Software Engineering

FAST Societies Hub

Deliverable #4: Iteration 3/Sprint3

Submitted By:

Haider Rizvi i210547

Maaz Ibne Khalid i210583

Hassan Raza i210465

Date of Submission:

28th April 2024

Table of Contents:

Project title and Introduction:

Title: FAST Societies Hub; FAST Societies Management System

Introduction:

The FAST Societies Management System is to be created to streamline the activities happening in each society of the university. It will allow the administrators to easily create, update or delete the societies of the university. Mentors/Faculty will approve the creation of societies as well as approve the request to join the society of the members. This system will make their job effortless and efficient. Students or members can easily view the list of societies and join whatever society which sparks their interest.

Overall, this system will make university life easier and more organized for administrators as well as the members or the students because societies are a significant part of your university life.

User stories:

User Story 1: Create Society (Student)

As a student, I want to be able to create a new society, so that I can initiate and lead university students around a specific interest or cause.

Sub-stories:

- As a student, I want to provide basic information about the new society including its name, department and its goals/ideas.
- As a student, I want to submit the society creation request for approval by the admin

User Story 2: View Society (Student)

As a student, I want to be able to view information about existing societies so that I can explore different options according to my interests

Sub-stories:

- As a student, I want to see a list of available societies with basic details such as ID, name and description.
- As a student, I want to be able to view upcoming events and activities organized by the society
- As a student, I want to see the societies I have already joined.

User Story 3: View Society (Admin)

As an admin, I want to be able to view information about existing societies so that I can monitor their activities and ensure that they follow the policies of university

Sub-stories:

- As an admin, I want to be able to see the list of existing societies with details such as ID of society, ID of student who created, name, description
- As an admin, I want to access administrative tools to delete society and manage approval requests.

User Story 4: Approve Society (Admin)

As an admin, I want to be able to review and approve new society creation requests to ensure they follow university policies.

Sub-stories:

- As an admin, I want to access the dashboard or notifications center to review pending society creation requests.
- As an admin, I want to approve or reject society creation requests based on university policies.

User Story 5: Delete Society (Admin)

As an admin, I want to be able to delete societies that are inactive or violate university policies to maintain the integrity of the university.

Sub-stories:

- As an admin, I want to access administrative tools to manage existing societies
- As an admin , I want to view a list of existing societies with options for deletion
- As an admin, I want to select and confirm deletion of a society and permanently remove the selected society along with its events and announcements from the system.

User Story 6: Edit Society (Admin)

As an admin, I want to be able to edit society's information so that the societies are up to date

Sub-stories:

- As an admin, I want to access administrative tools to edit existing societies
- As an admin, I want to be able to view all societies to select one for edit
- As an admin, I want to be able edit society name, society mentor and society description according to my own wishes.

User Story 7: Join Society (Student)

As a student, I want to be able to join an existing society, so that I can engage and work with university students around a specific interest or cause.

Sub-stories:

- As a student, I want to request to join society

- As a student, I want to be able to enter my information such as my ID, society ID, my department, team name and my ideas for society.
- As a student, I want to be able to choose society according to my own interests and see its details

User Story 8: Approve Society Member (Mentor)

As a mentor, I want to be able to approve requests to join societies so that I can have full control over who joins society and who doesn't.

Sub-stories:

- As a mentor, I want to view all society membership requests with details such as request ID, student ID, society ID, request date, status and other information necessary.
- As a mentor, I want to approve or reject Society members based on their profile and suitability in society

User Story 9: Create New Event (Society Executive)

As a society executive, I want to be able to create new events for my society so that I can create new opportunities for society members to engage in.

Sub-stories:

- As a society executive, I want to provide essential information about the event including its name, description, date and location.
- As a society executive, I want to send a request to create a new event to the mentor.

User Story 10: Approve Event(Mentor)

As a mentor, I want to be able to review and approve new events for my society so that I can ensure they align with the university policies and our goals.

Sub-stories:

- As a mentor, I want to receive notifications whenever there is a request to create a new event
- As a mentor, I want to review the details of the event.
- As a mentor, I want to accept or reject the request based on policies and goals

User Story 11: View Event(Society Executive)

As a society executive, I want to be able to view events for my society so that I can observe new opportunities for society members to engage in and overlook the event.

Sub-stories:

- As a society executive, I want to view essential information about the event including its name, description, date and location.

User Story 12: View Event(Society member)

As a Society member, I want to be able to view events for my society so that I can observe new opportunities for myself to engage in.

Sub-stories:

- As a Society member, I want to view essential information about the event including its name, description, date and location.

User Story 13: Edit Event(Society executive)

As a society executive, I want to be able to edit events for my society so that I can keep the events up to date with updated information.

Sub-stories:

- As a society executive, I want to edit essential information about the event including its name, description, date and location.

User Story 14: Create Announcement(Society Executive)

As a society executive, I want to be able to create new announcements for my society so that I can communicate with and notify the society members of important events.

Sub-stories:

- As a society executive, I want to add essential information about the announcement including my student ID, society ID, the title and the description.

User Story 15: Edit Announcement(Society Executive)

As a society executive, I want to be able to edit announcements for my society so that I can alter already communicated messages with the society members of important events.

Sub-stories:

- As a society executive, I want to edit essential information about the announcement including the title and the description.

User Story 16: View Announcement(Society Executive)

As a society executive, I want to be able to view announcements for my society so that I can decide whether to alter already communicated messages with the society members of important events.

Sub-stories:

- As a society executive, I want to view all the announcements including its ID, Society ID, title and description.

User Story 17: View Announcement(Society Member)

As a society member, I want to be able to view announcements made by executives so that I can be up to date with the latest news and information.

User Story 18: Assign Mentors (Admin)

As an admin, I want to be able to assign mentors to each society so that they can provide guidance to the society members

Sub-stories:

- As an admin, I want to view all the announcements including its ID, Society ID, title and description.

User Story 19: Delete Society Member (Mentor)

As a mentor, I want to be able to delete members from society so that I can remove inactive members who are not contributing to the society.

