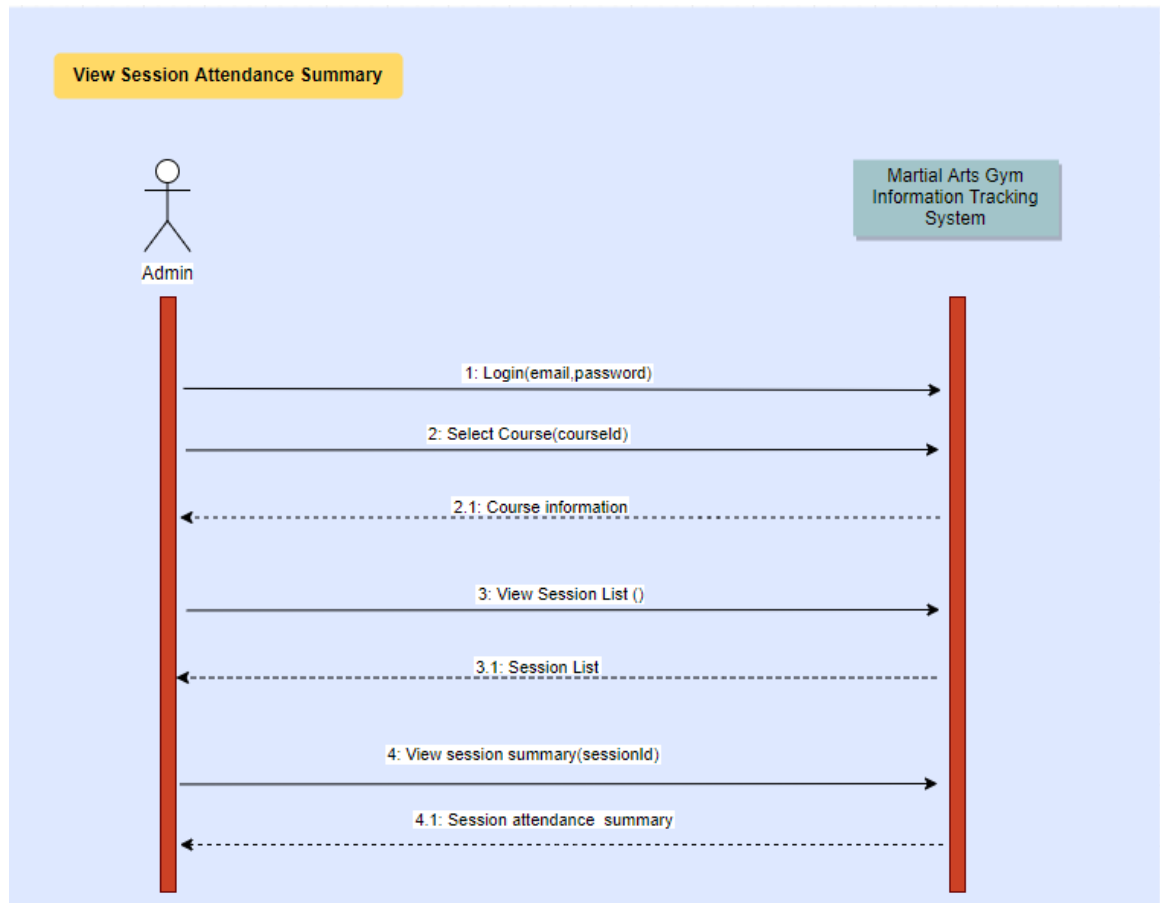


**View Attendance Summary** (refers to Use case 3.2.3.12)



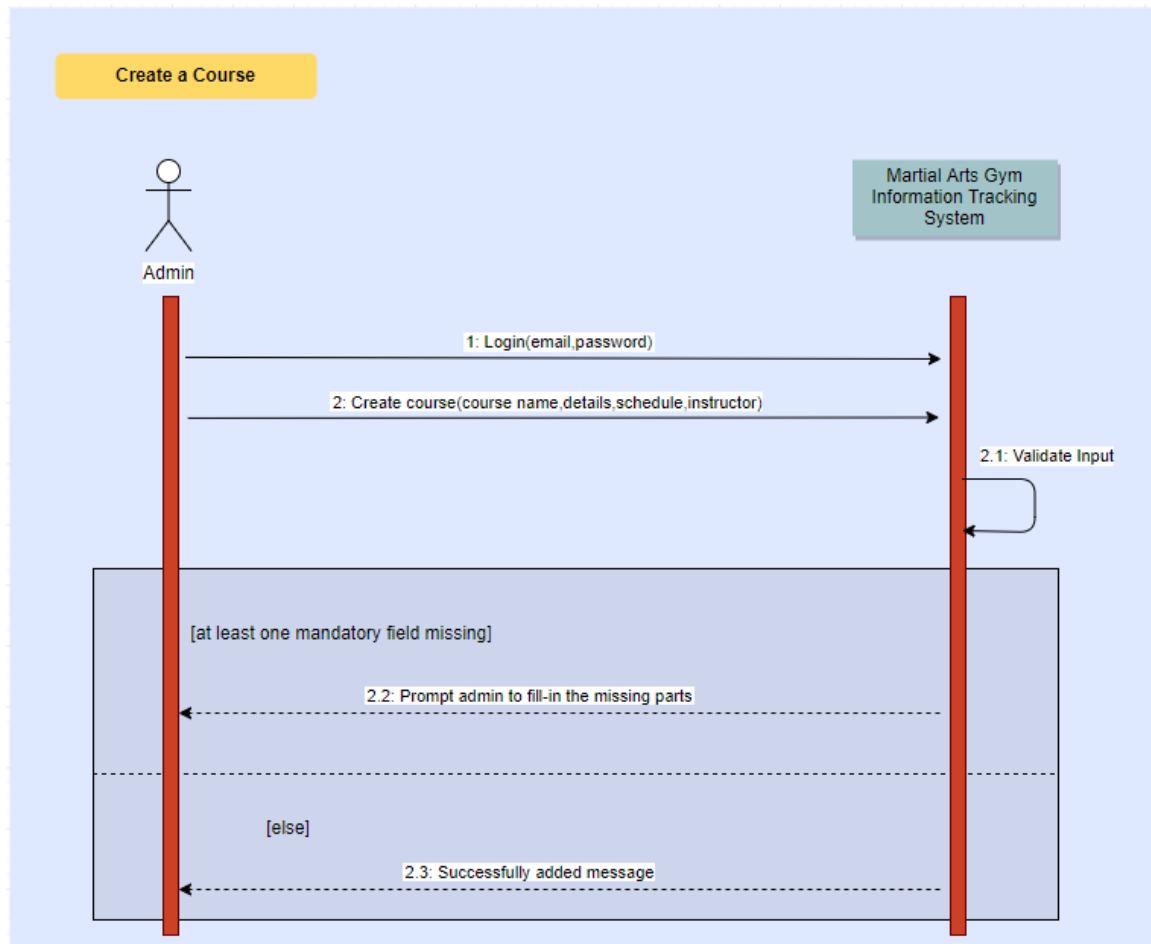
### 3.3.3.4 Admin Sequence Diagram

**View session attendance summary** (refers to use case 3.2.4.1)

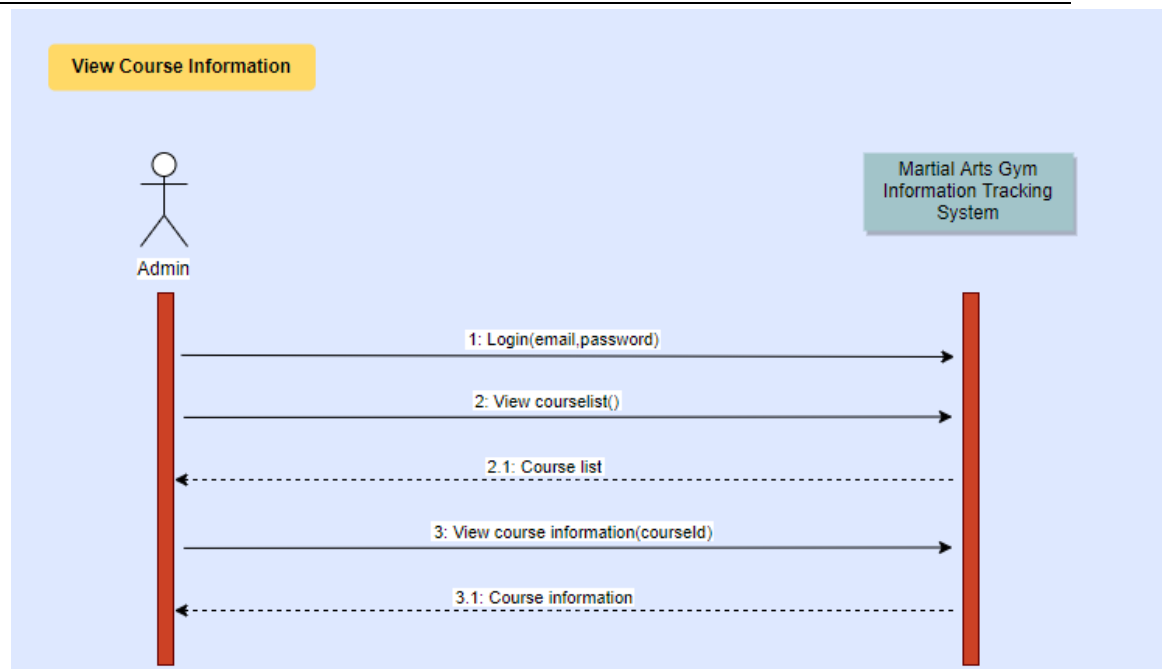


**Create a course** (refers to Use case 3.2.4.2)

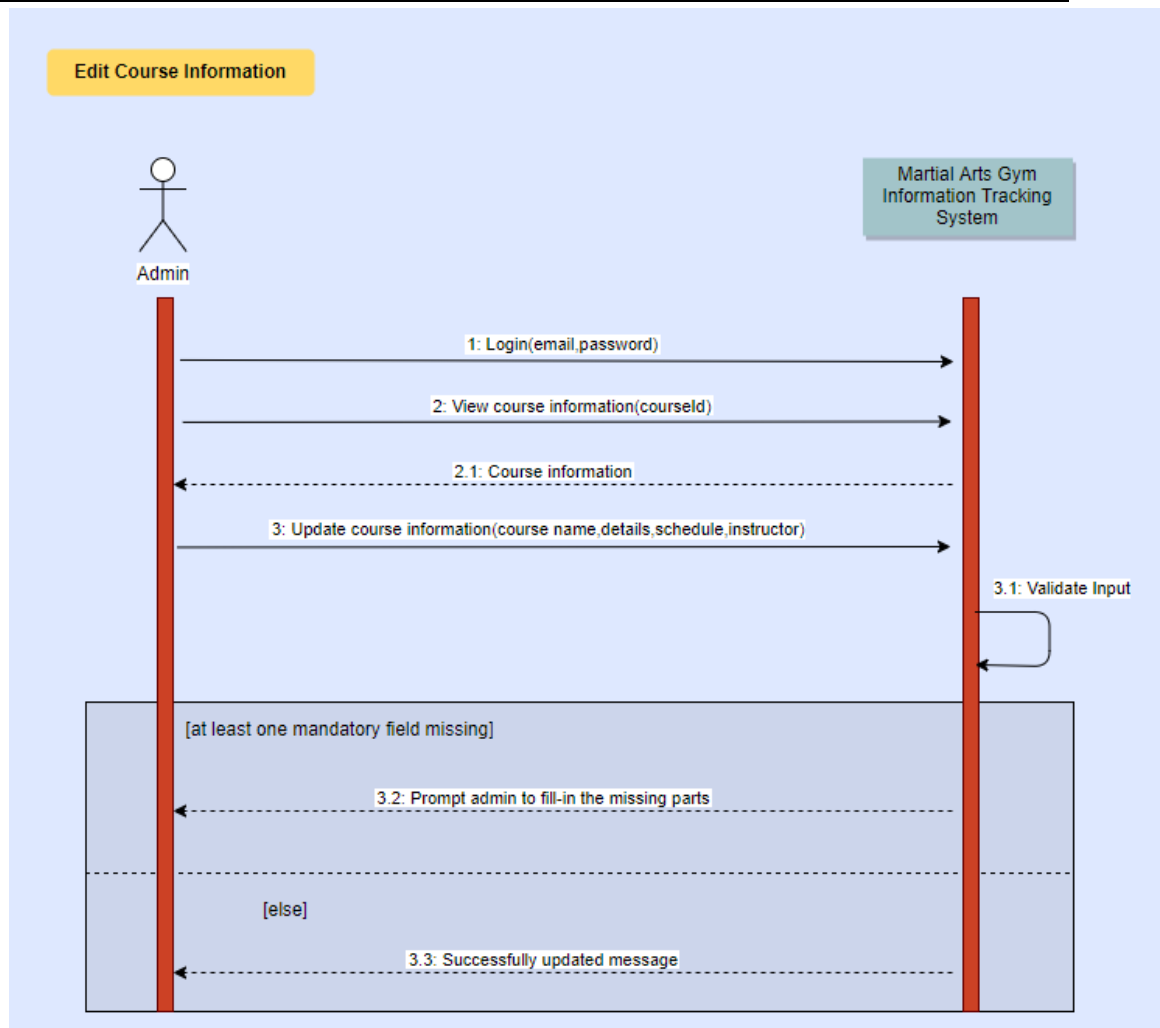




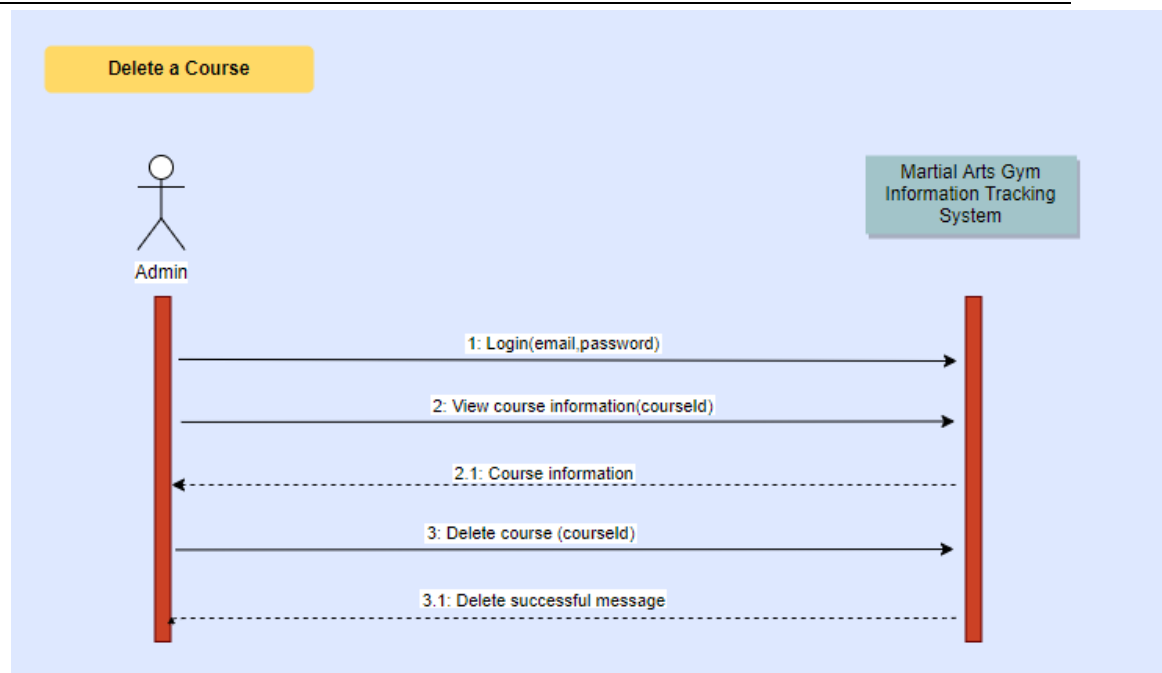
**View Course Information** (refers to use case 3.2.4.3)



