



SOFTWARE ENGINEERING

ITERATION - 2

Group Members:

Kalsoom Tariq (21i-2487)

Haris Sohail (21i-0531)

Aiman Safdar (21i-0588)

April 7th, 2024

Table of Contents

Project Module Description.....	3
Module Title.....	3
Introduction to Module.....	3
Pre-requisites.....	3
Basic WorkFlow.....	3
User Stories.....	4
US 1.1: View Society.....	4
US 3.1: View Society Request.....	5
US 4: Search Society.....	5
US 5: Register Event.....	6
Structured Specifications.....	6
SS 1.1: View Society.....	6
SS 3.1: View Society Request.....	7
SS 4: Search Society.....	8
SS 5: Register Event.....	9
Design.....	10
Activity Diagram.....	10
AD1: Society Registration.....	10
UD2: Search Societies.....	12
UD3: Event Registration.....	13
UseCase Diagram.....	14
Sequence Diagrams.....	15
SD1: Society Registration:.....	15
SD2: Search Society.....	16
SD3: Register Event:.....	17
Class Diagram.....	18
Architecture.....	19
Scrum Board.....	20
Work division.....	22
Implementation.....	23
Technical Debt:.....	23
Event Registration:.....	25
BurnDown Chart.....	28

Project Module Description

Module Title

Societies and Events

Introduction to Module

This module enables society members to effortlessly register events. Users can efficiently search for societies, enhancing the functionality and accessibility of the society management system.

Pre-requisites

Prior to the implementation of the aforementioned module, our team had to rethink the design of the system for the following reasons:

- 1) To integrate an object oriented approach in the system.
- 2) To integrate a meaningful architecture in the system.

The relevant documentation of the design is available on the Trello board.

Basic WorkFlow

Starting from the Log-In Screen, a student shall be validated and redirected to the CampusNex Main Interface.

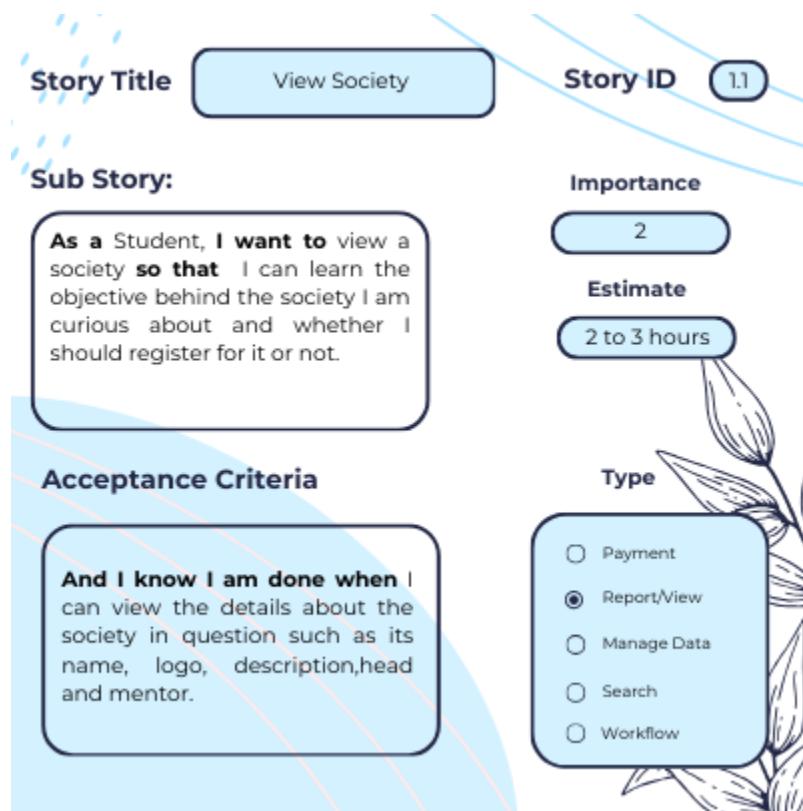
1. The user will be able to search for their relevant society, by typing keywords in the search bar on the societies page . The societies page will

show the matching societies where user can explore and engage with their relevant societies

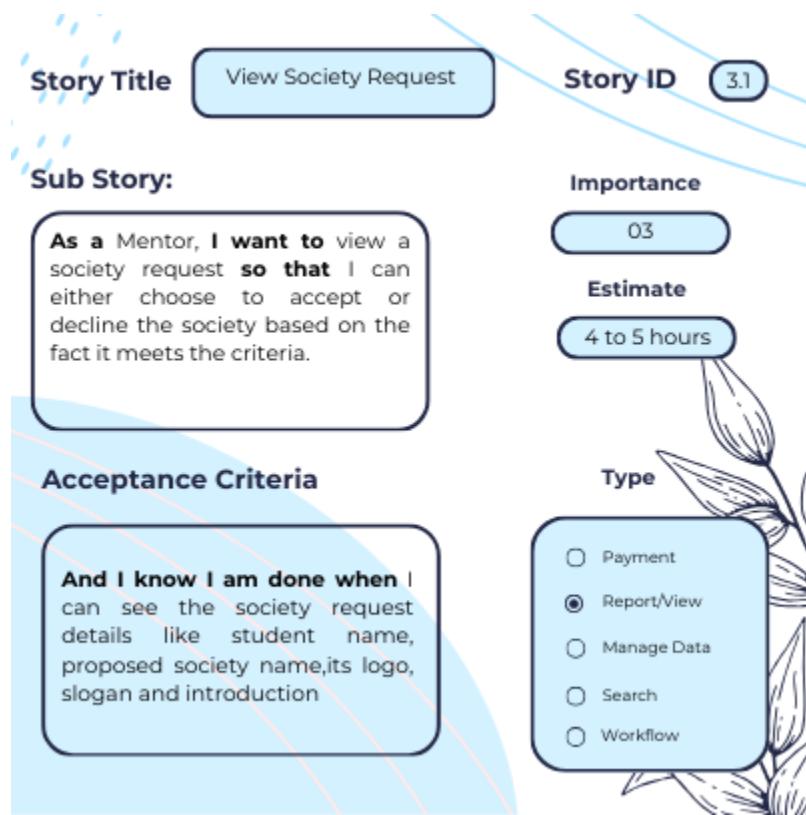
2. The student shall be able to register events. On the 'Events' page, the student will enter event information through a form and request an event registration. This request will be accepted/rejected by the mentor of the society. The status of the request will be updated to the student. If accepted the events page will be updated with the newly registered event.