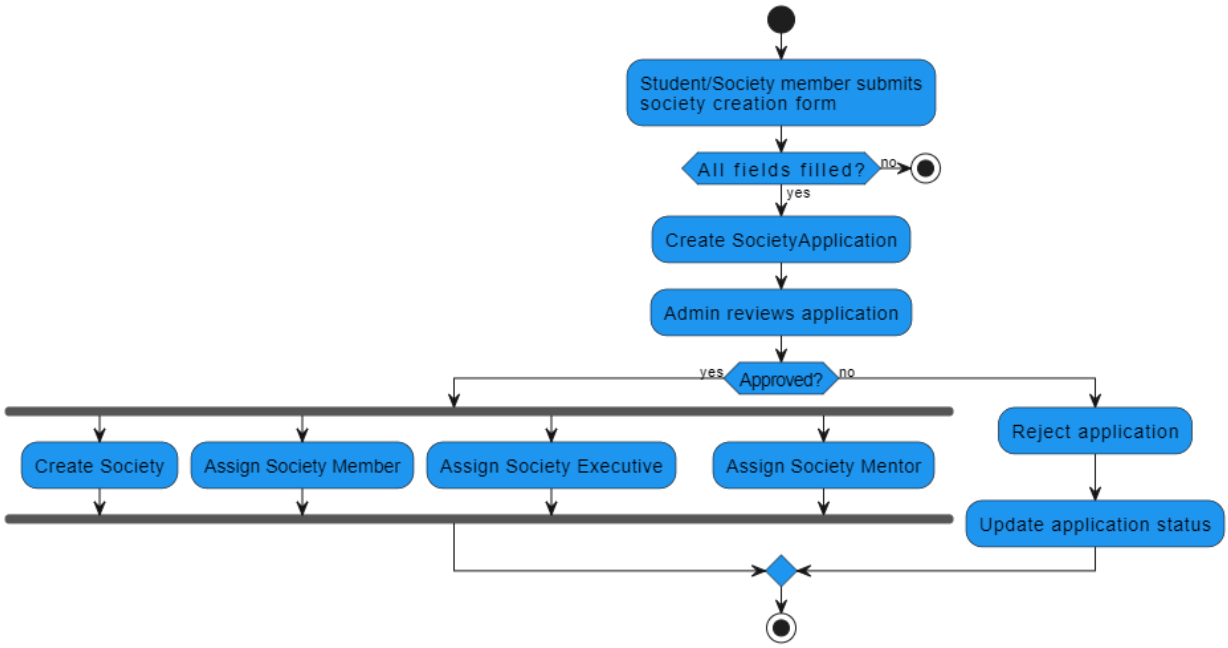
Sub-stories:

- As a mentor, I want to view all the members of the society along with their information and choose which one to delete.

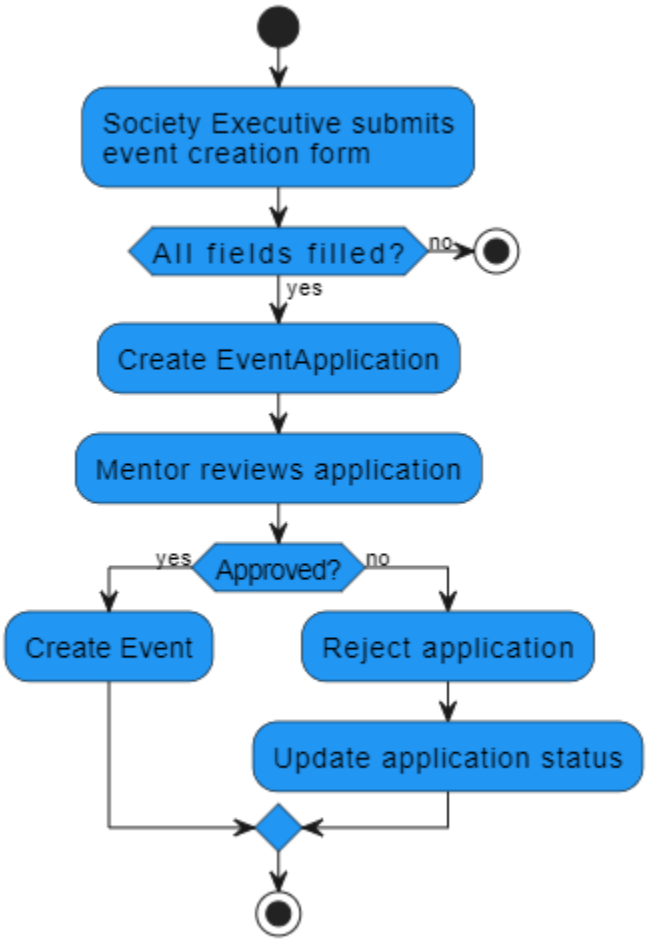
Design:

Activity Diagrams:

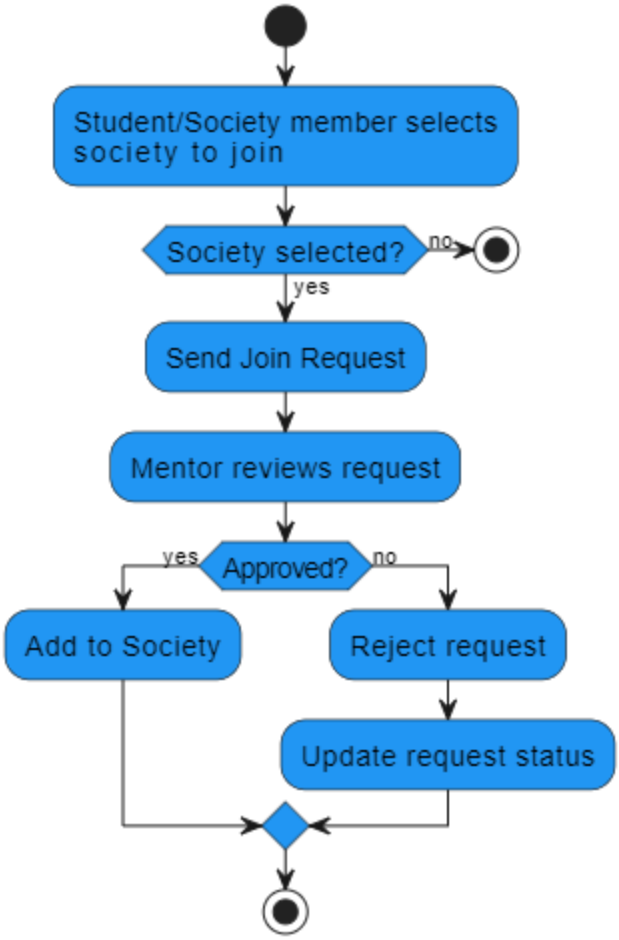
1. Create Society



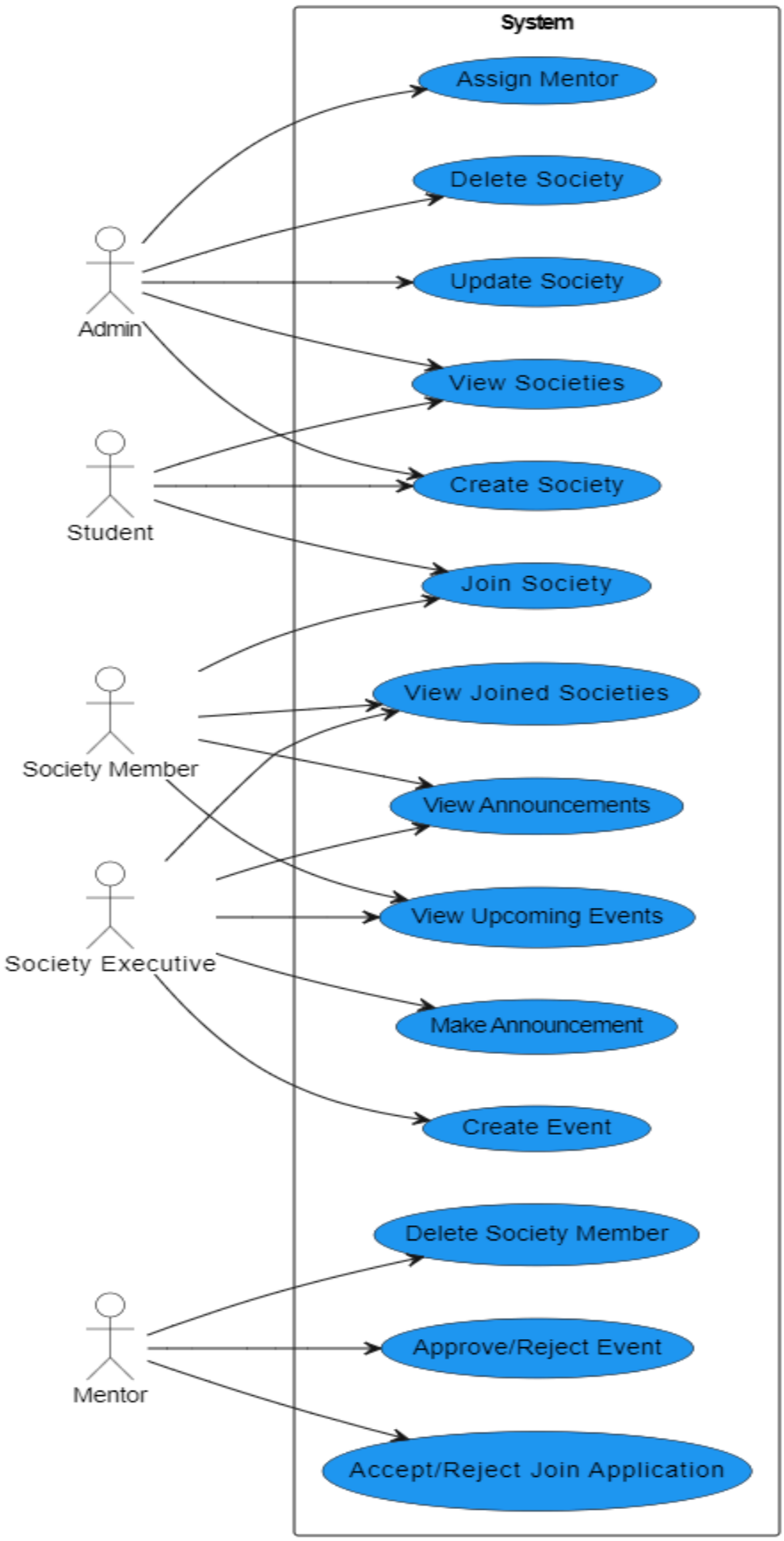
2. **Create Event**



3. **Join Society**

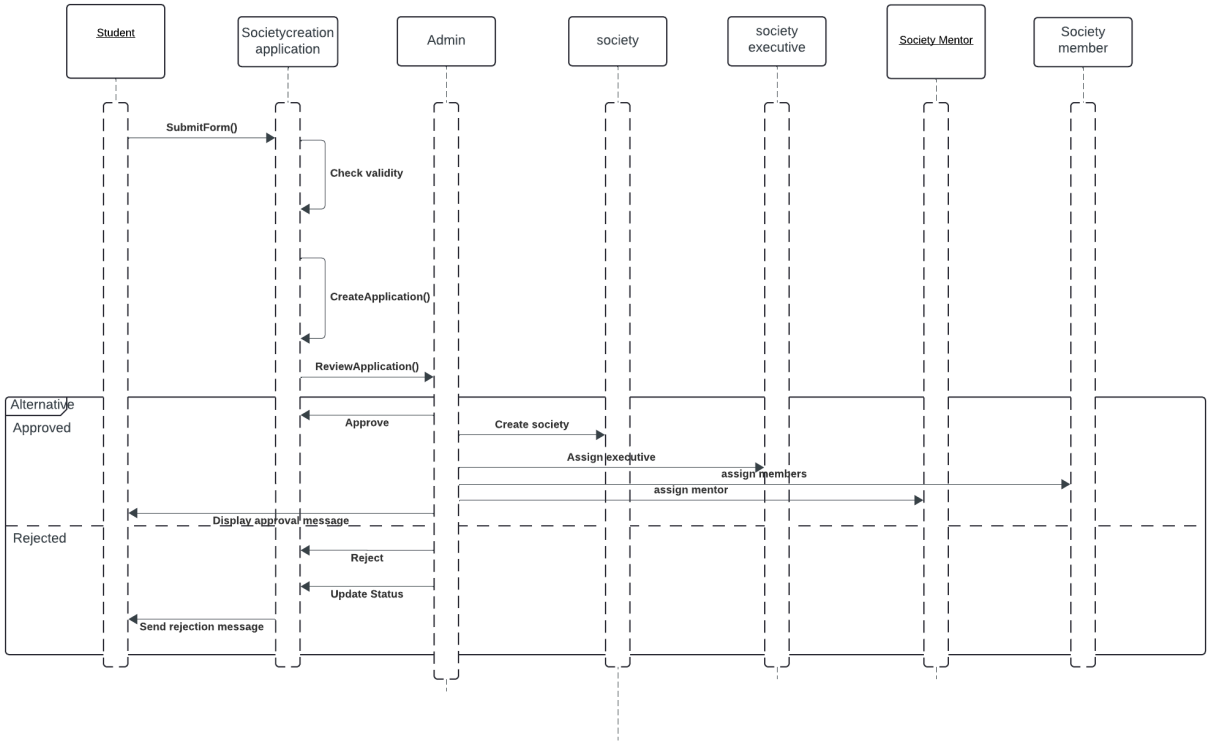

Student/Society member selects society to join
Society selected?
no
yes
Send Join Request
Mentor reviews request
yes
Approved?
no
Add to Society
Reject request
Update request status