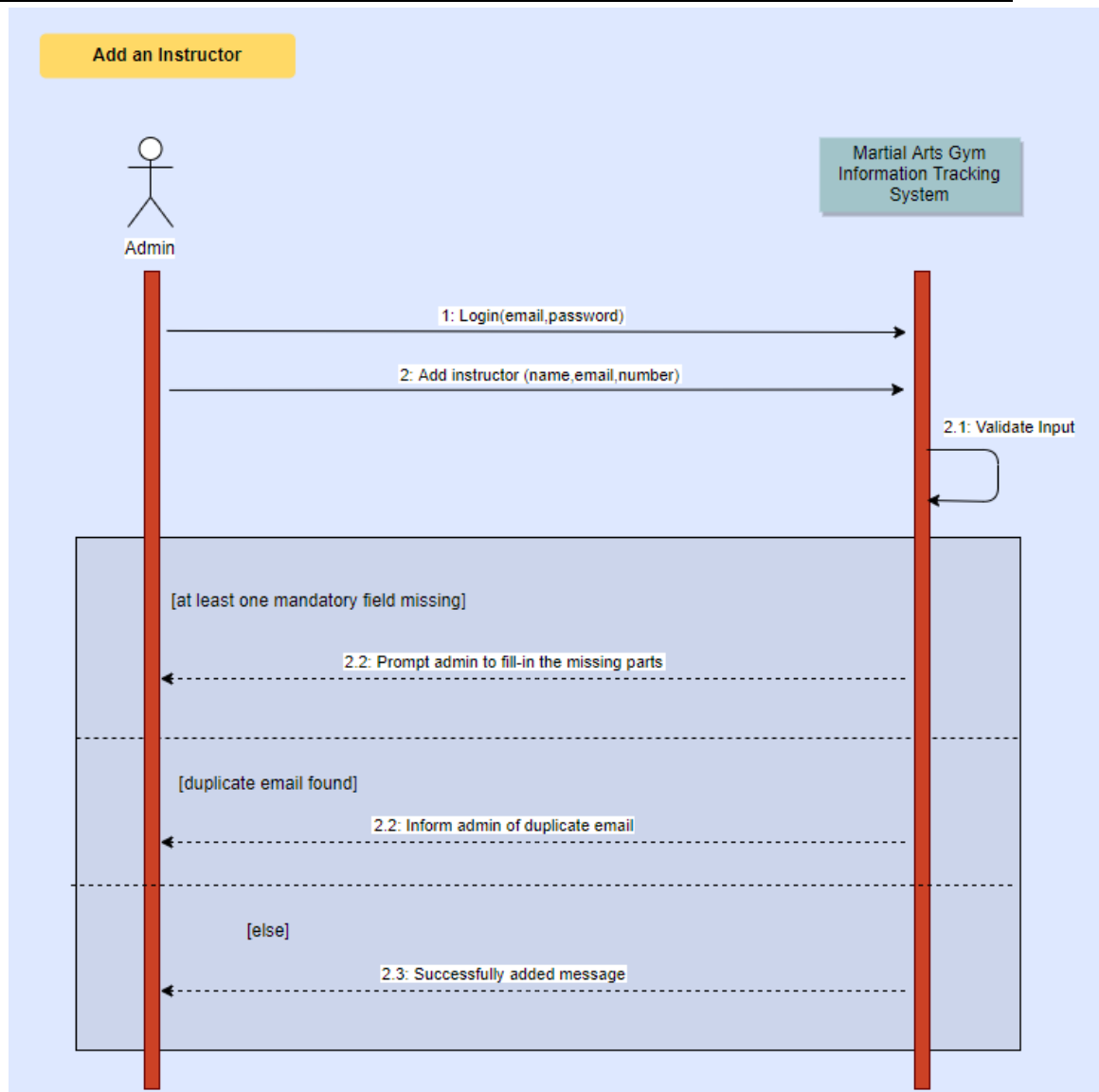
**Edit course information** (refers to use case 3.2.4.4)



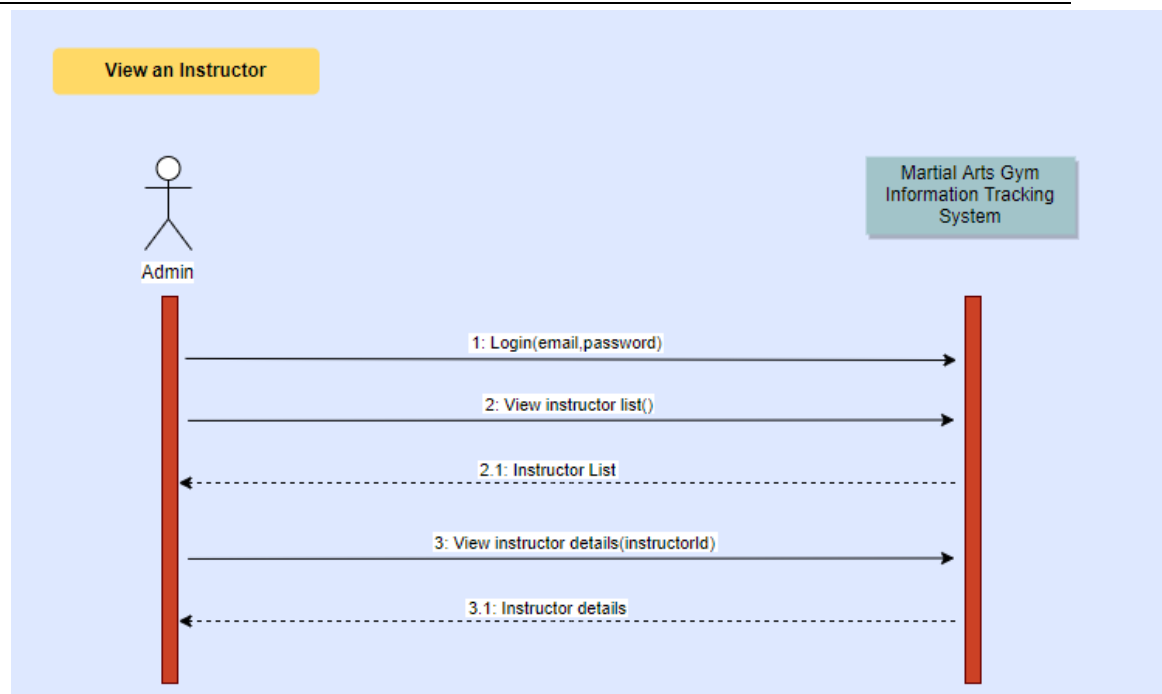
**Delete a course** (refers to use case 3.2.4.5)



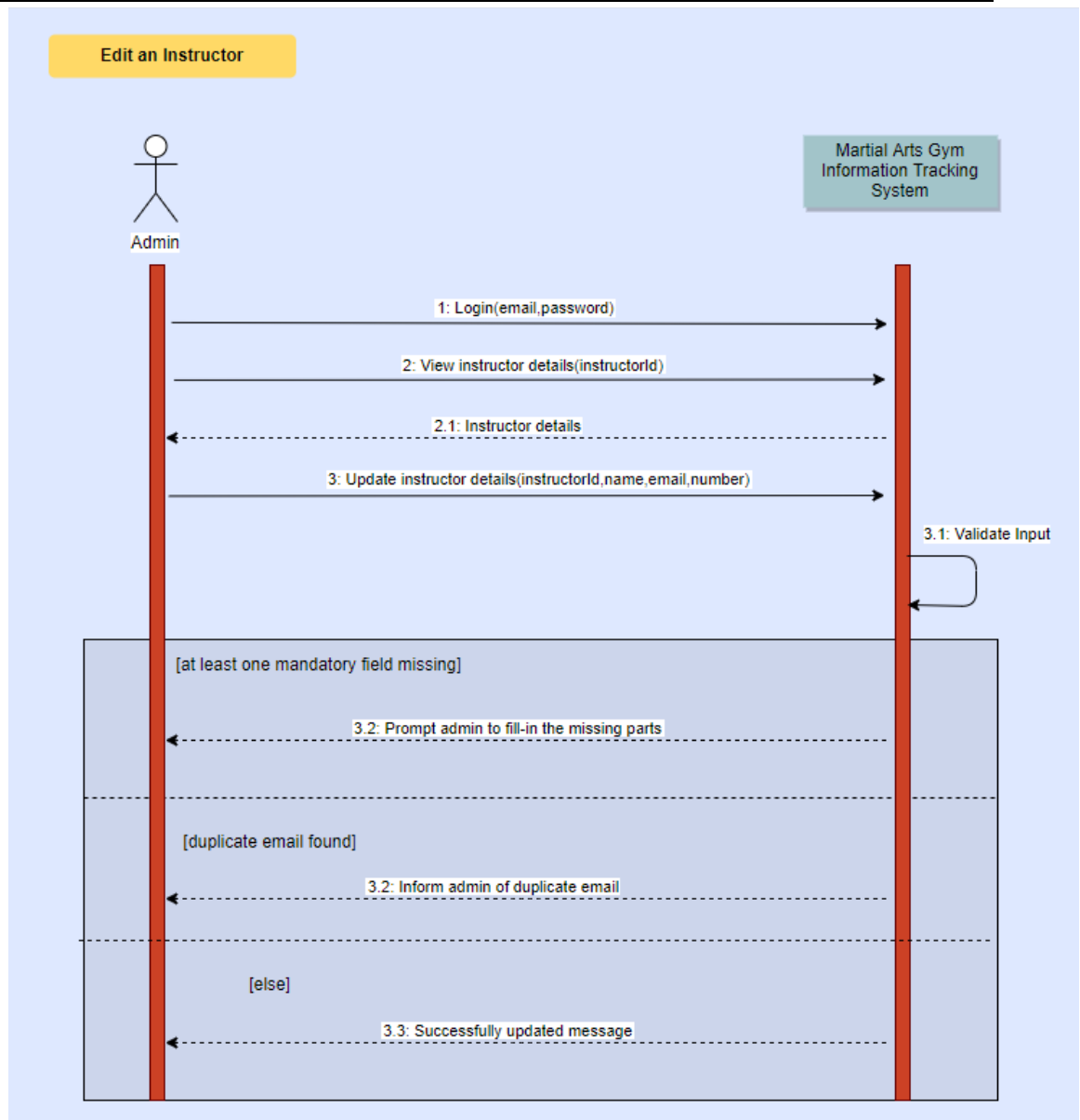
**Add an instructor** (refers to use case 3.2.4.6)



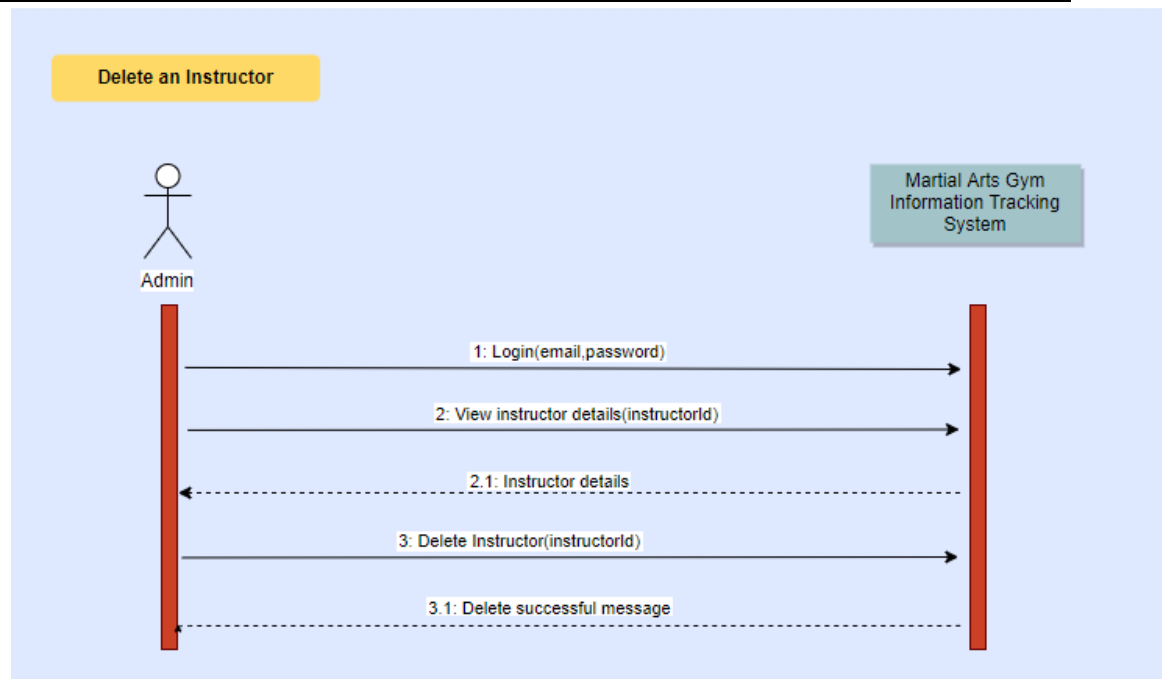
**View instructor** (refers to use case 3.2.4.7)