User Stories

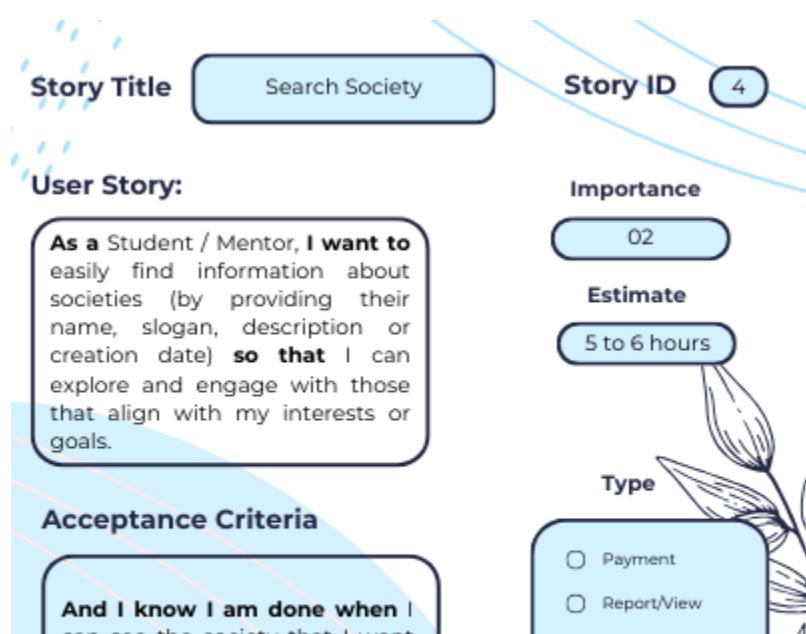
US 1.1: View Society



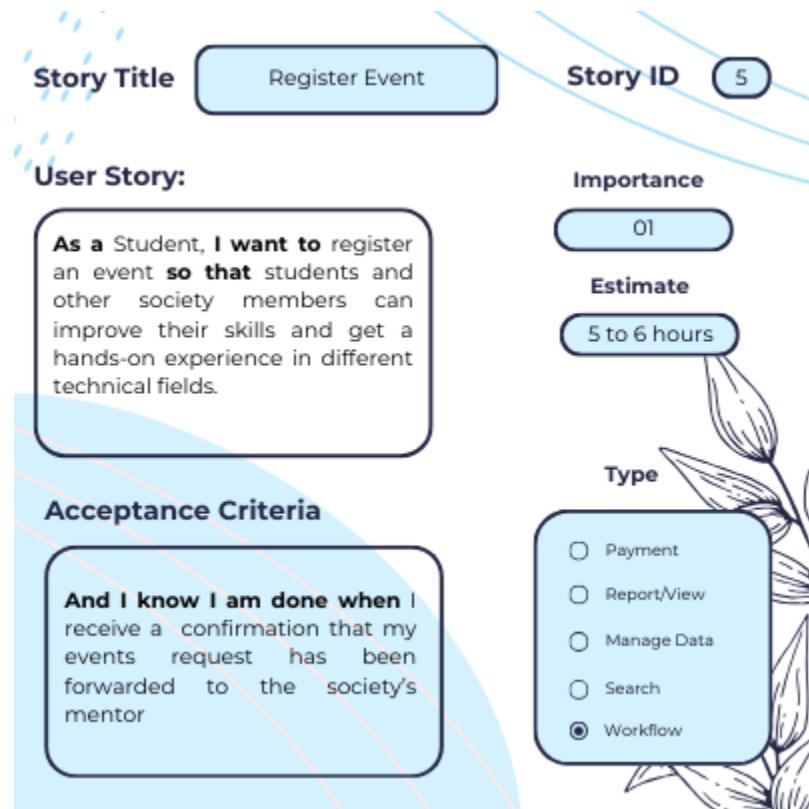
US 3.1: View Society Request



US 4: Search Society



US 5: Register Event



Structured Specifications

SS 1.1: View Society

	Details
Description	Students can explore and gather information about the society they are interested in.

Inputs	View Society Button
Source	Database
Outputs	Particulars about the society
Destination	Society Page
Action	<ol style="list-style-type: none"> 1. The Student selects the society they are interested in viewing. 2. The Student shall be shown a window displaying the Society details.
Requirements	<ul style="list-style-type: none"> • Student must be enrolled in the institution • Society must be registered and in active status
Pre-condition	<ul style="list-style-type: none"> • Valid student details • Society must be visible on dashboard
Post-condition	Students have gained insight about the active and registered societies.
Side effects	None

SS 3.1: View Society Request

	Details
Description	Mentor can view the new society requests and choose to accept or decline them.
Inputs	View Request Button
Source	Database (Fetch Request Information)
Outputs	<ul style="list-style-type: none"> • Student name • New Society Name • Logo • Society Type(e.g academic, cultural etc)

	<ul style="list-style-type: none"> • A detailed introduction for the new society.
Destination	Society request tab.
Action	<ol style="list-style-type: none"> 1. Mentor proceeds to Request Tab. 2. Mentor observes the requests in the table presented on the tab. 3. Mentor can choose to accept or decline the society by clicking on the button present in the same row as the society requests. 4. Mentor will click on View Request Button 5. Society Request Tab will open and mentor will be able to view all details.
Requirements	Society request must be made by a student to the specific mentor
Pre-condition	<ul style="list-style-type: none"> • Mentor is logged into account • Request Tab is open • There is at least one request made for the concerned mentor.
Post-condition	Mentor shall have gained insights about the new society request.
Side effects	None

SS 4: Search Society

	Details
Description	A streamlined method that enables users to quickly discover university societies through real-time dynamic search functionality
Inputs	<p>Any one of the following</p> <ul style="list-style-type: none"> • Society name • Society slogan • Society description

	<ul style="list-style-type: none"> • Society creation date
Source	User input via the search box on the societies page
Outputs	One or many societies that match the search result dynamically populated on the societies page
Destination	The societies page showing the matching societies where user can explore and engage with their relevant societies
Action	<ol style="list-style-type: none"> 1. User clicks on the search box 2. User types keywords of society (name, description, slogan or creation date) 3. System searches the relevant societies in the database. 4. System dynamically updates the societies page with the searched societies
Requirements	<ul style="list-style-type: none"> • There is at least one society registered that matches the keywords.
Pre-condition	<ul style="list-style-type: none"> • User is typing only the relevant keywords (name, slogan, description or creation date)
Post-condition	<ul style="list-style-type: none"> • User is presented with the relevant societies • User can explore and engage with the search results
Side effects	None