Use Case Diagram:

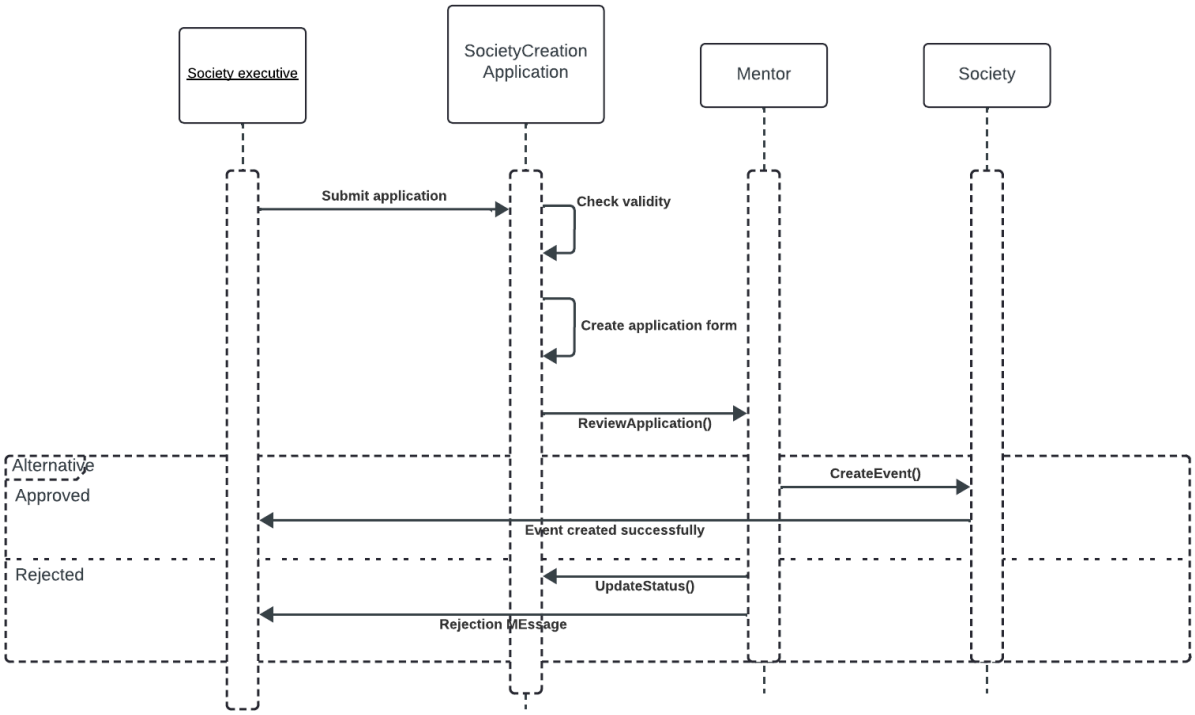


Sequence Diagrams:

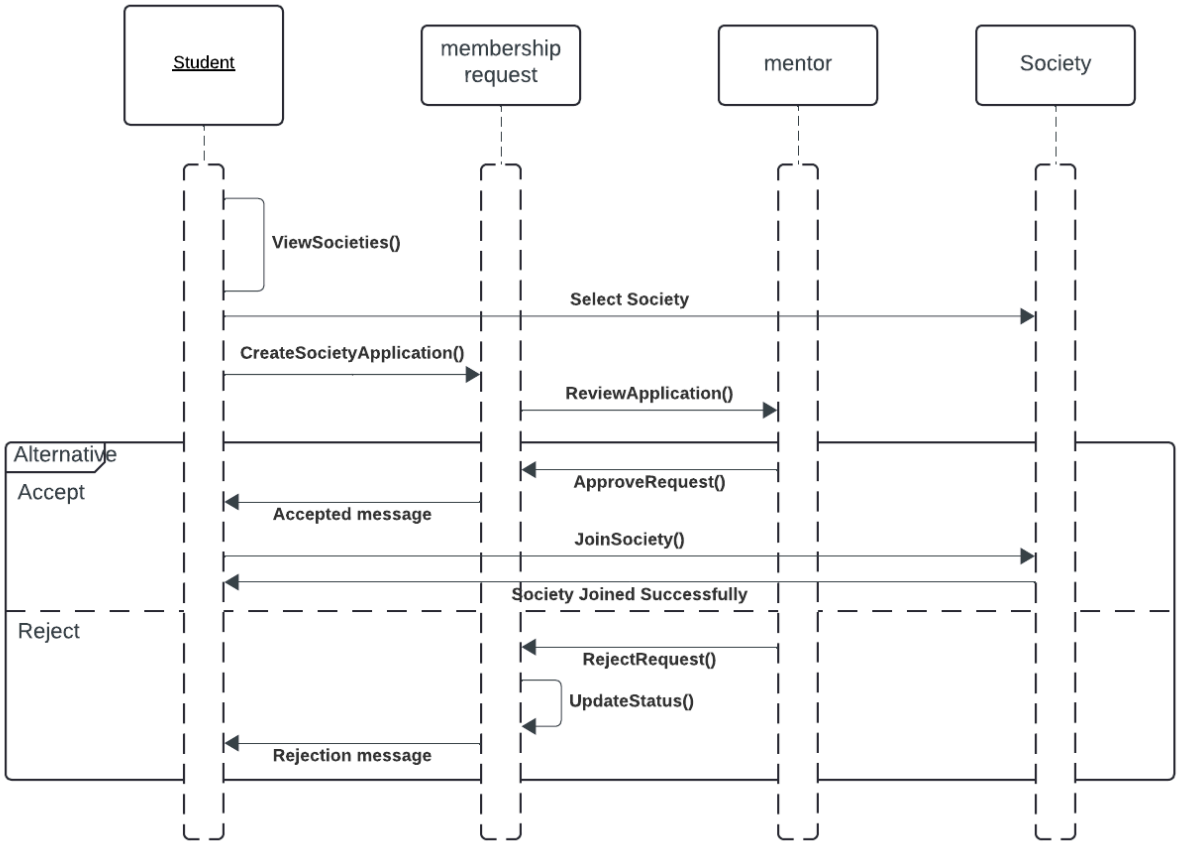
1. Create Society



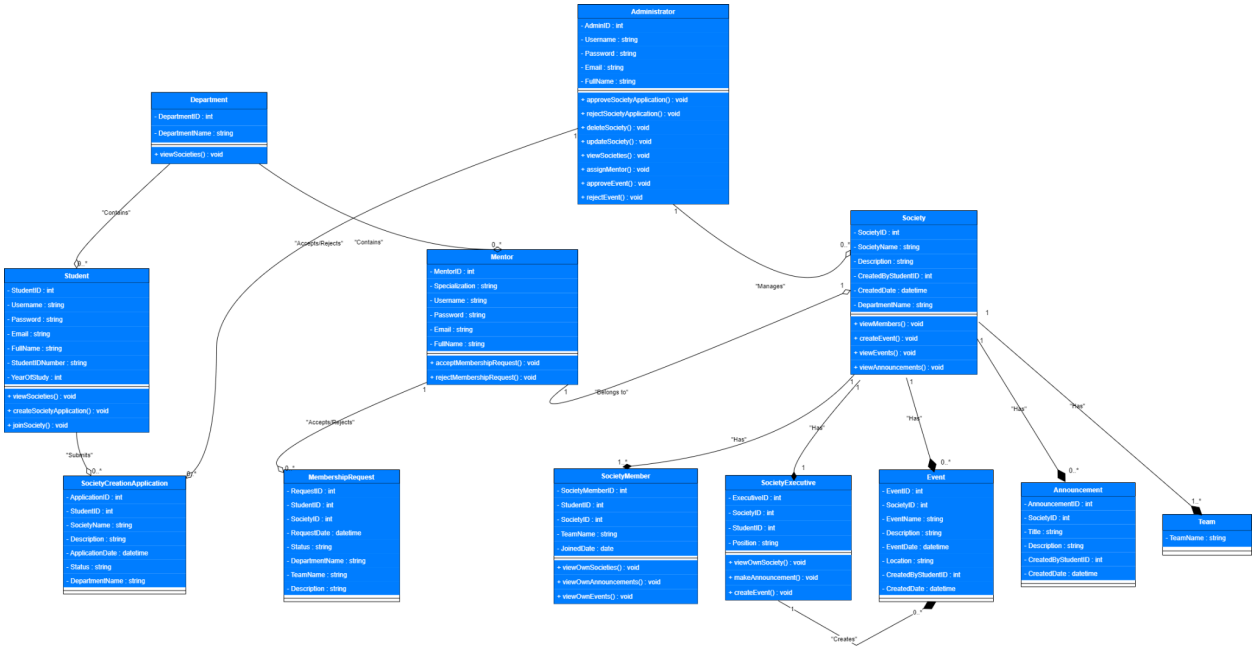
2. Create Event



3. Join Society



Class Diagram:



Architecture:

N Layered Architecture:

Our system architecture has incorporated a layered approach, specifically the n-tier architecture pattern, which helps us define our software into different layers, each serving a specific purpose. These layers are designed to promote modularity, scalability, and maintainability in our application.

Presentation Layer:

This is the user interface layer, it is the most top layer of our architecture. This is the layer in which users interact with the system and this layer includes all the components like forms, user input and all graphical elements.

Business Logic Layer:

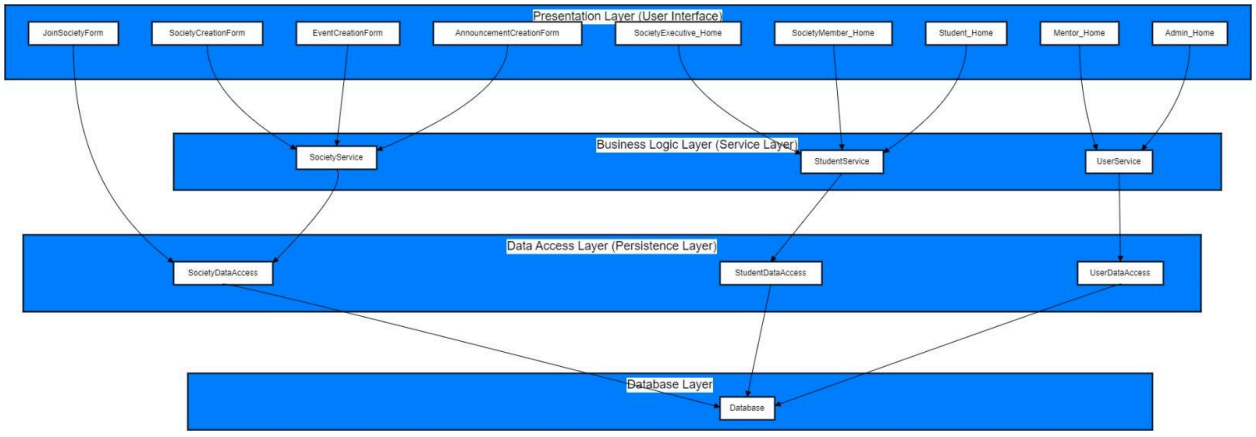
This is the layer between presentation and data access layer. It contains the logic of our system where needed. Extra code is eliminated from the presentation layer and all the reasoning and logic of the system is used in this layer. For example, the presentation layer calls the function to assign a mentor, now all the logic is enclosed in this layer i.e determining if the mentor exists in the system and then performing actions according to that. This layer is directly communicating with the data access layer which is beneath it and providing results to the presentation layer which is above it.