**Edit instructor** (refers to use case 3.2.4.8)

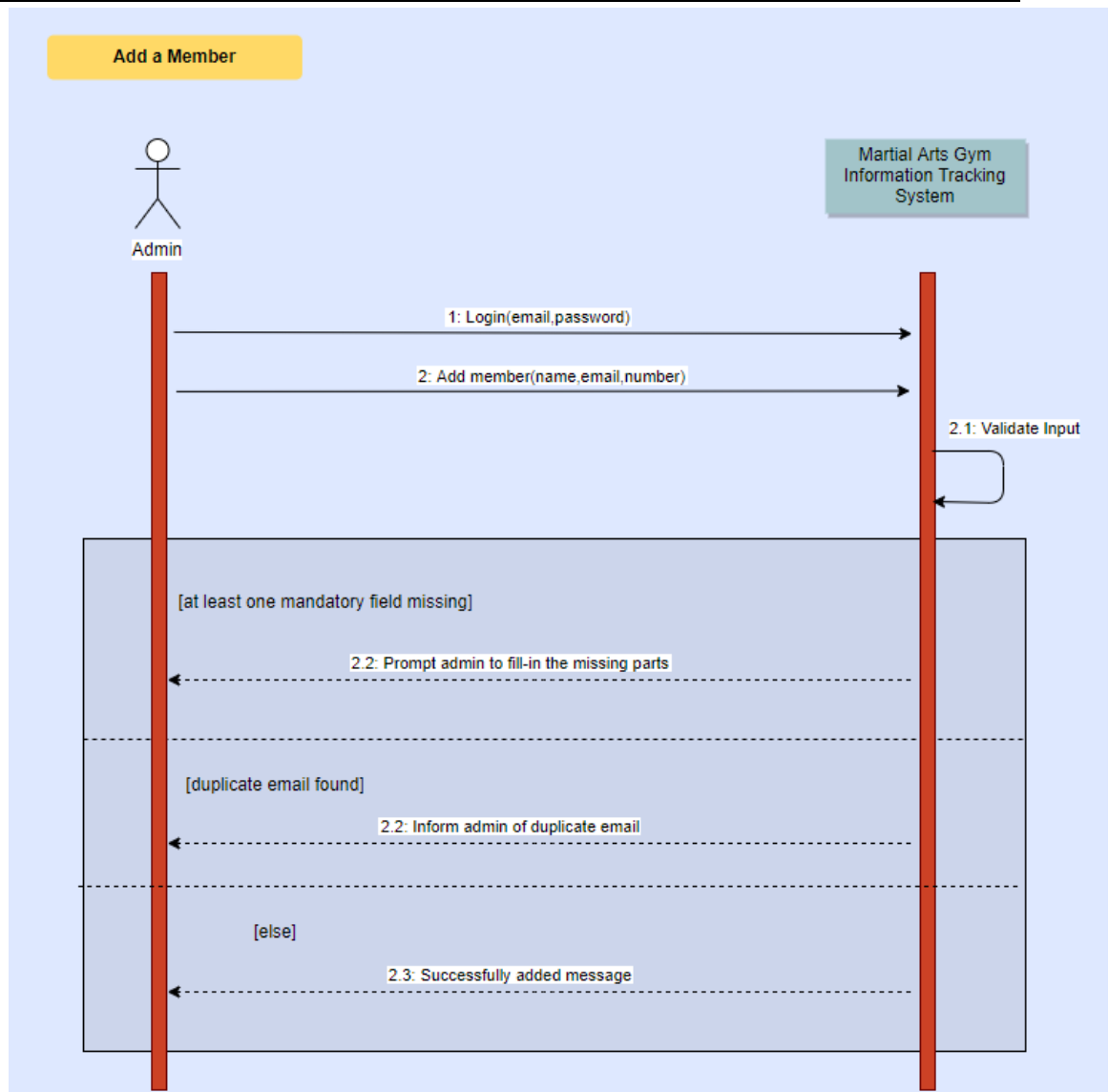


**Delete instructor** (refers to use case 3.2.4.9)

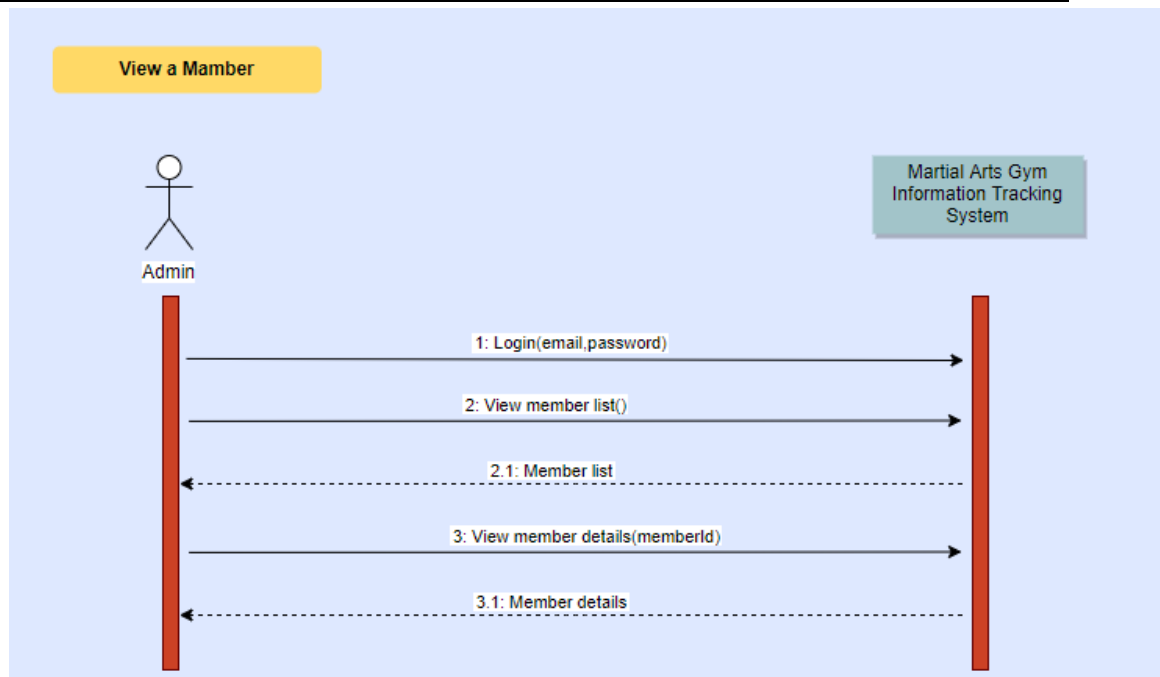


**Add a member** (refers to use case 3.2.4.10)

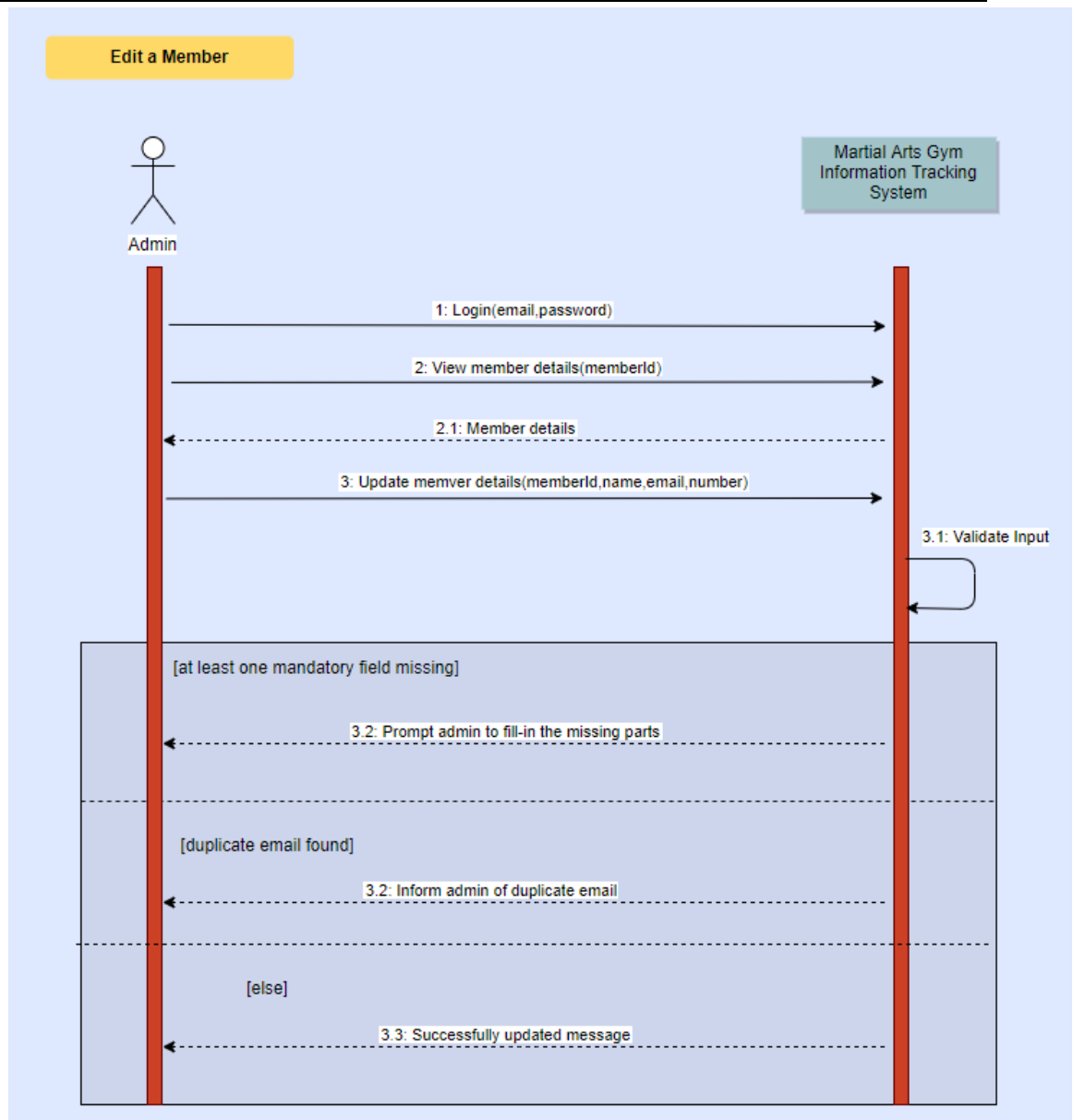




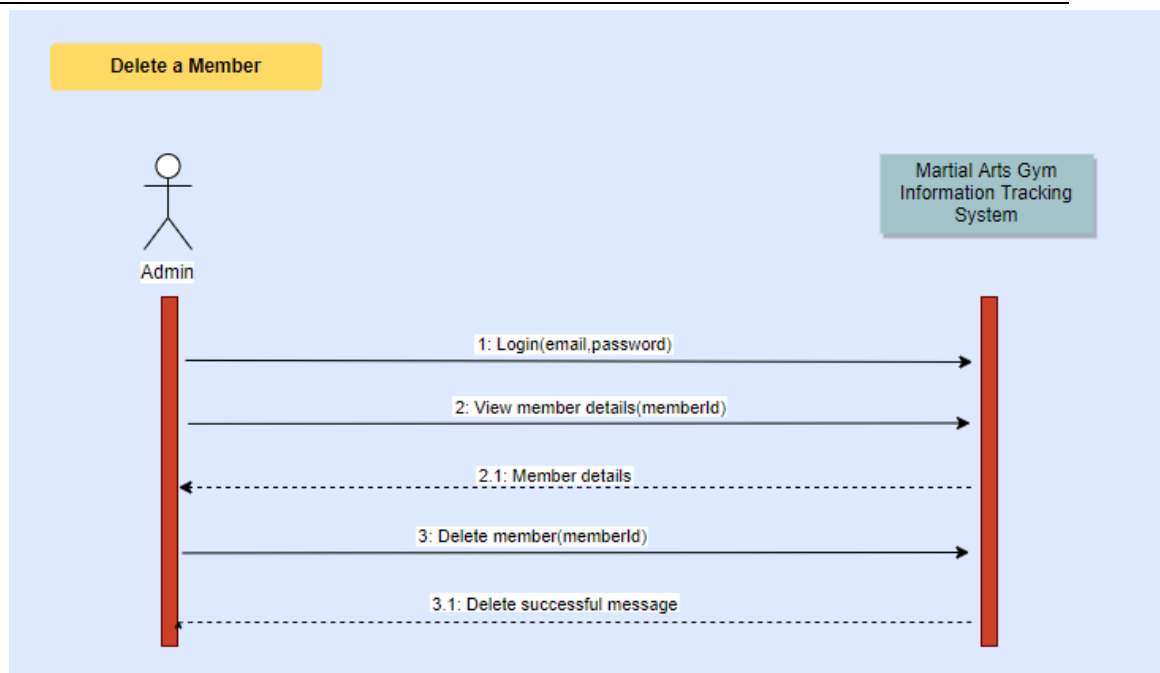
**View member information** (refers to use case 3.2.4.11)



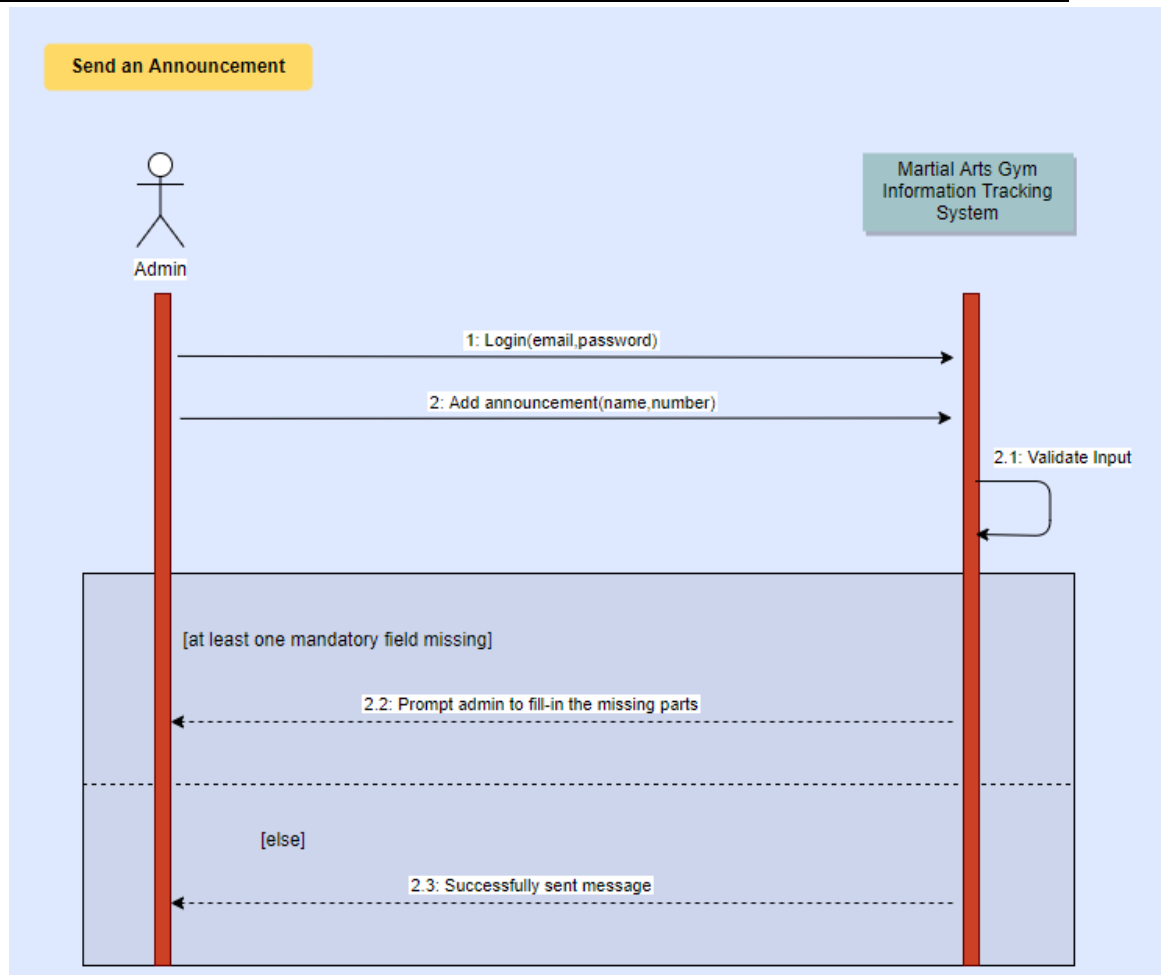
**Edit member information** (refers to use case 3.2.4.12)