SS 5: Register Event

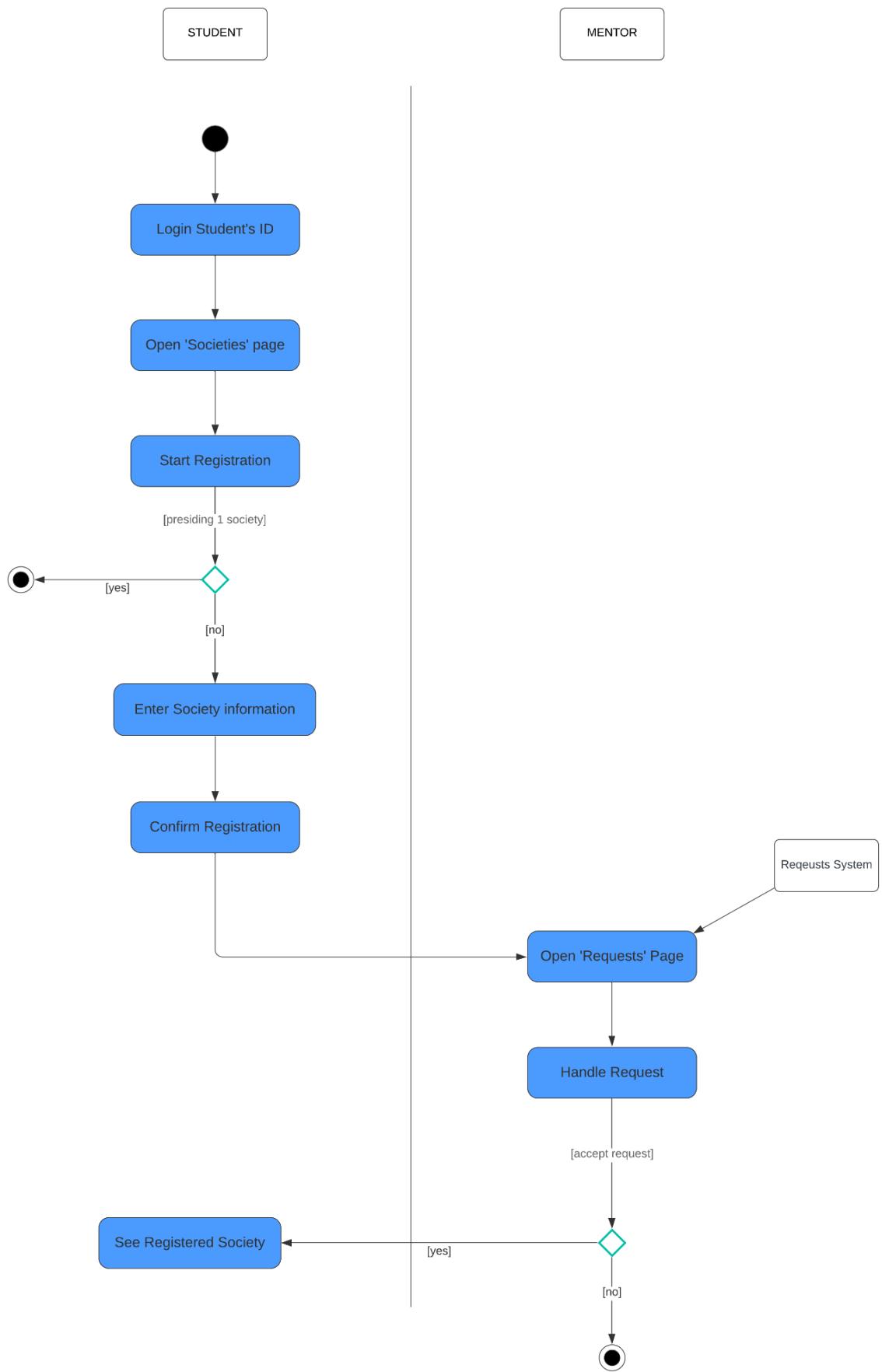
	Details
Description	Method that allows members of a society to register an event officially.
Inputs	<ul style="list-style-type: none"> • Event Title • Society Name • Event Date And Time • Event Location

	<ul style="list-style-type: none"> • Event Description • Relevant Image [Optional]
Source	Student
Outputs	A Confirmation Message that event has been sent for approval to the mentor of society
Destination	Student Events Page
Action	<ol style="list-style-type: none"> 1. The student Navigates to the Events Page. 2. The student clicks the “Register a New Event Button” 3. The student enters the details as outlined in the Inputs in the required fields. 4. The student clicks the Register Button. 5. A confirmation message appears that the request for an event registration has been forwarded to the society mentor.
Requirements	The student is already a member of the society.
Pre-condition	<p>The student has logged in and navigated to the Events Page.</p> <p>The relevant Society for event registration exists.</p>
Post-condition	<p>The database is updated with the new event Info.</p> <p>The Event Request appears on the society mentor's Dashboard.</p>
Side effects	None

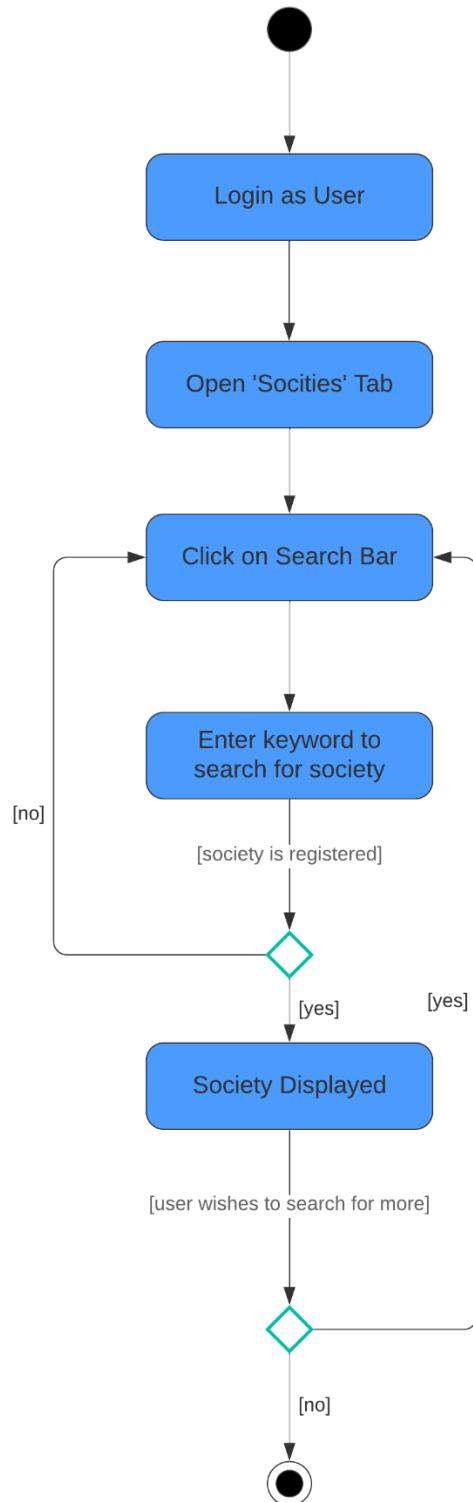
Design

Activity Diagram

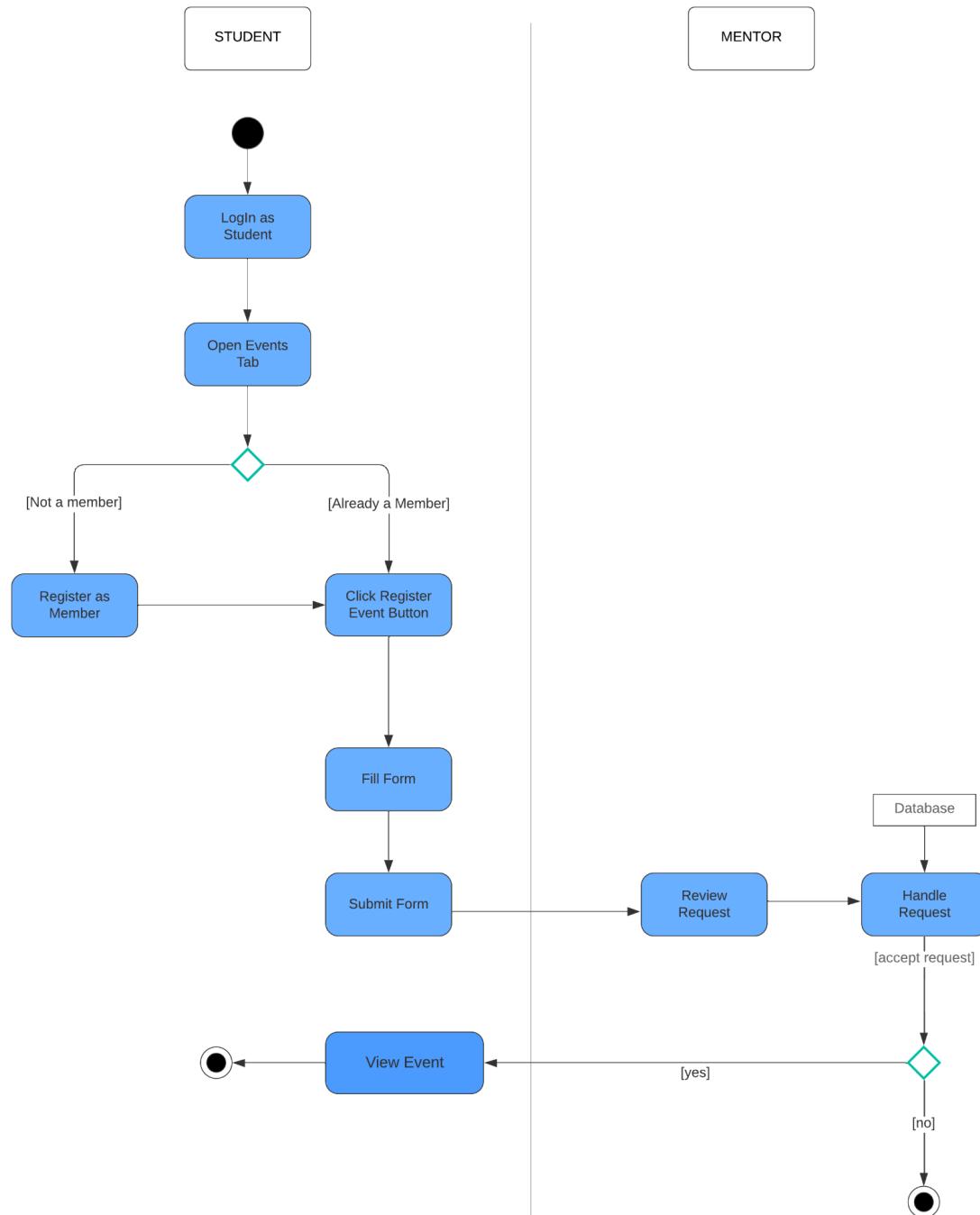
AD1: Society Registration



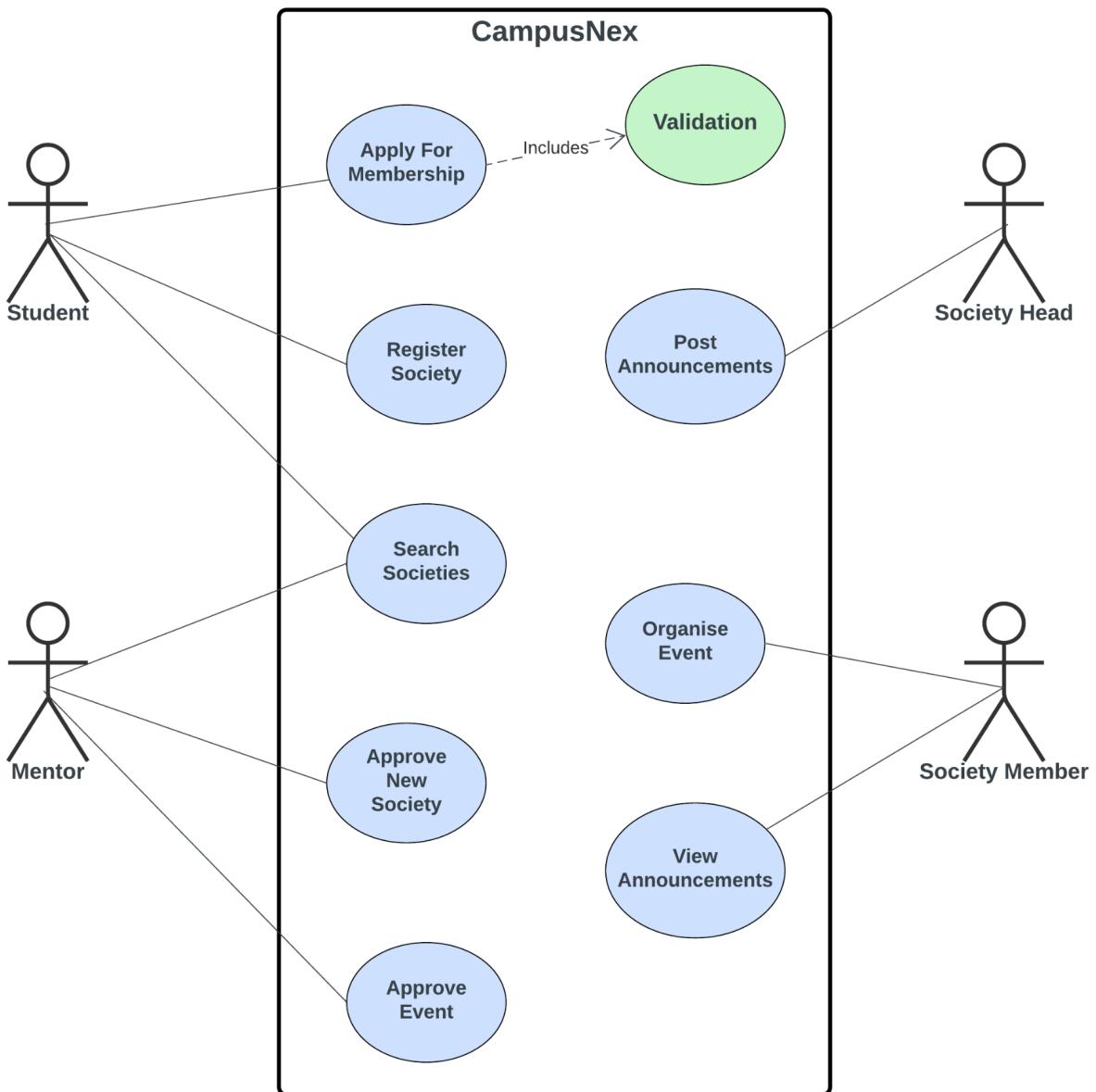