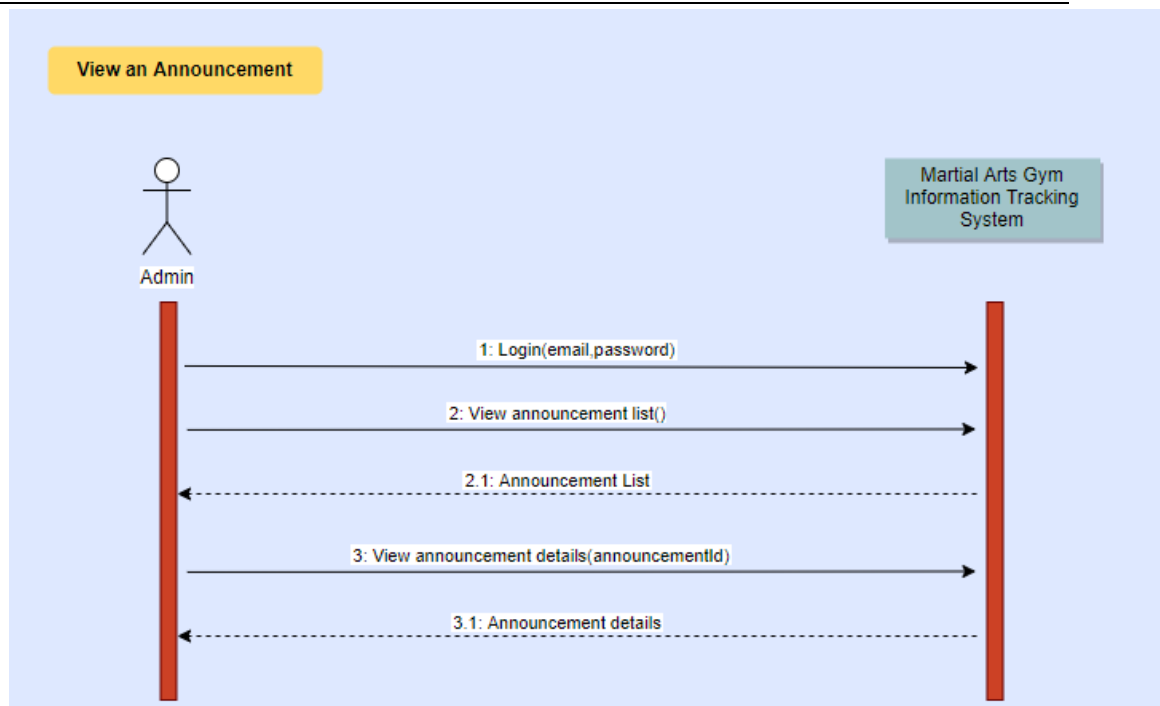
**Delete a member** (refers to use case 3.2.4.13)



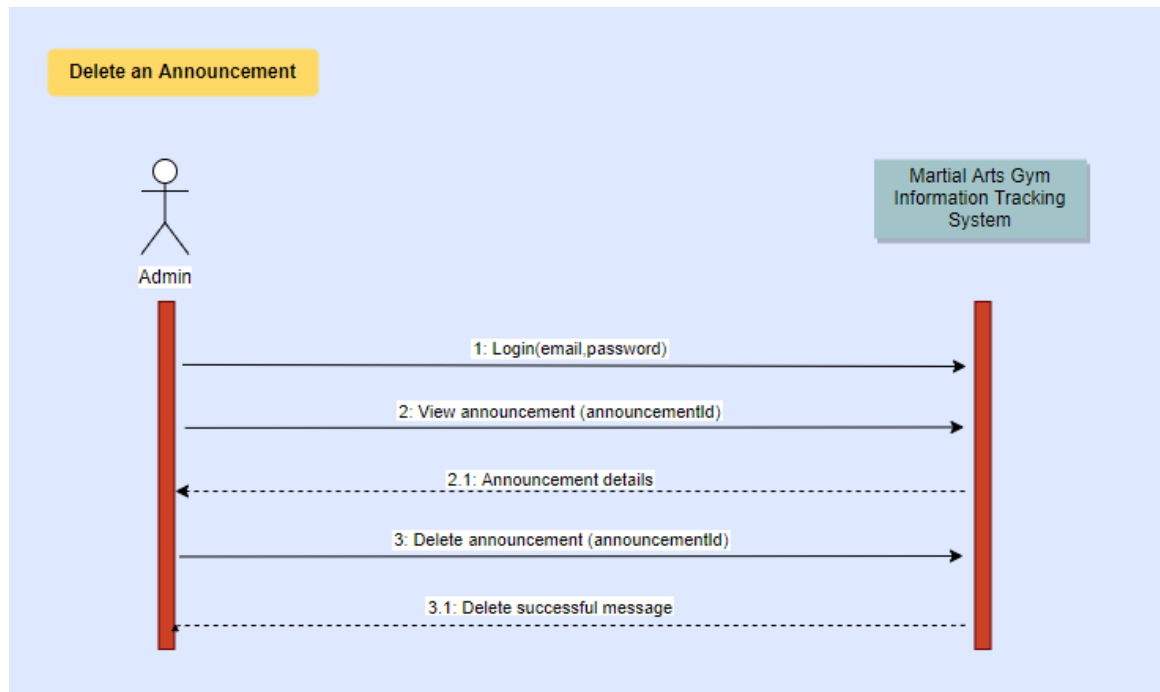
**Send an announcement**(refers to use case 3.2.4.14)



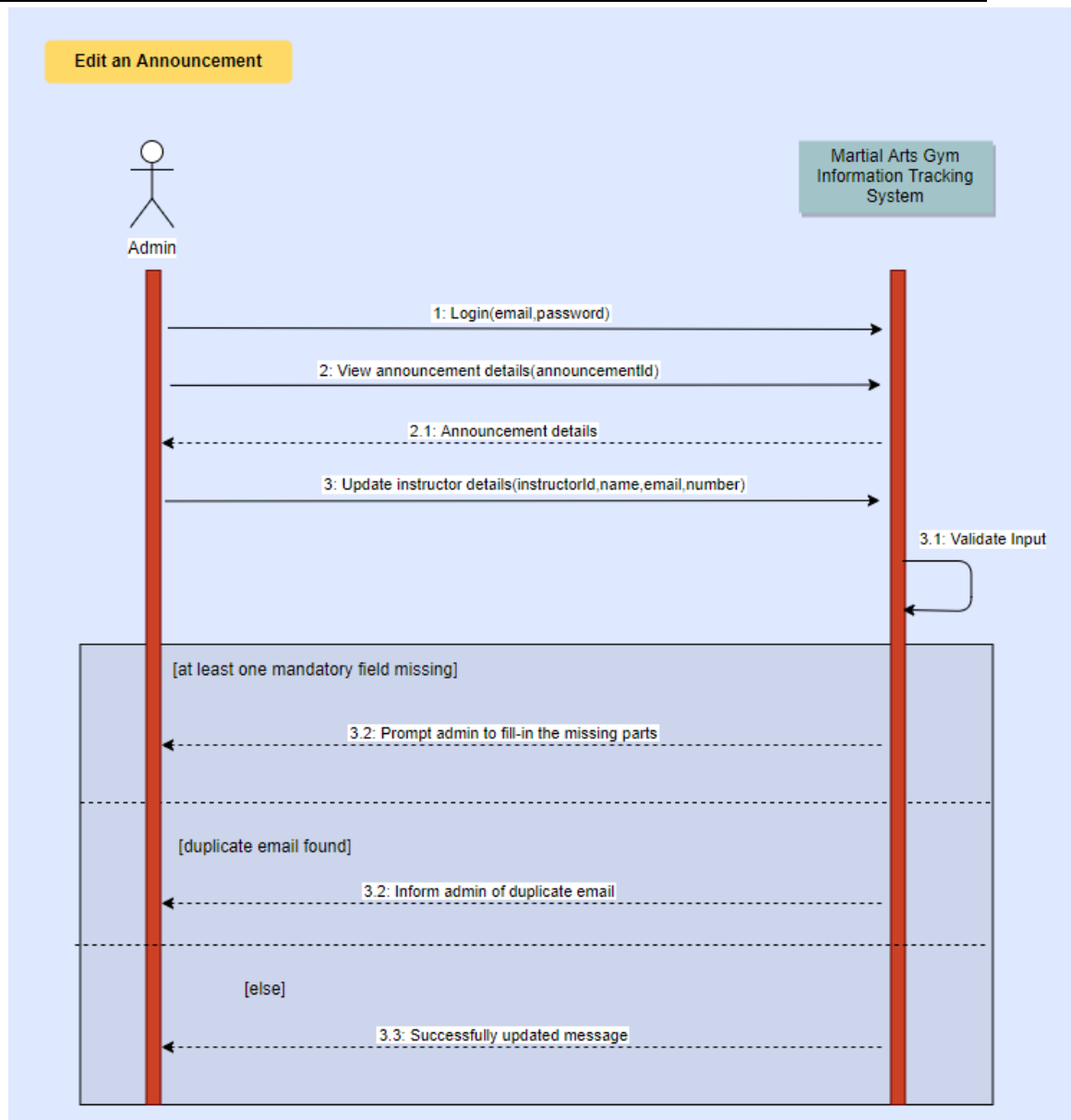
**View an announcement** (refers to use case 3.2.4.15)



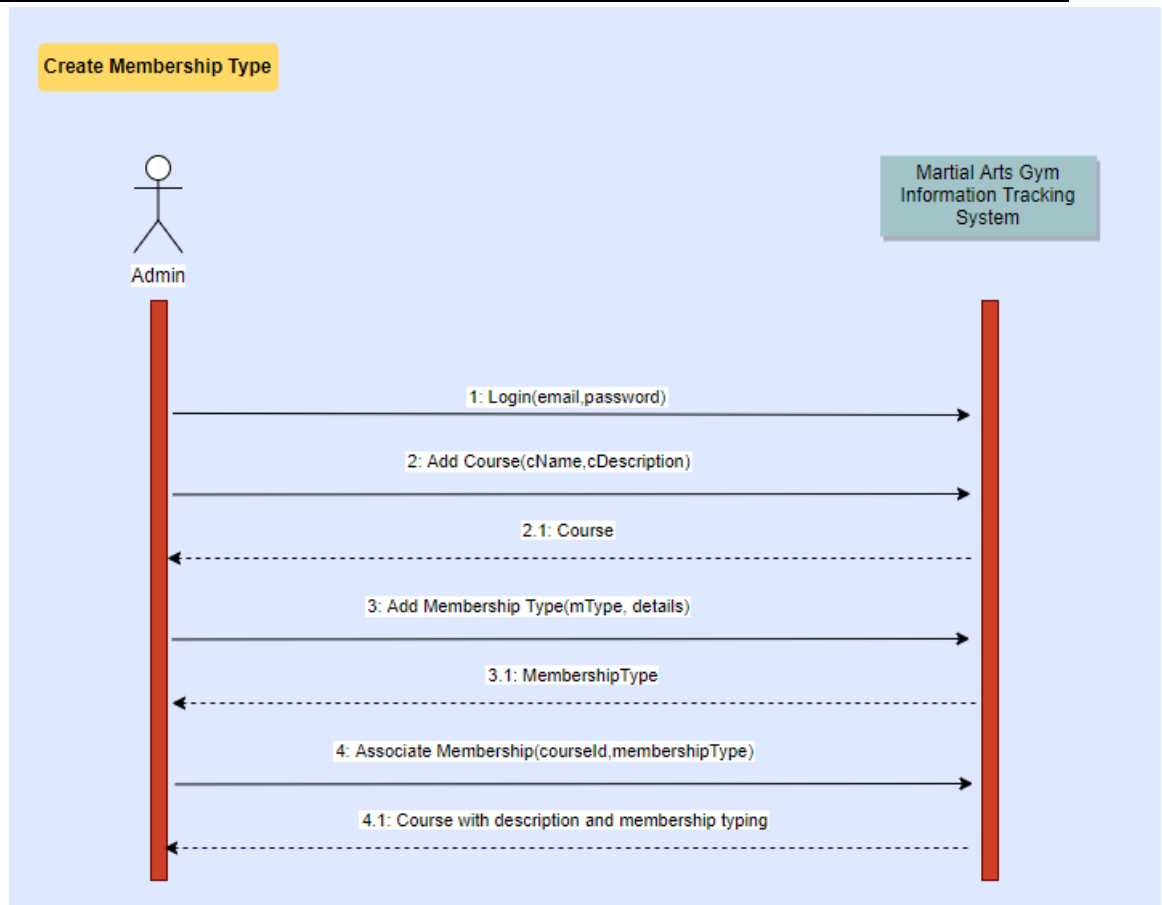
**Delete an announcement** (refers to use case 3.2.4.16)



**Edit announcement** (refers to use case 3.2.4.17)

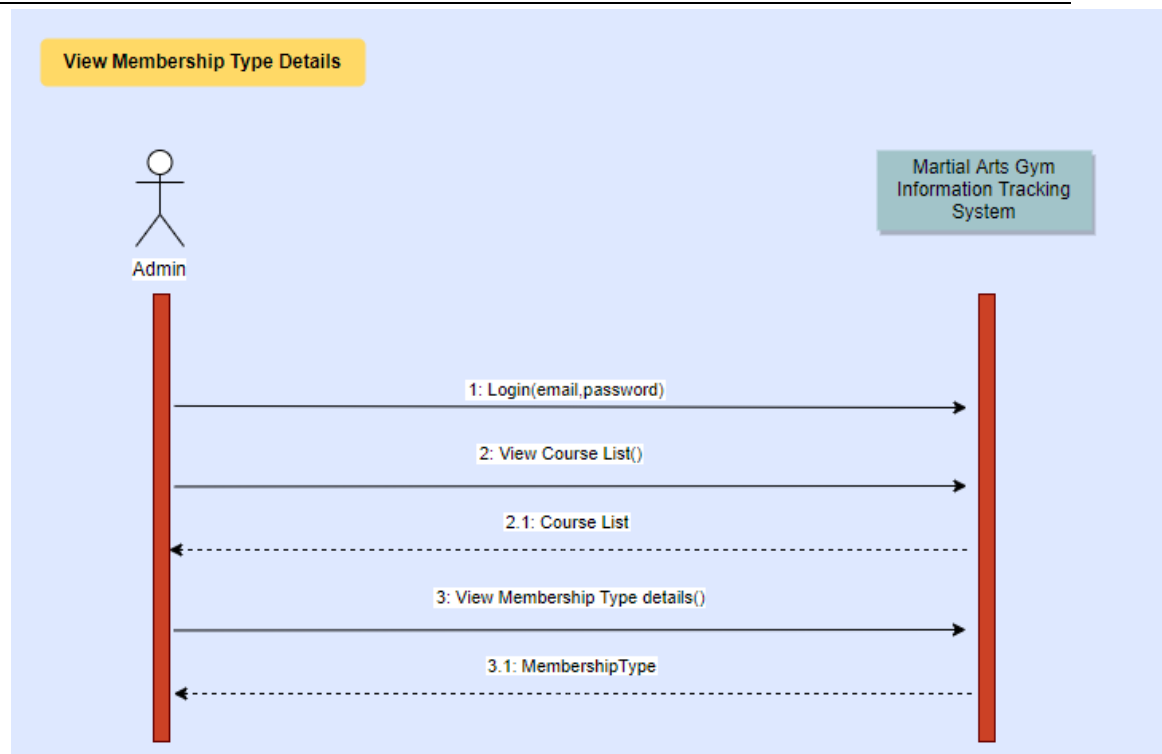


**Create membership type** (refers to use case 3.2.4.18)

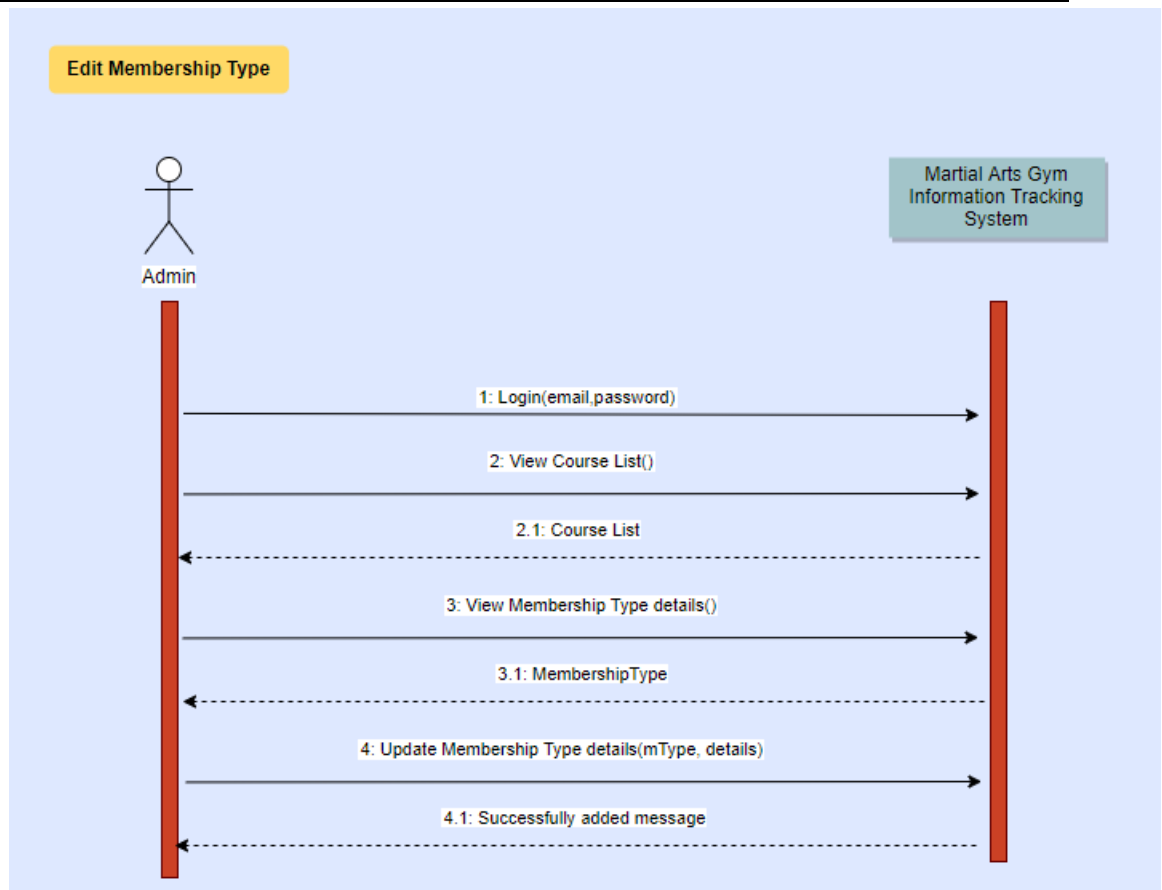


**View membership types** (refers to use case 3.2.4.19)

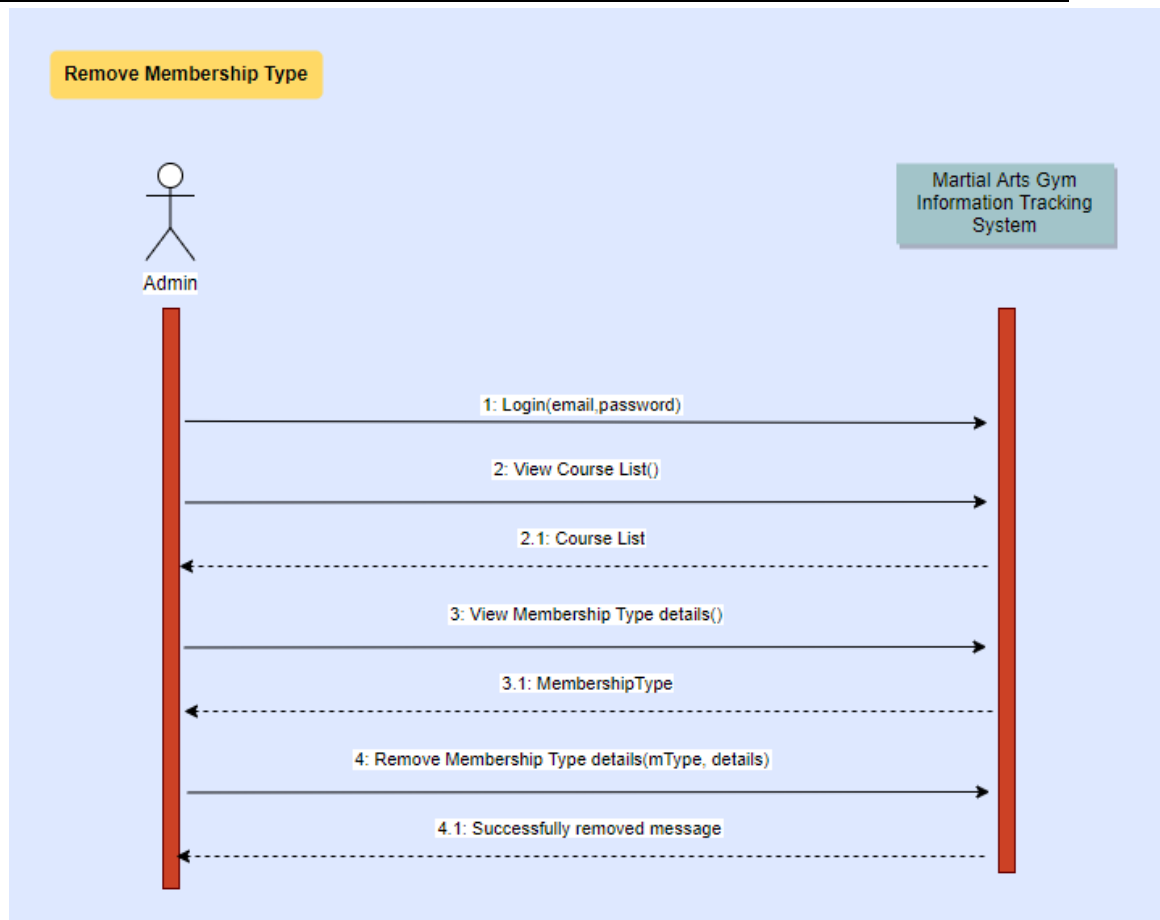




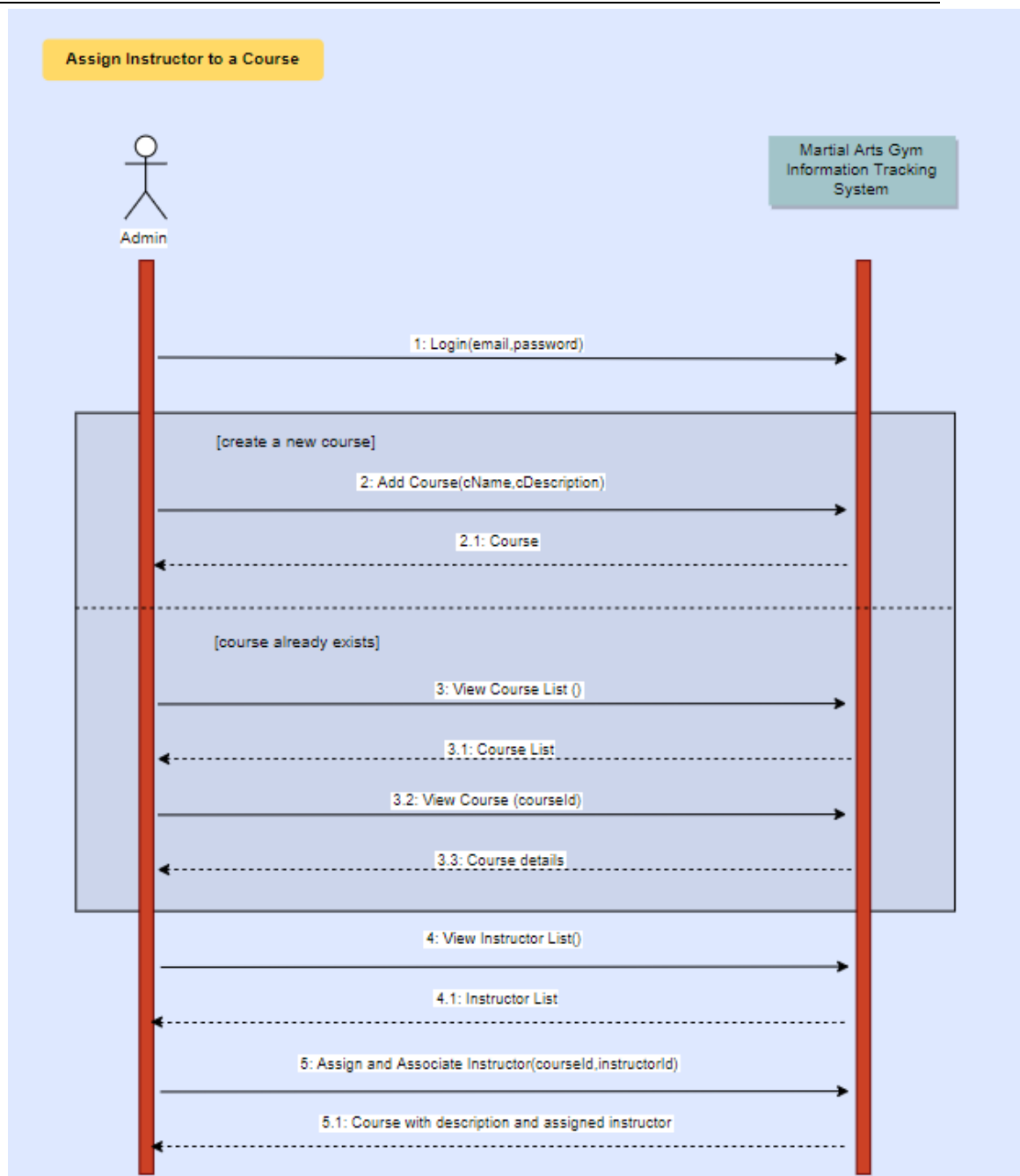
**Edit membership types** (refers to use case 3.2.4.20)



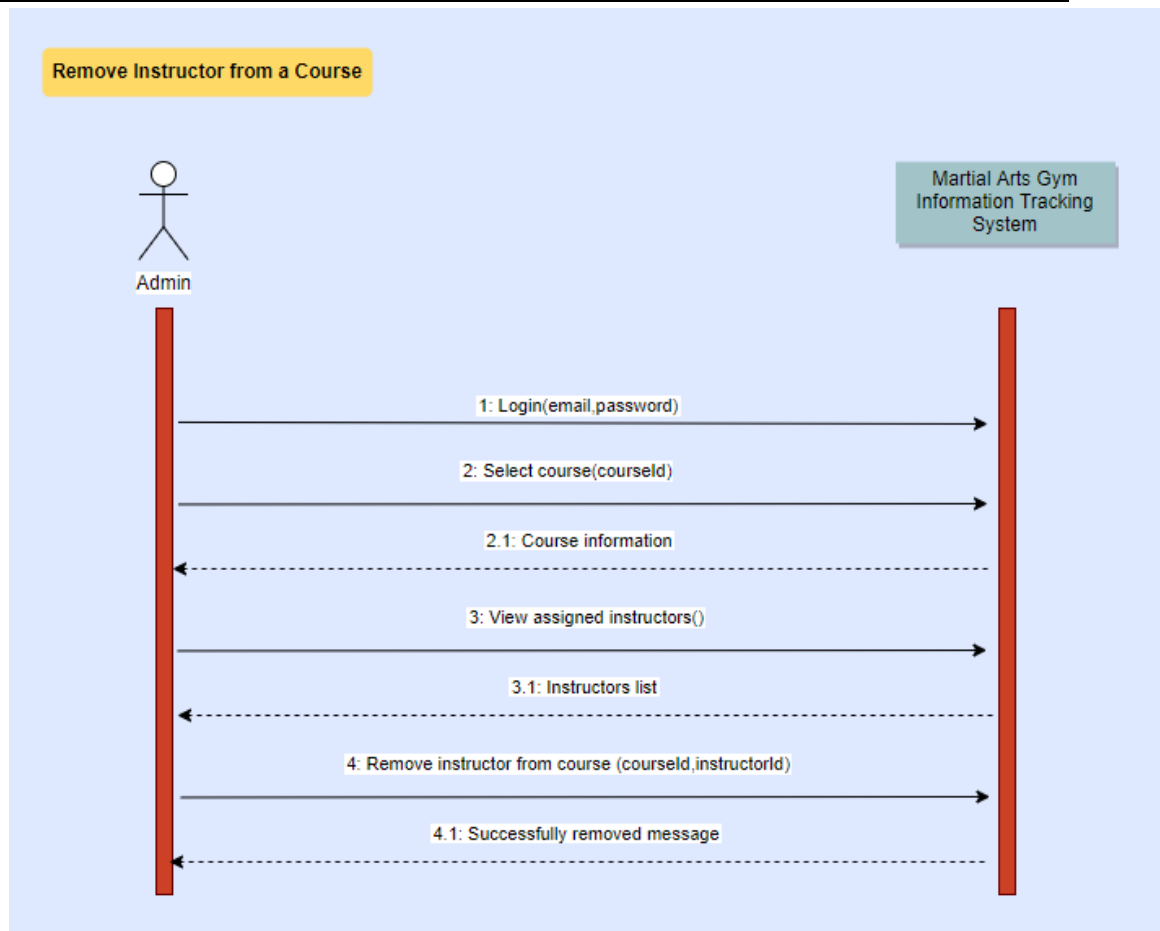
**Remove membership types** (refers to use case 3.2.4.21)