UD2: Search Societies



UD3: Event Registration

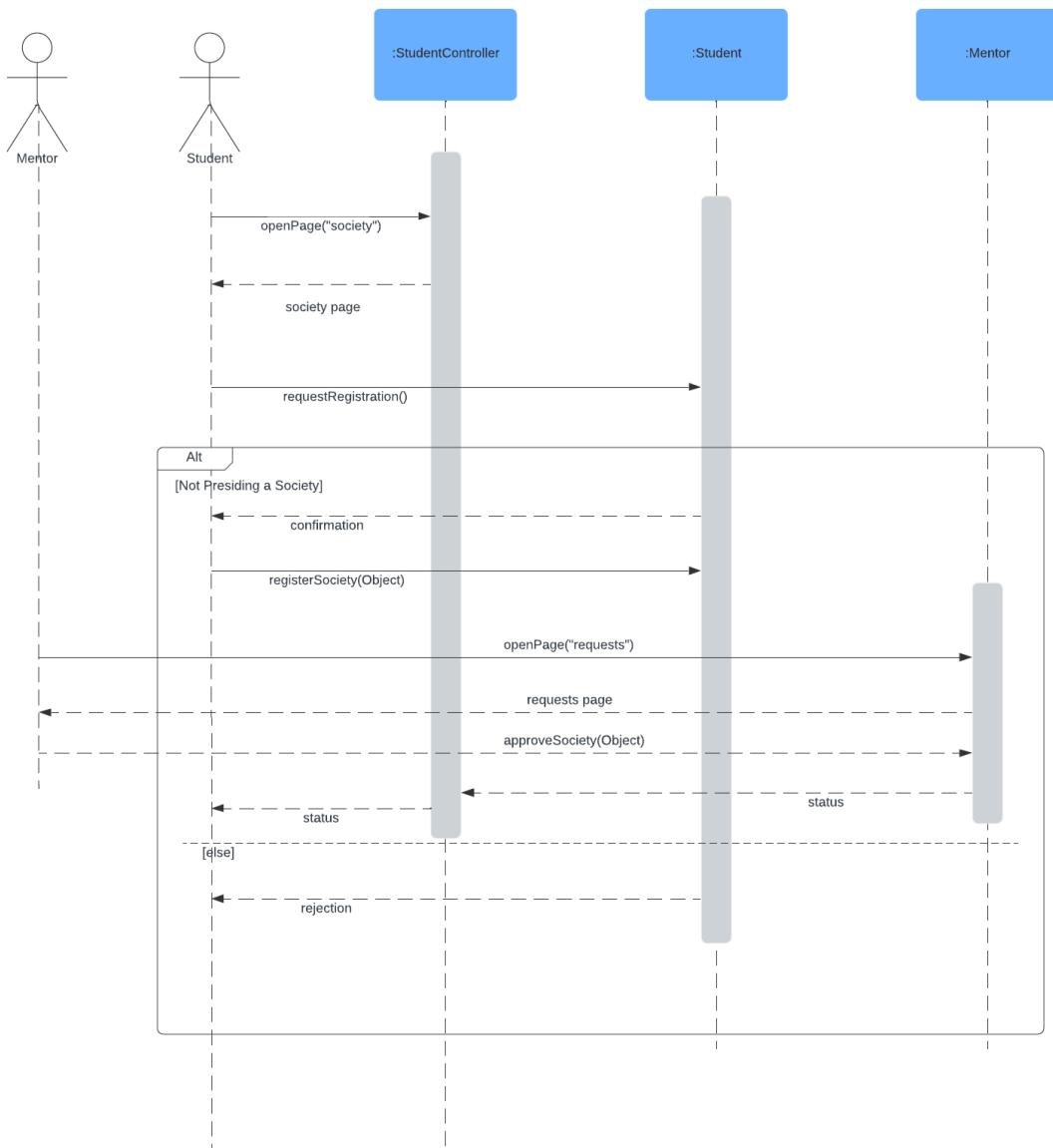


UseCase Diagram

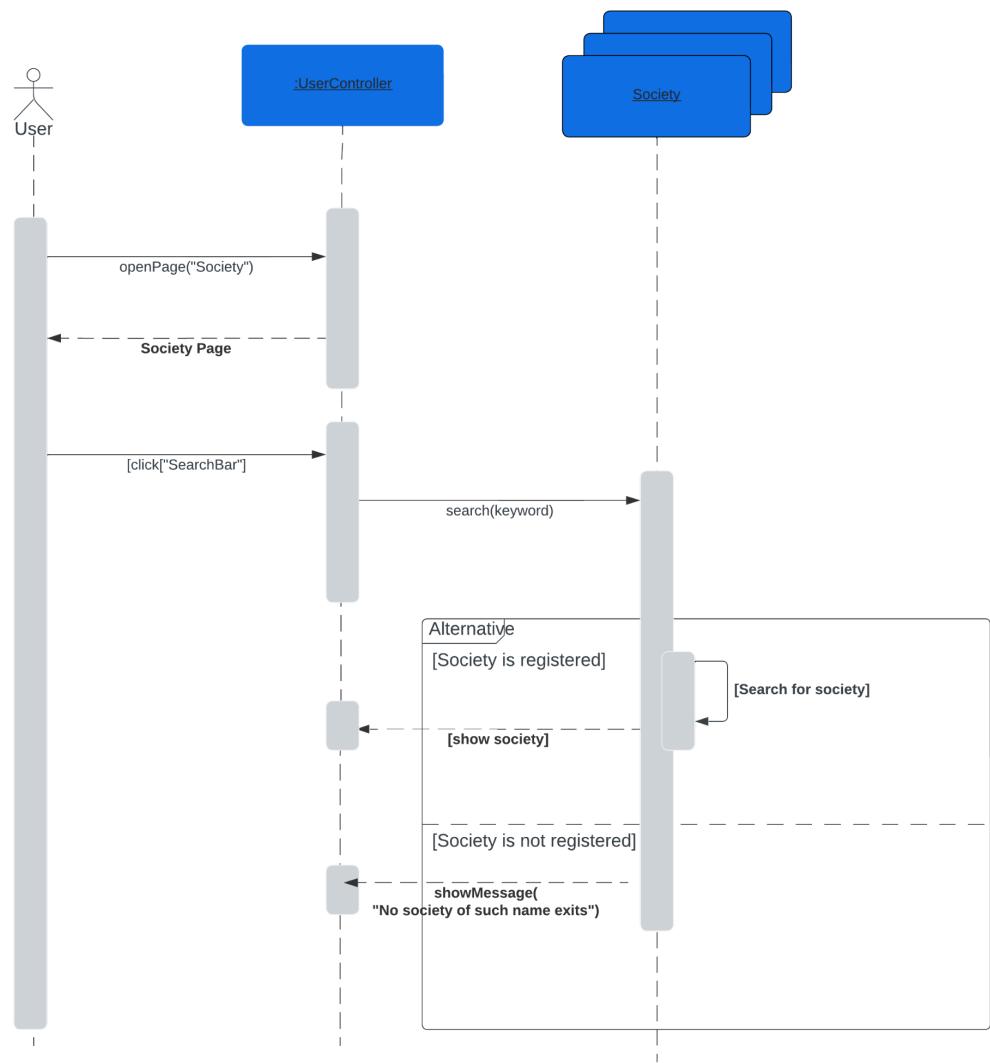