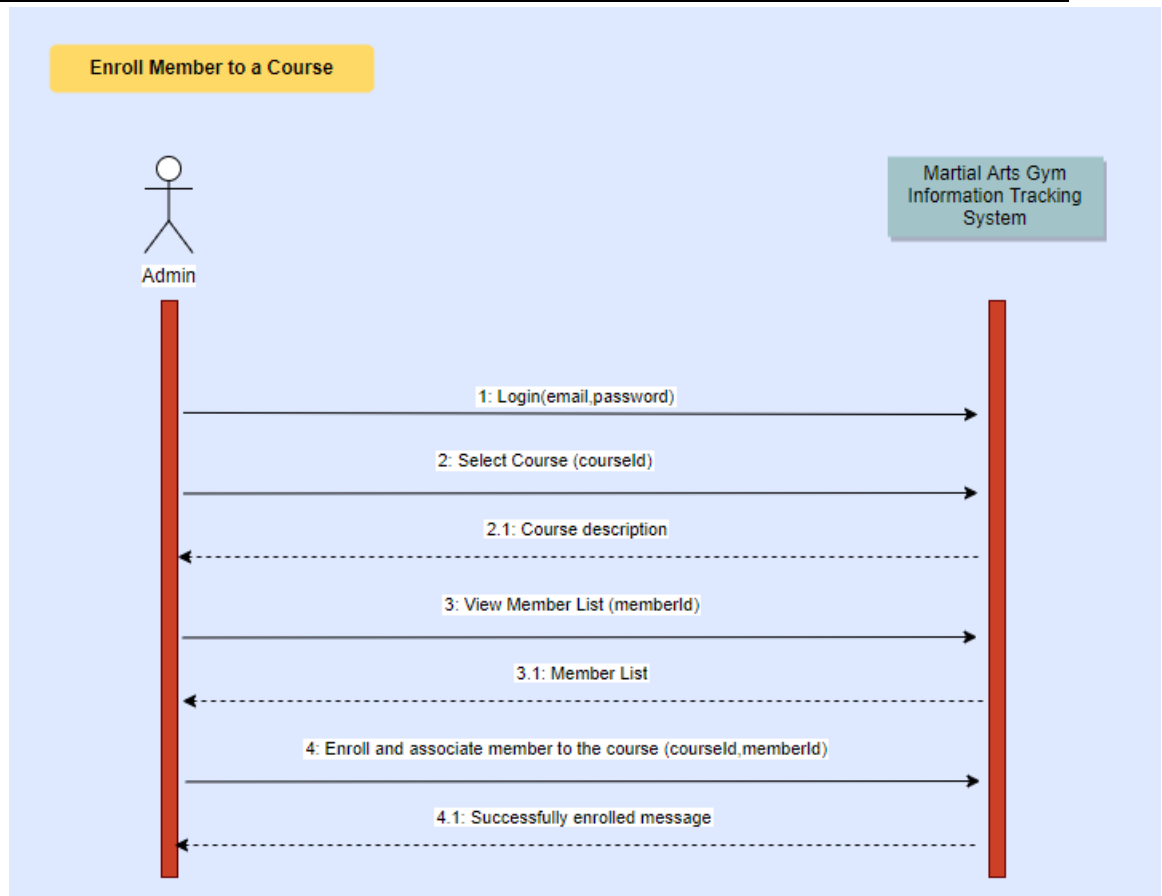
**Assign instructor to a course** (refers to use case 3.2.4.22)



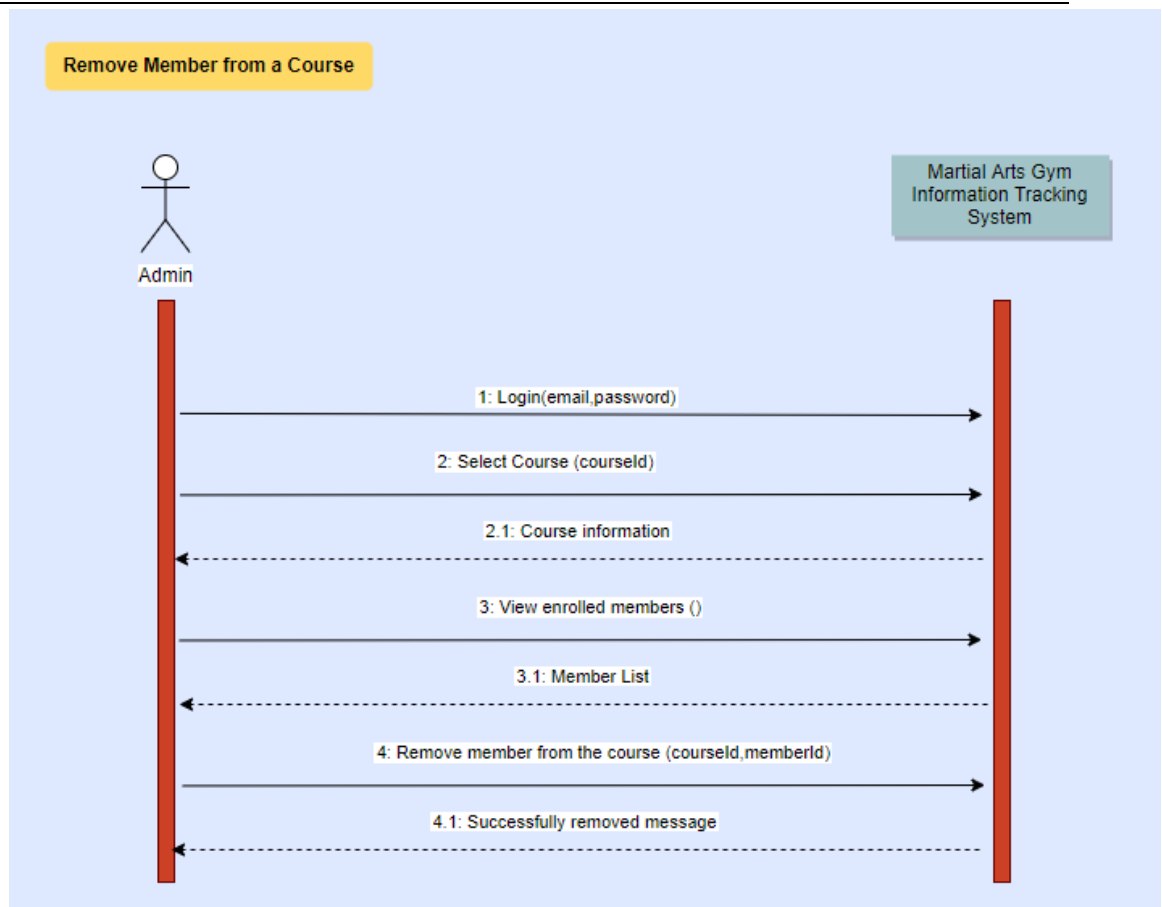
**Remove instructor from a course** (refers to use case 3.2.4.23)



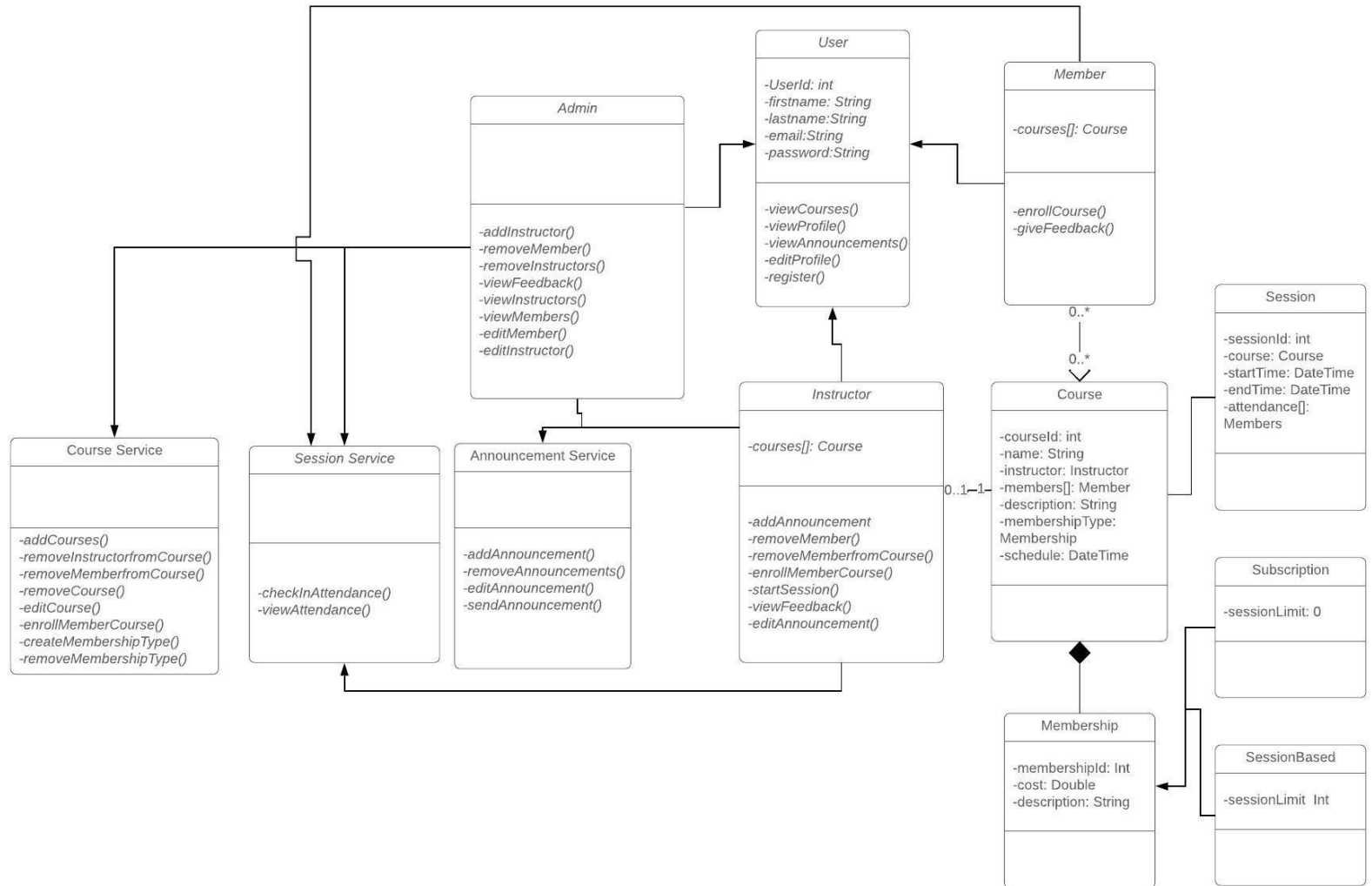
**Enroll members to a course** (refers to use case 3.2.4.24)



**Remove members from a course** (refers to use case 3.2.4.25)



#### d. 3.3.4 UML Class Diagram

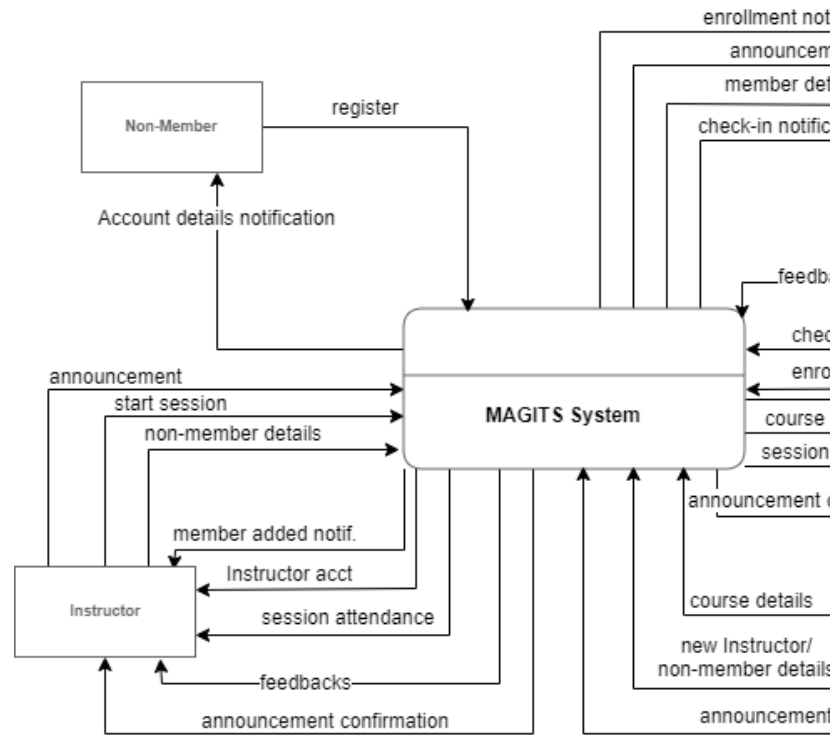




13.

## 14. 3.4 Process Modelling

## a. 3.4.1 Data Flow Diagram Ivl 0



i.

ii.

iii.

iv.

v.

vi.

vii.

viii.

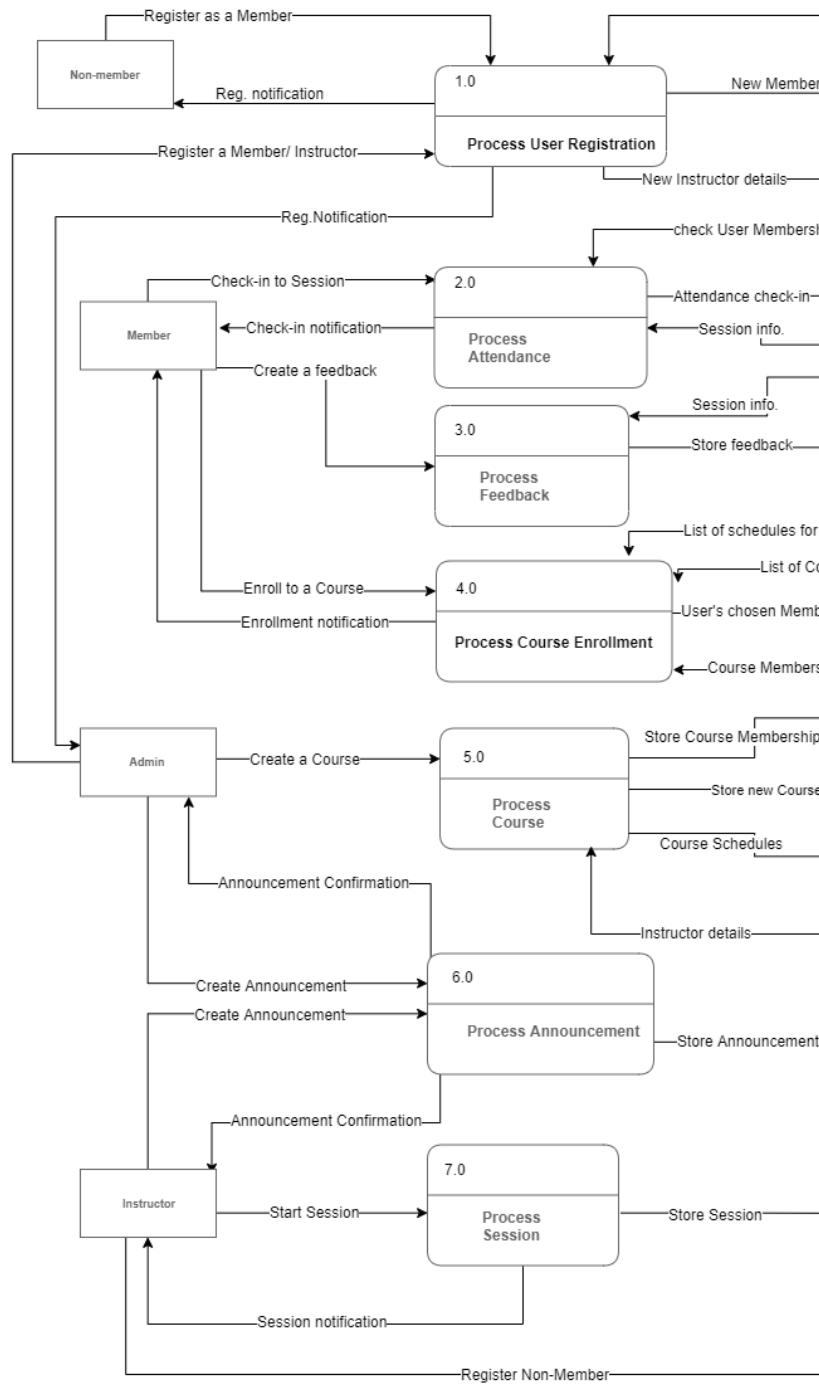
ix.

x.

xi.

xii.

## b. 3.4.2 Data Flow Diagram Ivl 1



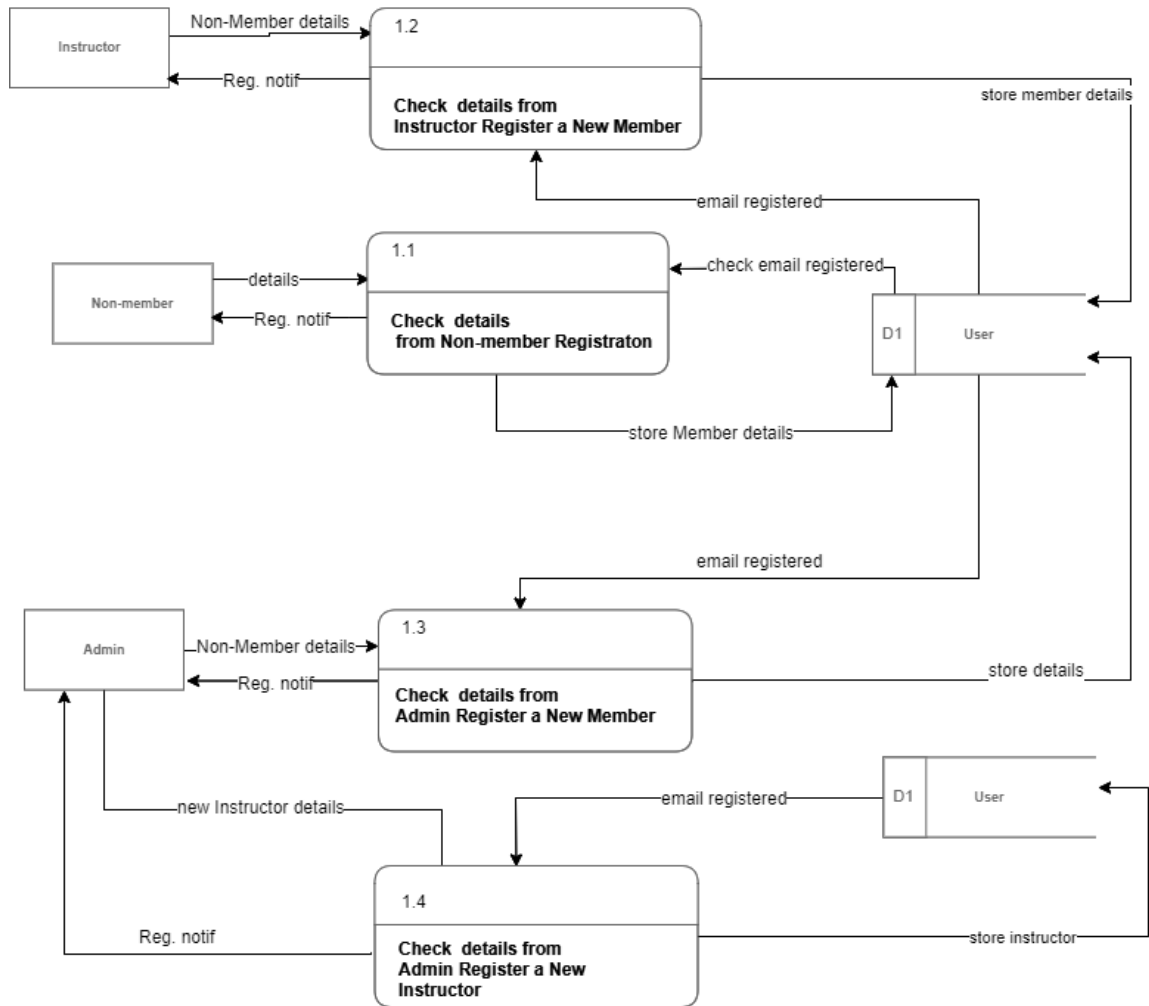
Data Flow Diagram: Context Ivl 1 MAGITS System