Sequence Diagrams

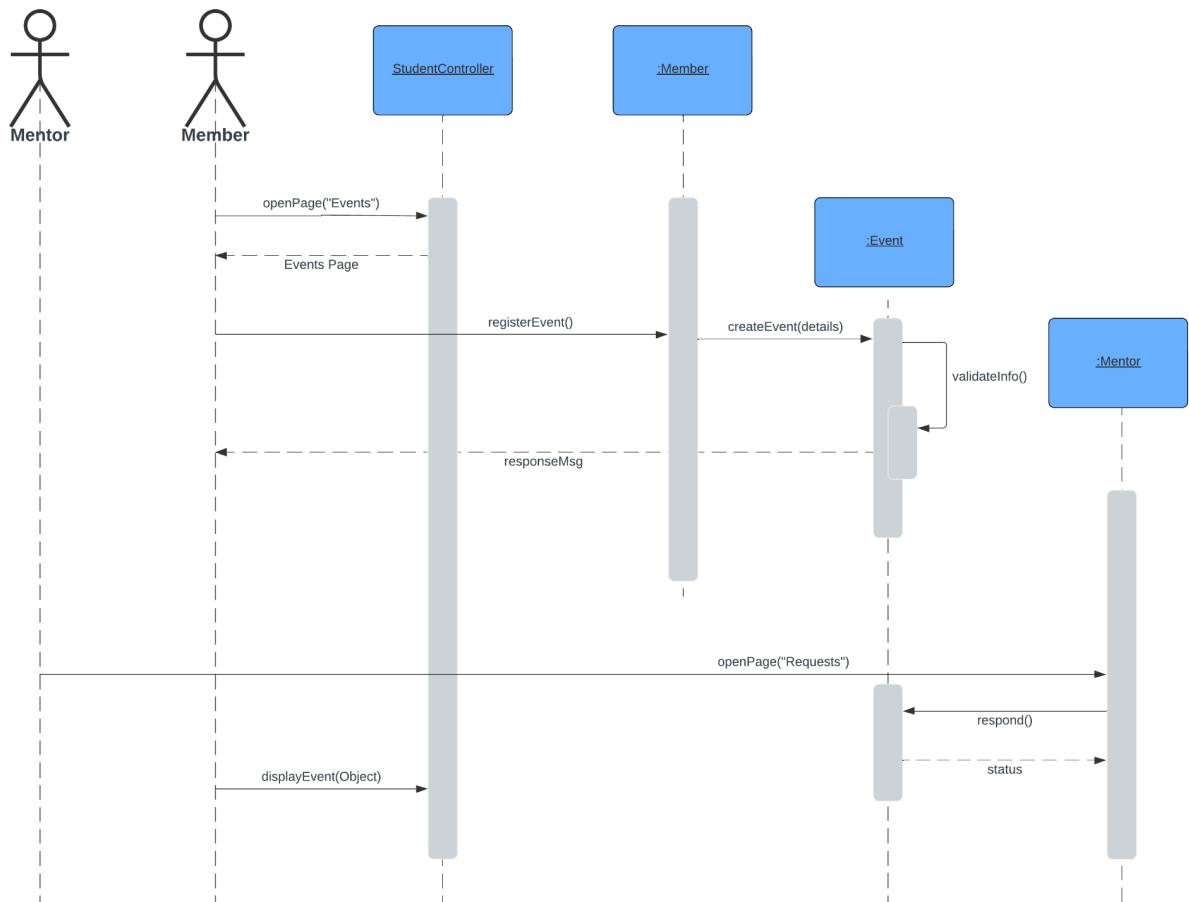
SD1: Society Registration:



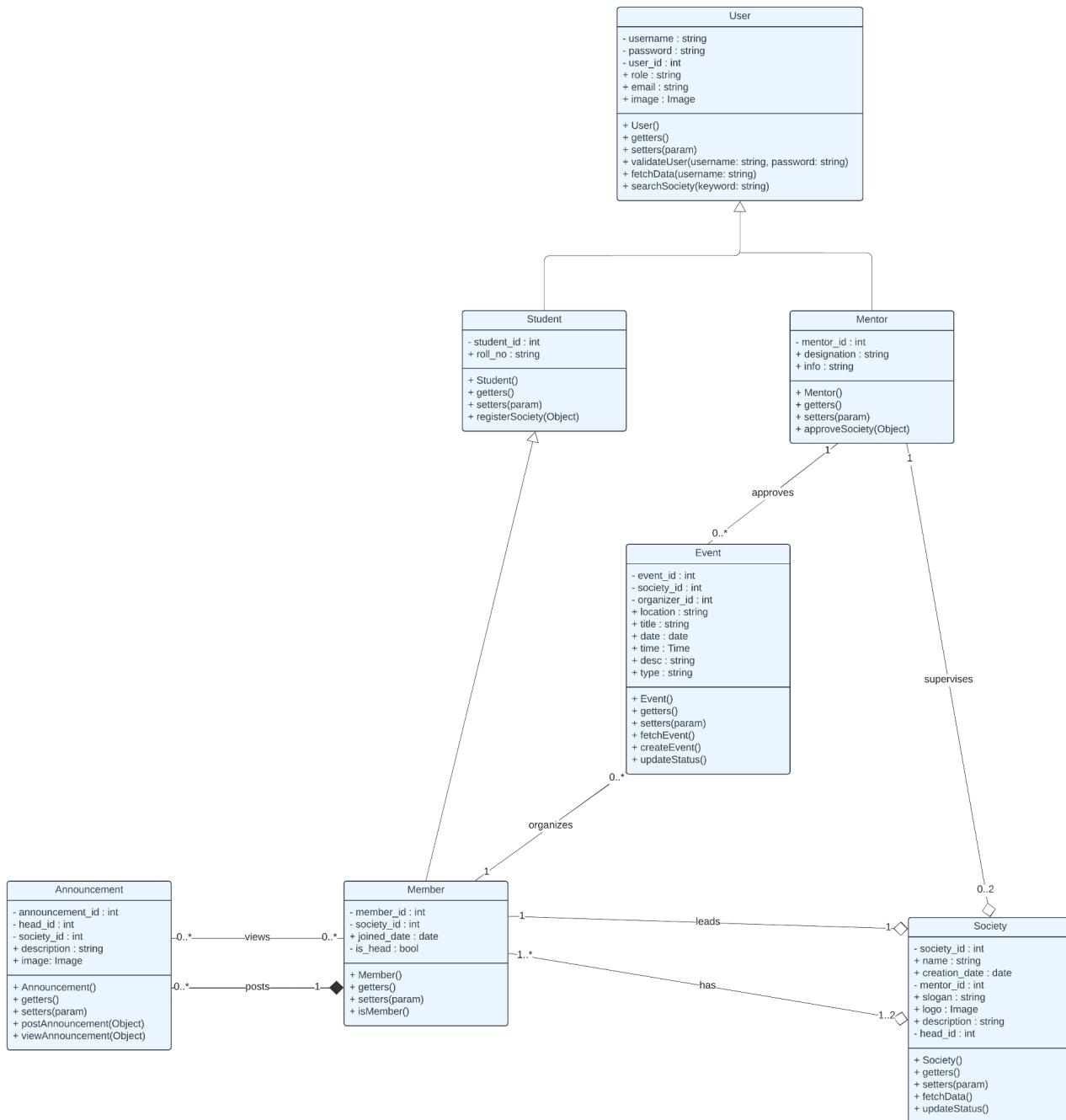
SD2: Search Society



SD3: Register Event:



Class Diagram



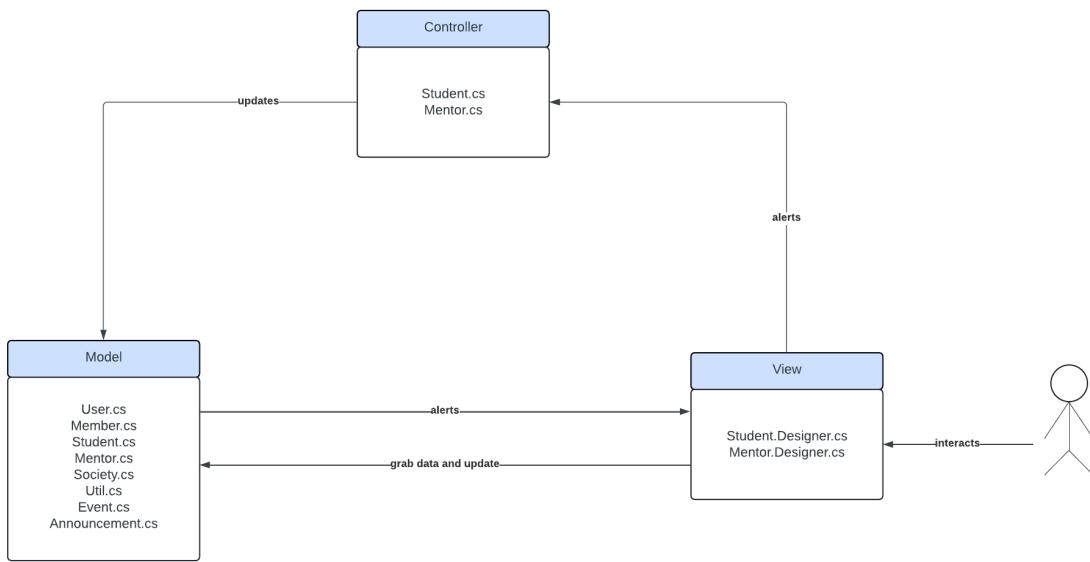
Architecture

The project is implemented using the Model-View-Controller (MVC) architecture to facilitate the maintainability and efficiency of the application. This pattern helps in segregating the codebase into three layers. This separation ensures that the application is maintainable and scalable.

The primary reason to implement this architecture was to address performance issues. A significant lag was observed after logging in, when only one class per user (student or mentor) was invoked.

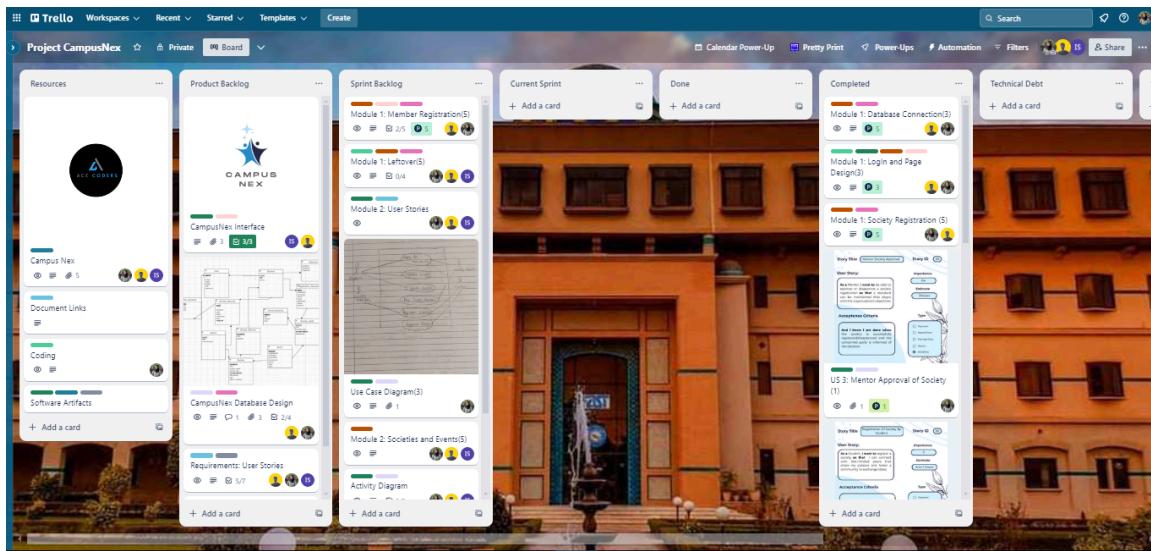
This bottleneck was solved by introducing the following model classes:

- *Announcement*
- *Event*
- *Member*
- *Mentor*
- *Society*
- *Student*
- *User*
- *Util*

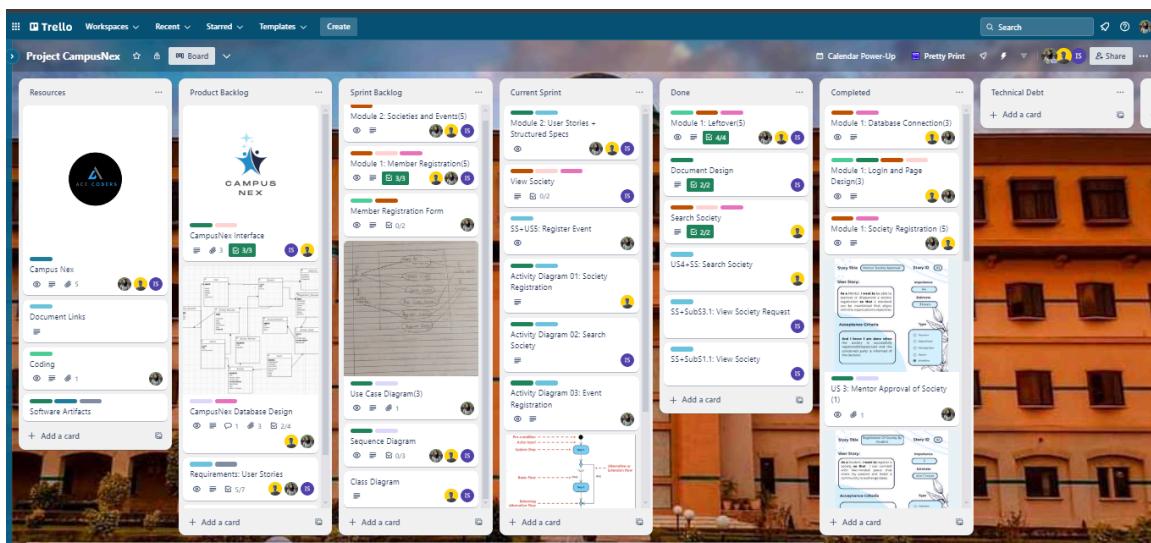


Scrum Board

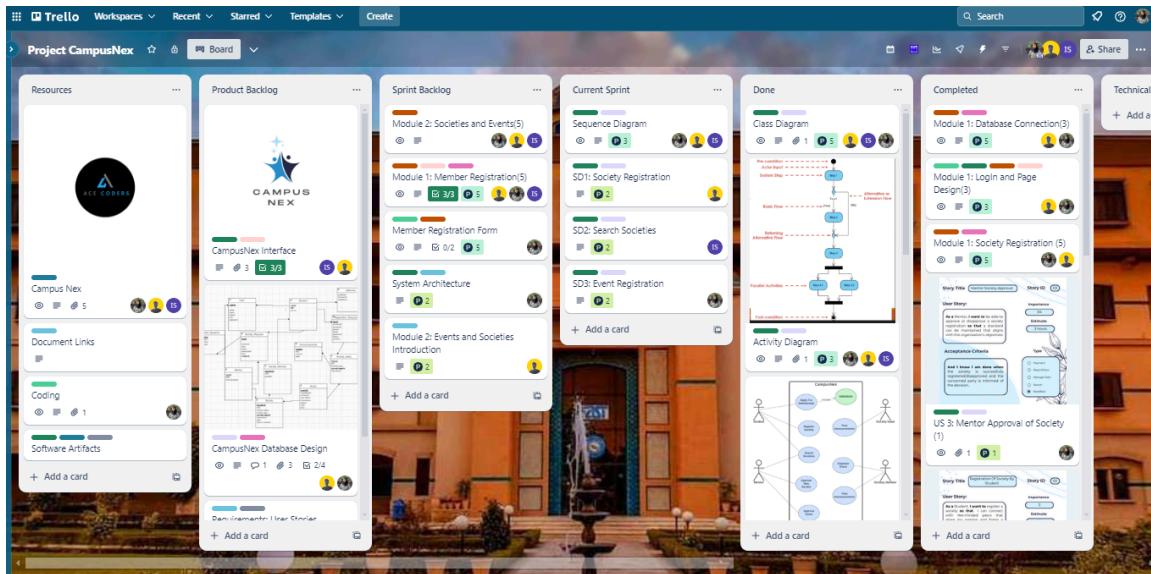
Initial Screenshot:



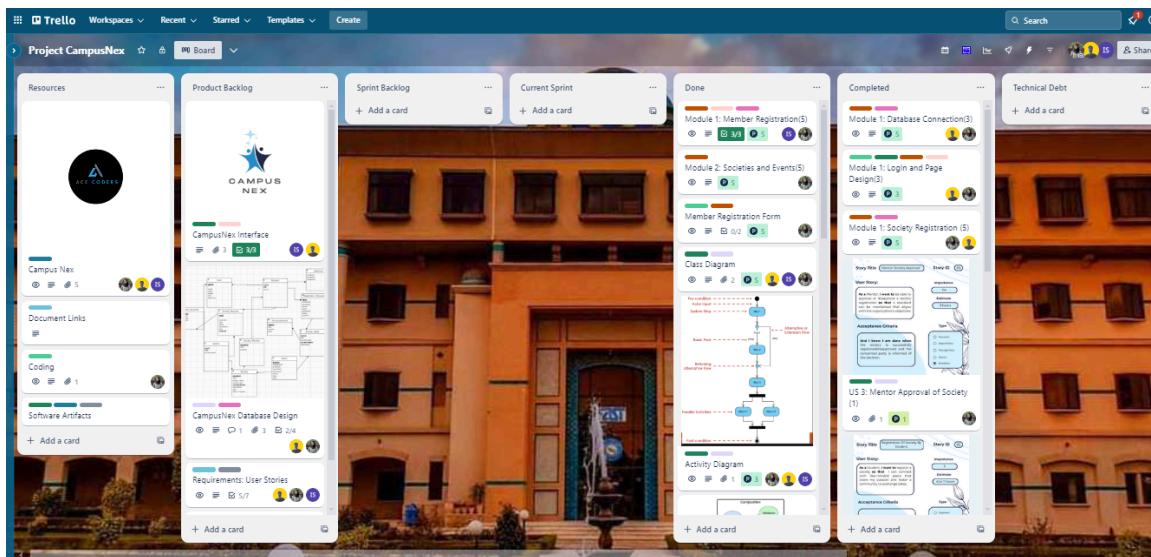
Mid Screenshot 1:



Mid Screenshot 2:



Final Screenshot:



Work division

Contributors	Document and Design	Implementation
Aiman Safdar	Sub Story 1.1 + 3.1 Structured Specification 1.1 + 3.1 Activity Diagram 02 Sequence Diagram 02 Document Design Use Case Diagram Class Diagram	View Society Page
Haris Sohail	User Story 4 Structured Specification 4 Activity Diagram 01 Sequence Diagram 01 Module Introduction Use Case Diagram Class Diagram System Architecture	Search Societies
Kalsoom Tariq	User Story 5 Structured Specification 5 Activity Diagram 03 Sequence Diagram	Member Registration Event Registration

03

Use Case Diagram
Class Diagram

Implementation

Technical Debt:

Societies Page

Fast Data Science Society

FDSS

Data, Data Everywhere

Our Head: aliza

About Us:

FDSS is a student organization dedicated to promoting and advancing knowledge, skills, and innovation in the field of computing through various educational, collaborative, and networking

[Register](#)



Member Registration Form

Registration Form For *Fast Data Science Society*

Name: kissa

ID: 572

Previously joined society:

Date Joined: 

Why do you want to join this society?



Request Sent

Registration Form For *Fast Data Science Society*

Name: kissa

ID: 572

Previously joined society:

Date Joined: 

Why do you want to join this society?



Head Accepts Request

The screenshot shows the CampusNex interface. On the left, there's a sidebar with a logo and navigation buttons: Societies, Events, Announcement, and Member Requests (which is highlighted). The main area is titled "Member Requests" and contains a table:

ID	Society Name	Student Name	View Request	Accept
4	Fast Data Science Society	hamna	More Details	Accept
5	Fast Data Science Society	kissa	More Details	Accept

Event Registration:

Event Page

The screenshot shows the CampusNex interface. On the left, there's a sidebar with a logo and navigation buttons: Societies, Events (which is highlighted), Announcement, and Member Requests. The main area is titled "All Events" and displays two events:

All Events

Fast Data Science Society
presents
Data Escapes

Date: 4/15/2024
Time: 15:00:00
Status: accepted

[More Details](#)

Society Events

Fast Computing Society
presents
Code Craft

Date: 6/15/2024
Time: 12:30:00
Status: pending

[More Details](#)

[Organize An Event !!](#)

More Details

The page displays event details for "Data Escapes" presented by "Fast Data Science Society". It includes a description, location, date, and time.

Fast Data Science Society presents Data Escapes

Description:
Join us for an immersive journey into the world of data science at "Data Escapes." This captivating event brings together data enthusiasts, experts, and novices alike to explore the endless possibilities and applications of data science.

Location: Community Hall

Date: 4/15/2024

Time: 15:00:00

Register New Event

The form allows users to register a new event, including fields for society selection, event title, type, description, date/time, and location.

Register a New Event

Select Society: Fast Computing Society

Event Title: CodingJail

Event Type: Workshop

Event Description: A Challenging and Fun Event

Insert Logo:

Date/Time: Monday, April 15, 2024

Location: Karakoram Lab

Register

Mentor Approval of Event

The screenshot shows the 'Society Requests' section with a table:

Society Name	Student Name	View Request	Accept
Fast Computing So...	kissa	View	Accept
Fast Computing So...	kalsoom	View	Accept

The 'Events Requests' section also contains a similar table:

Society Name	Event Name	Organizer Name	View Request	Accept
Fast Computing So...	Code Craft	kissa	View	Accept
Fast Computing So...	Coding Jail	kalsoom	View	Accept

Status Updated

The 'All Events' section displays an event card for 'Data Escapes':

Fast Data Science Society presents Data Escapes

Date: 4/15/2024
Time: 15:00:00
Status: accepted

[More Details](#)

The 'Society Events' section displays two event cards:

Fast Computing Society presents Code Craft

Date: 6/15/2024
Time: 12:30:00
Status: pending

[More Details](#)

Fast Computing Society presents Coding Jail

Date: 4/15/2024
Time: 22:07:00
Status: accepted

[More Details](#)

[Organize An Event !!](#)

BurnDown Chart