*i.**ii.**iii.*

### c. 3.4.3 Data Flow Diagram Ivl 2

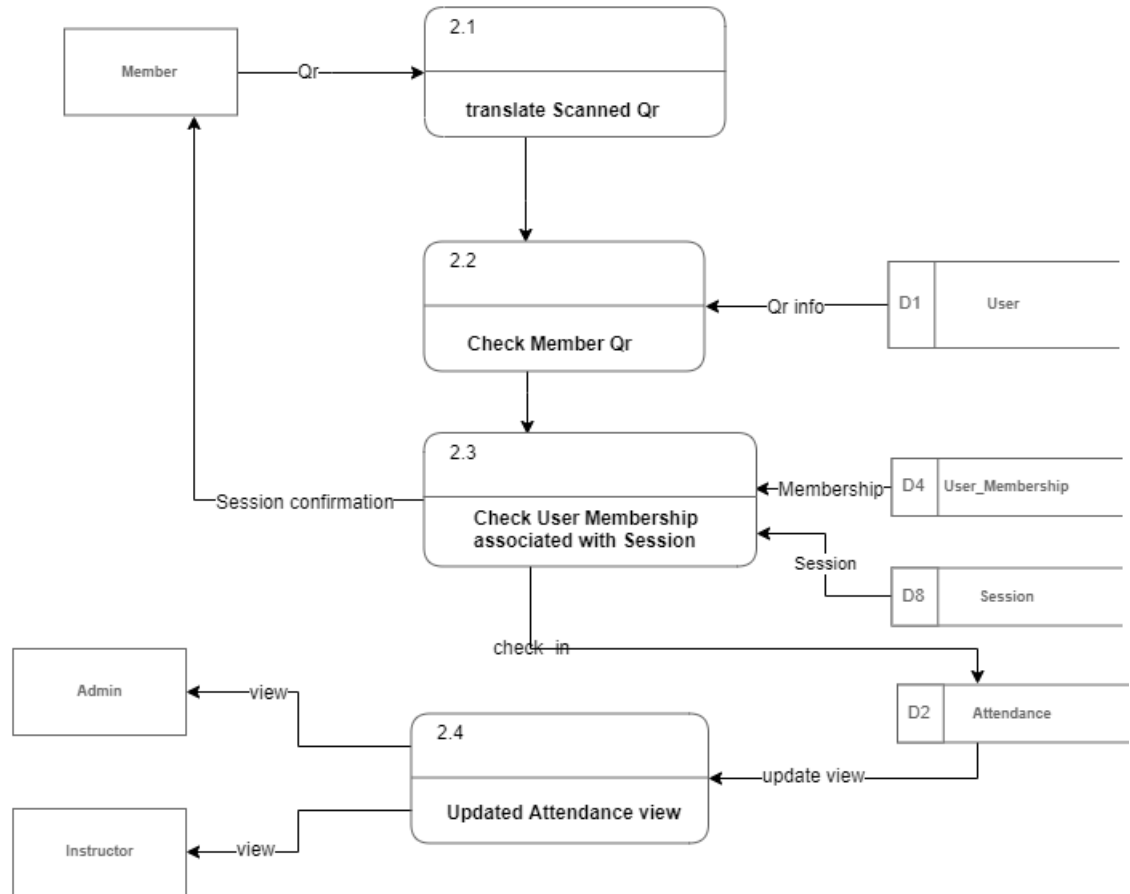
#### 3.4.3.1 Process User Registration

## Process User Registration



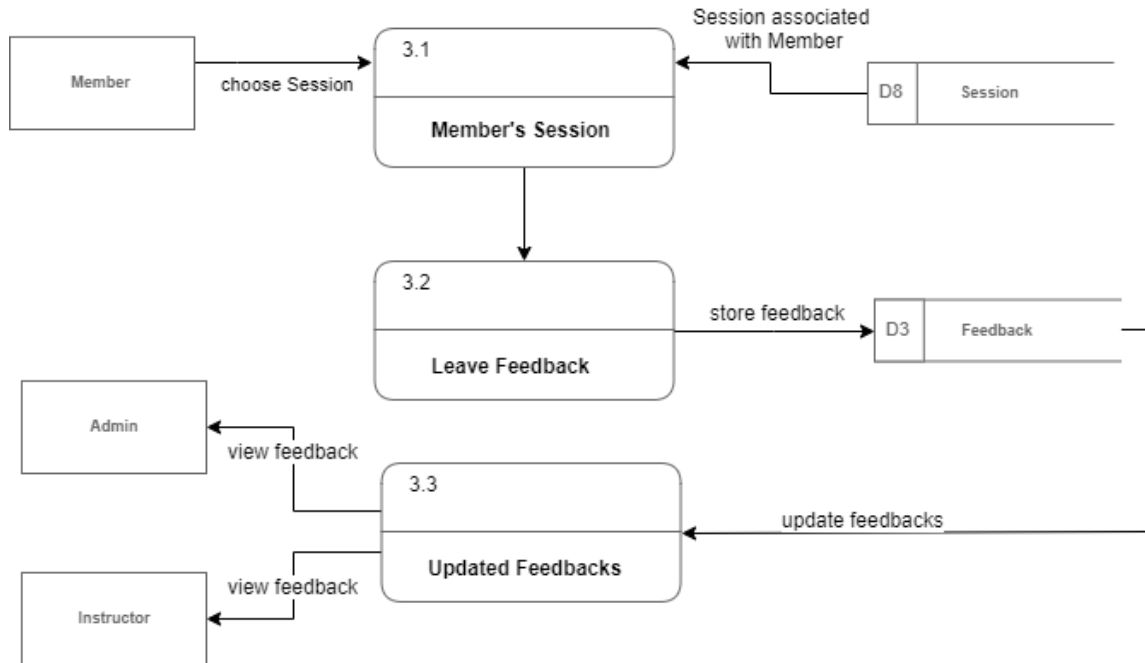
## 3.4.3.2 Process Attendance

## Process Attendance



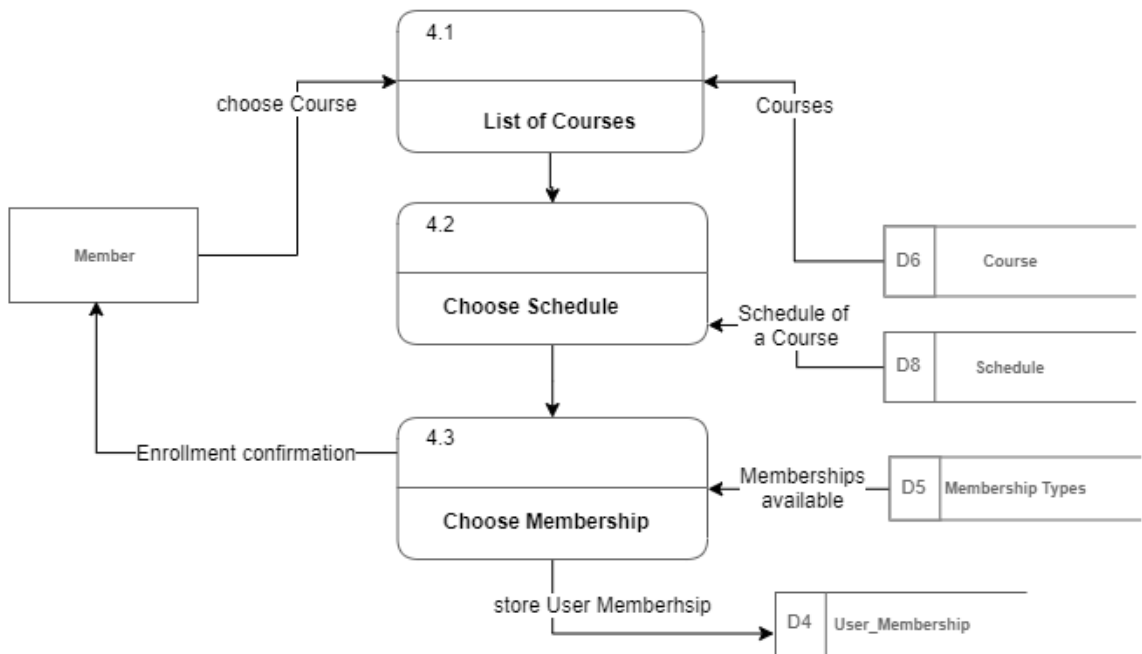
## 3.4.3.3 Process Feedback

## Process Feedback



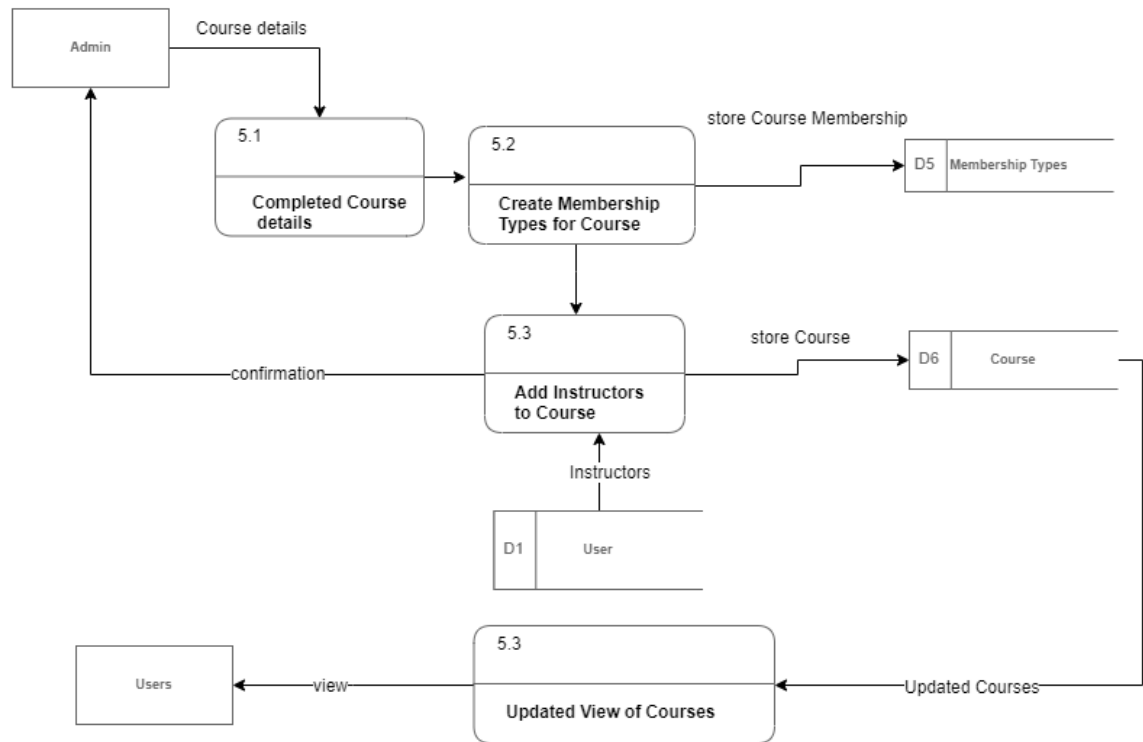
## 3.4.3.4 Process Course Enrollment

## Process Course Enrollment



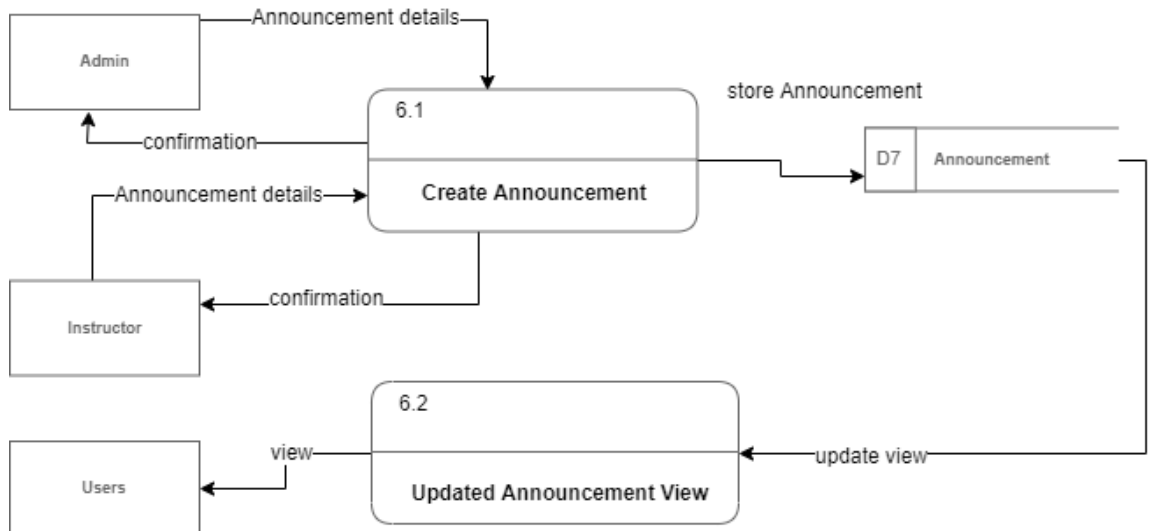
## 3.4.3.5 Process a Course

## Process a Course

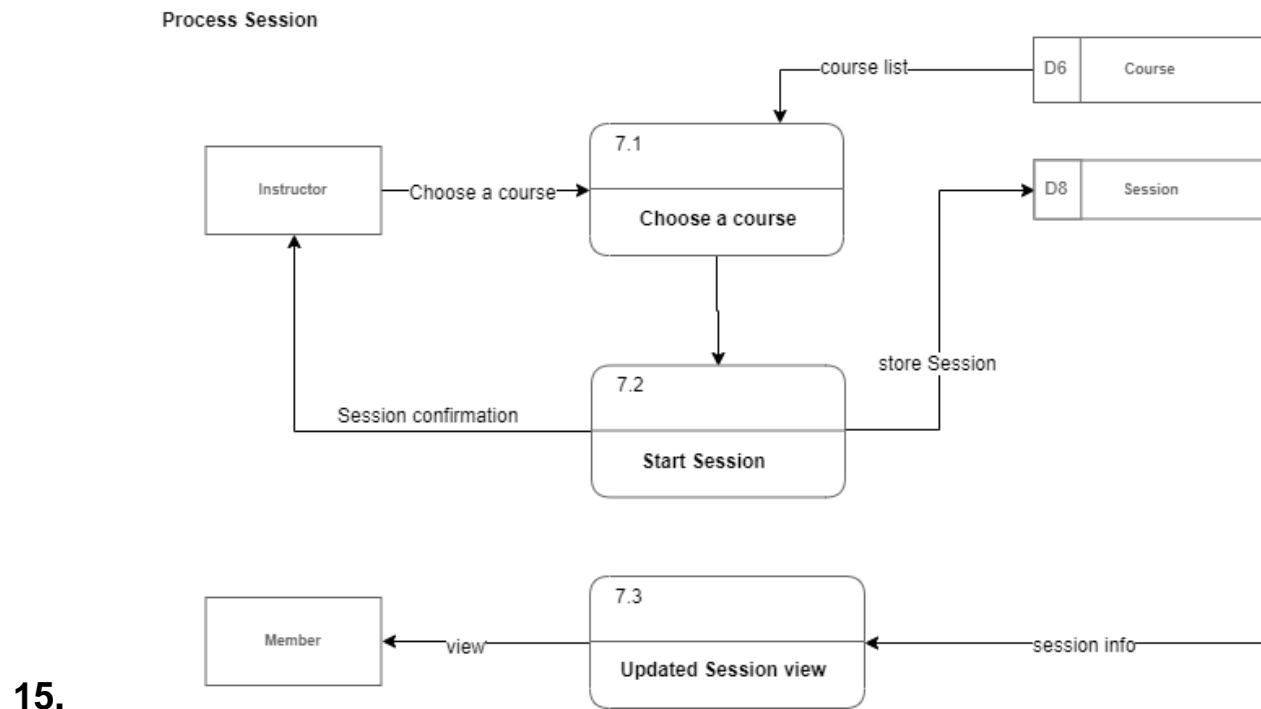


## 3.4.3.6 Process Announcement

## Process Announcement



## 3.4.3.7 Process Session



## 16. 4.0 Non-Functional Requirements

### Performance

- The website should be able to handle at least 500 concurrent users without affecting its performance.

### Reliability

- The camera responsible for scanning QR codes must be able to identify images clearly.
- The MAGITS system is required to have 90% reliability.

### Security

- Admin must never share their passwords since the admin account can affect the whole data integrity of the system.

### Consistency

- The course description must be entered by the Admin accurately and in detail in accord with the Gym's standards.
- The website must have a consistent UI aesthetic.

**Portability**

- The website must be able to perform on major web browsers such as Google Chrome and Firefox.
- The system must be able to run on Windows Operating Systems (7,8,10).

**Usability**

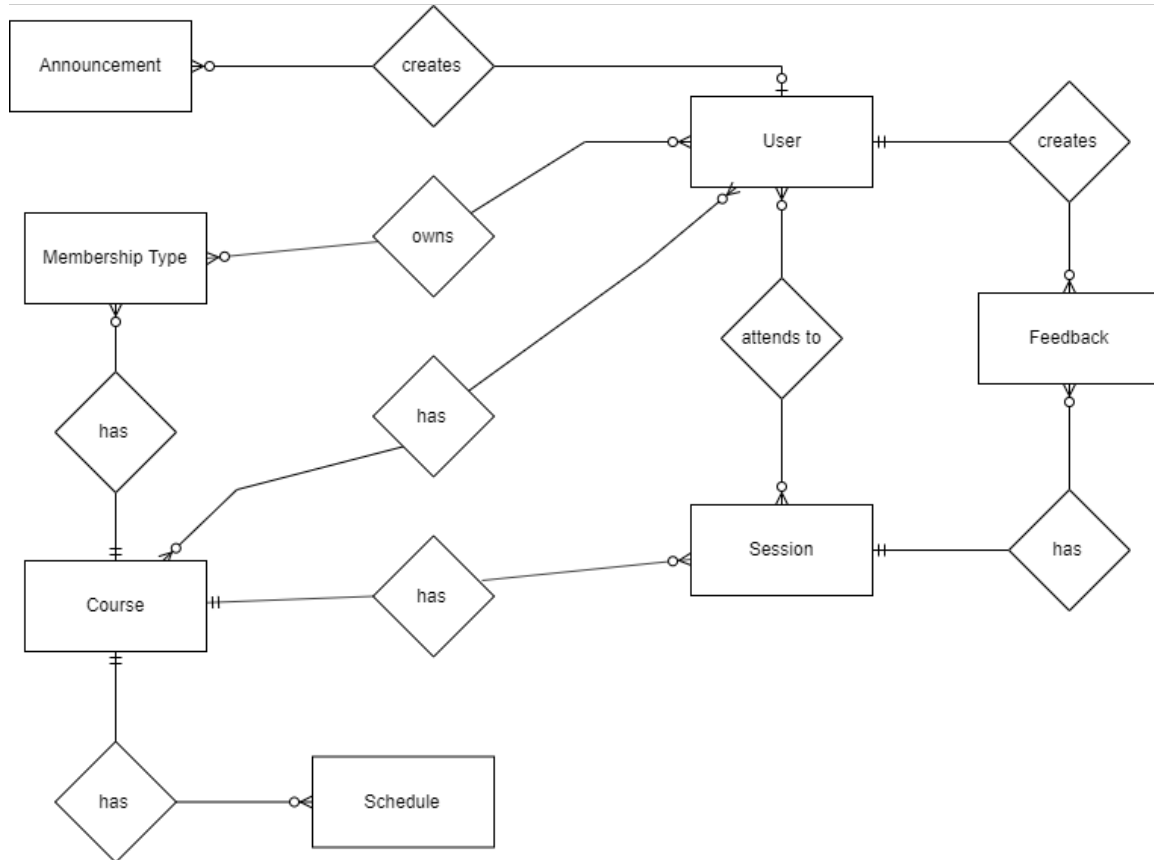
- The website UI must be designed such that 99% of all customers will interact with it with ease.

**Scalability**

- The system must be able to accommodate future enhancements.



## 17. 5.0 Logical Database Requirements



### 5.1 Main Entities:

#### 1. User Data Entity

Data Item	Type	Description	Comment
First Name	Text	first name of the User	
Last Name	Text	last name of the User	
email	Text	email address of the User	

password	Text	password of the User	
----------	------	----------------------	--

## 2. Role Data Entity

Data Item	Type	Description	Comment
Role Name	Text	name of role	
Description	Text	details of role	

## 3. Course Data Entity

Data Item	Type	Description	Comment
Course Name	Text	name of the Course	
Description	Text	extra details of the Course	

## 4. Session Data Entity

Data Item	Type	Description	Comment
-----------	------	-------------	---------

Course	Pointer	Course Entity	May have multiple similar course entities
Date	Text	Date/Time when the session was activated	
Start Time	Time	Time that the session started	
Duration	Integer	Duration of the session	

### 5. Membership Data Entity

Data Item	Type	Description	Comment
Course	Pointer	Course Entity	May have multiple similar course entities as courses may have one or two types of membership
Name	Text	name of the Membership	
Cost	Text	cost of Membership	
Type of Membership	Text	type of the Membership	if the member's type of membership is 'subscription' his/her session limit would be 0

Session Limit	Text	defines the number of sessions allowed for a membership	Depends on membership type
---------------	------	---	----------------------------

## 6. Announcement Data Entity

Data Item	Type	Description	Comment
Content	Text	Contents of Announcement	

## 7. Feedback Data Entity

Data Item	Type	Description	Comment
Session	Pointer	Session Entity	May have multiple similar session entities
Member	Pointer	Member Entity	May have multiple similar member entities

## 8. Schedule Data Entity

Data Item	Type	Description	Comment
-----------	------	-------------	---------

Course	Pointer	Session Entity	May have multiple similar course entities as each course may have more than one schedule
Day	Text	Day of the week	
Start Time	Time	Start time of the schedule	
End Time	Time	End time of the schedule	

## 5.2 Associative Entities:

### 1. User\_Role

Data Item	Type	Description	Comment
User	Pointer	User Entity	May have multiple similar users for various roles
Role	Pointer	Role Entity	May have multiple similar roles for various users

### 2. Attendance

Data Item	Type	Description	Comment
User	Pointer	User Entity	May have multiple similar users

Session	Pointer	Session Entity	May have multiple similar sessions
Date	Date/Time	Date and time attendance was taken	

## 3. Course\_User

Data Item	Type	Description	Comment
Course	Pointer	Course Entity	May have multiple similar courses
User	Pointer	User Entity	May have multiple similar users

## 4. User\_Membership

Data Item	Type	Description	Comment
User	Pointer	User Entity	May have multiple similar user entities
Membership	Pointer	Membership Entity	May have multiple similar membership entities





## 18. 6.0 Other Requirements

The system uses an api such as <https://www.npmjs.com/package/qr-code-scanner>. This is a simple qr-code-scanner that will allow you to open the user's camera, scan it and match a pattern or read the string out of the QR Code.

## 19. 7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
--------------	------	-----------	------

Developer	Aldrin John Jacildo		10/07/2019
Developer	Sir Angel Naguit		10/07/2019
Product Owner/ Developer	Francis Victa		10/07/2019
Team Lead	Maria Lilian Yang		10/07/2019