Data Access Layer:

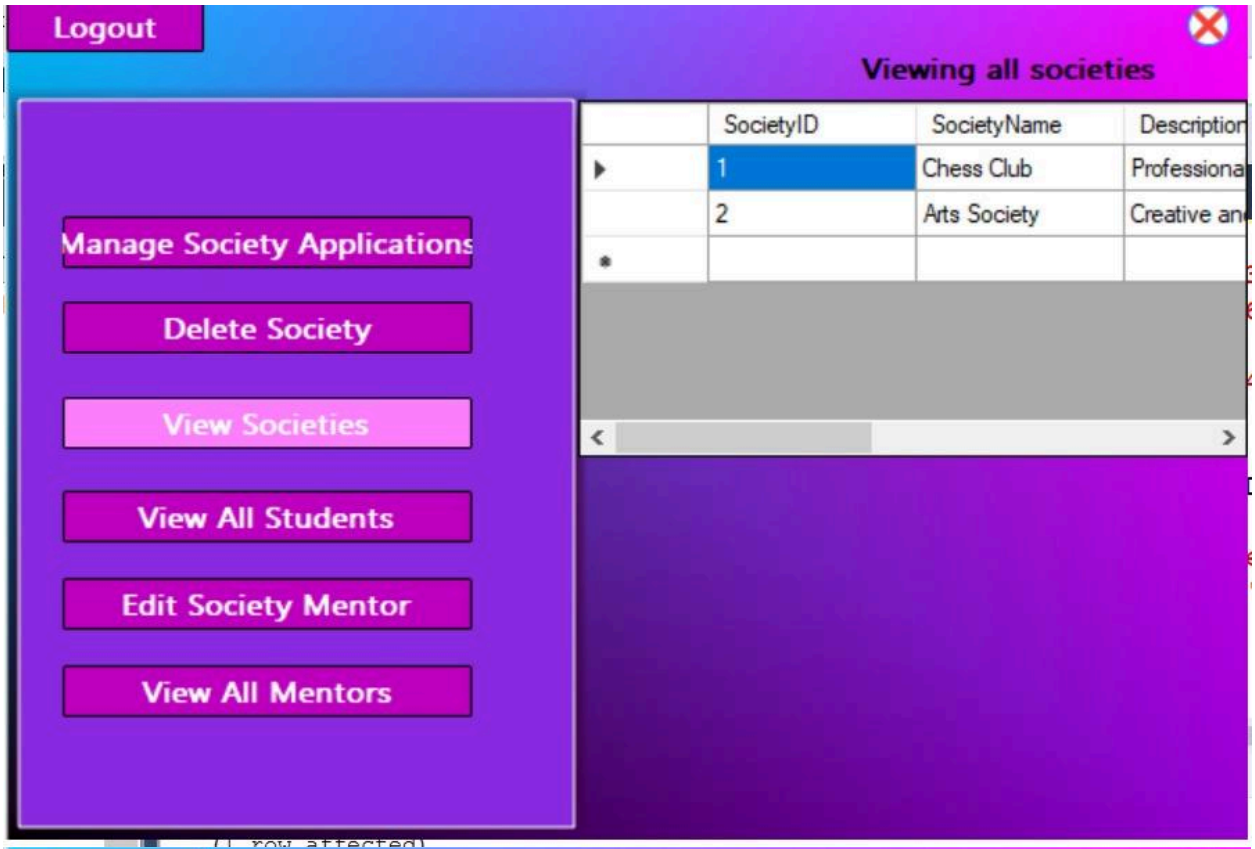
This layer handles the interaction with the database. It creates, inserts, deletes, and updates the entries directly in the database and this layer is the most important layer of our system. This layer abstracts the details of underlying architecture in the database for the layer above it which is the business logic layer.

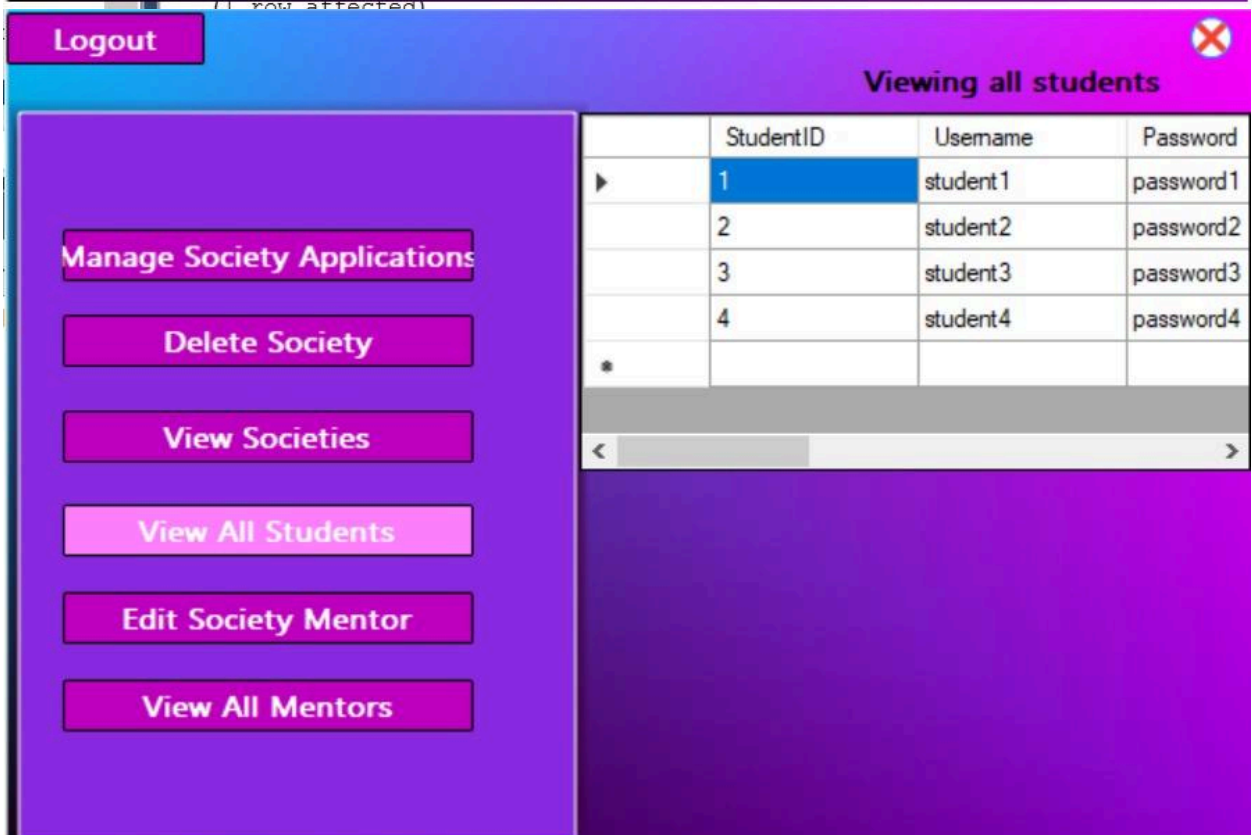
Database Layer:

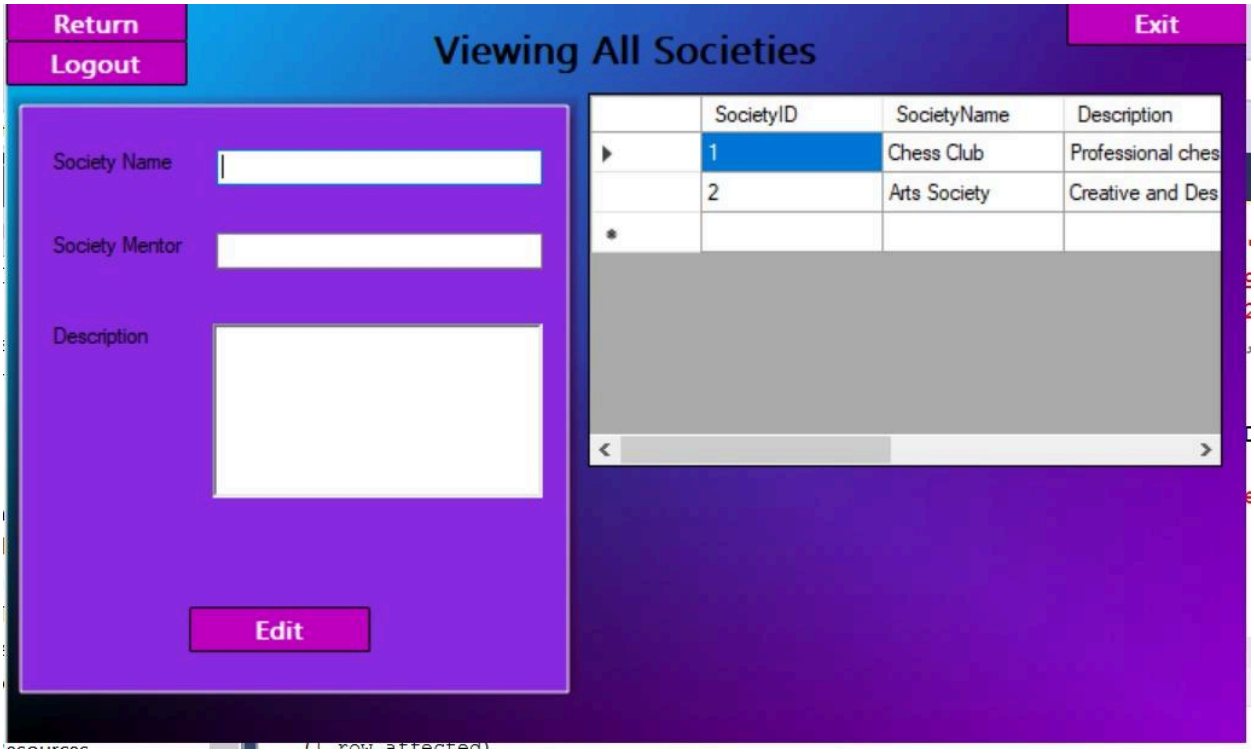
This includes the actual database storage system which in our case is Microsoft SQL Server.

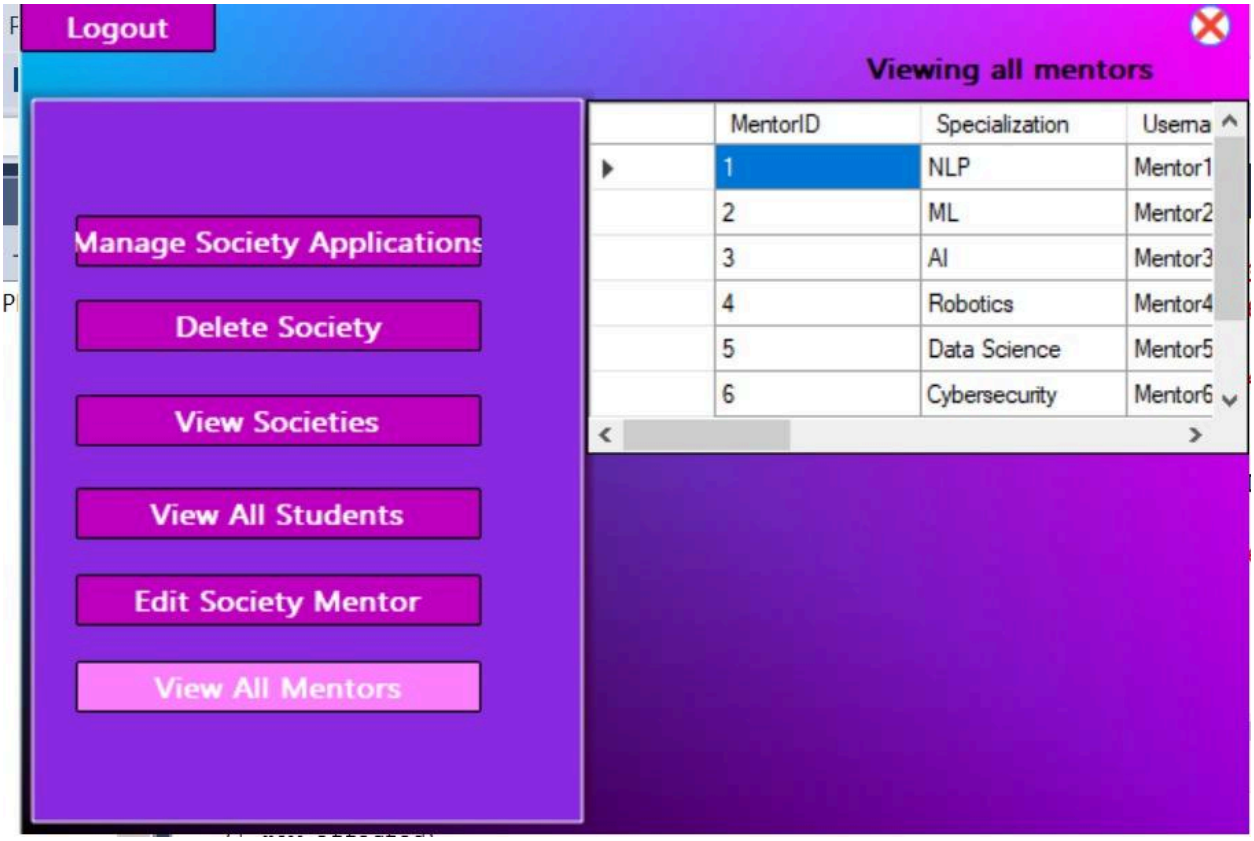


Implementation Screenshots:



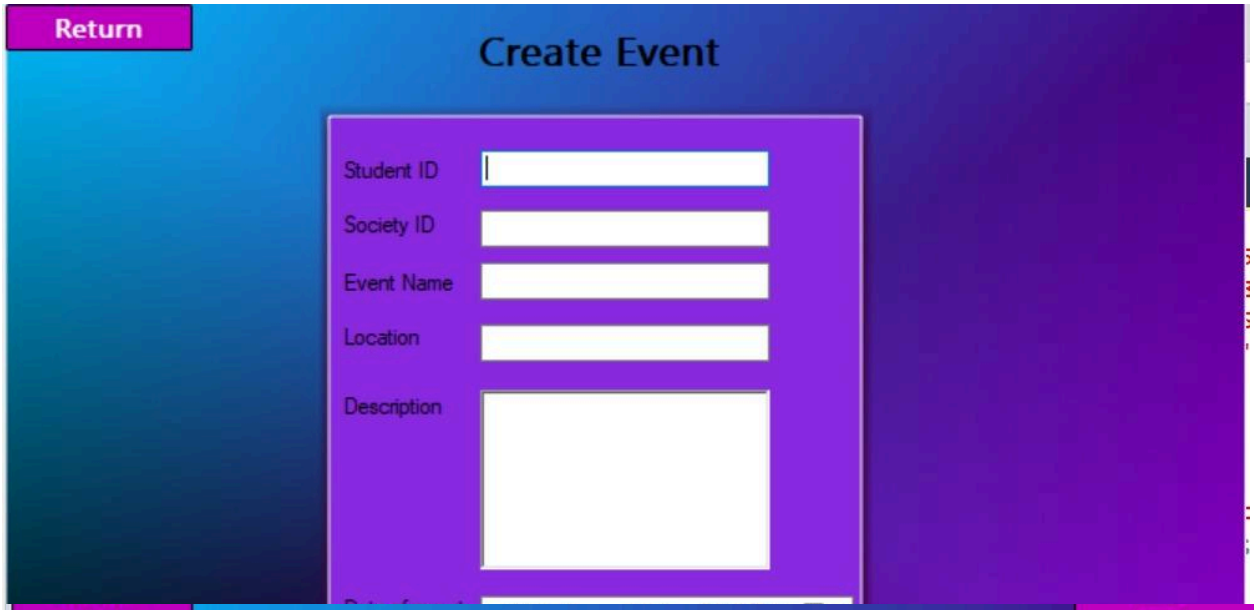


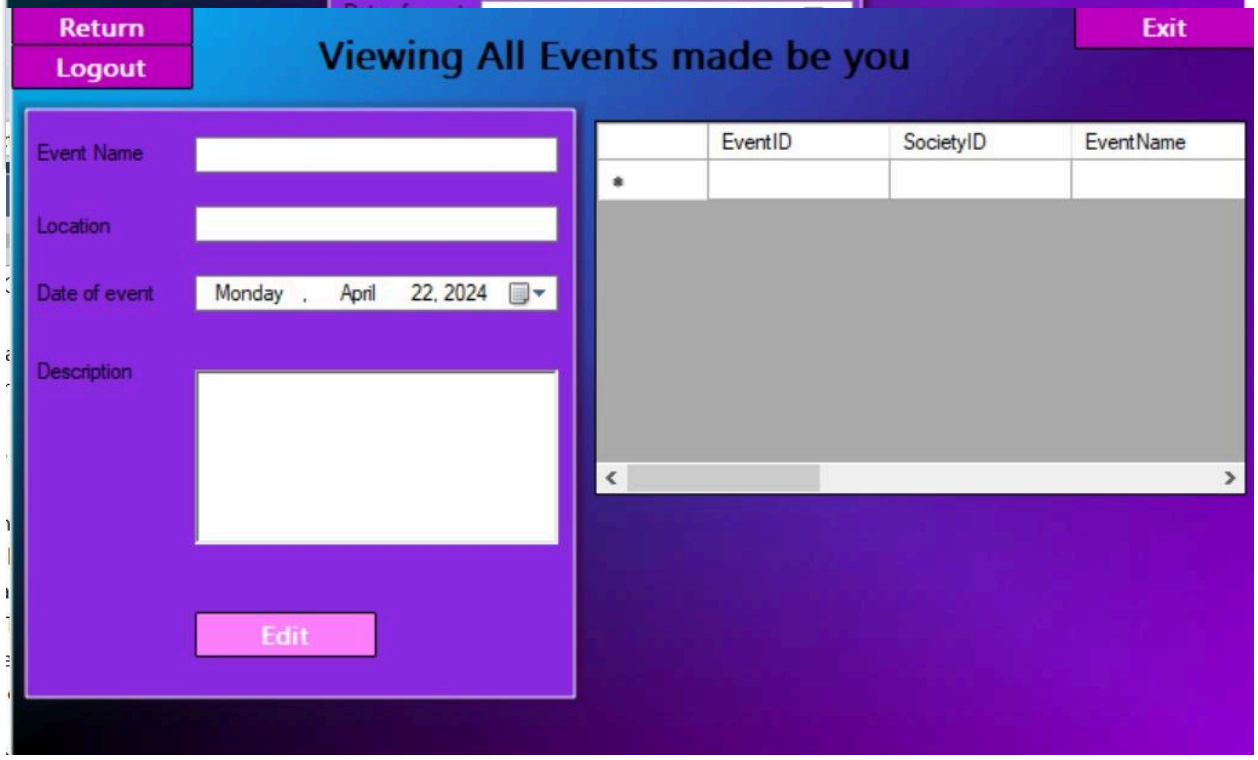

Return
Logout
Viewing All Societies
Exit
Society Name
Society Mentor
Description
Edit
SocietyID
SocietyName
Description
1
Chess Club
Professional ches
2
Arts Society
Creative and Des


Logout
Viewing all mentors
Manage Society Applications
Delete Society
View Societies
View All Students
Edit Society Mentor
View All Mentors
MentorID
Specialization
Userna
1
NLP
Mentor1
2
ML
Mentor2
3
AI
Mentor3
4
Robotics
Mentor4
5
Data Science
Mentor5
6
Cybersecurity
Mentor6


Logout
Exit
Welcome John Doe As Society Executive
Create Event
View your Event Requests
View Scheduled Events
Create Announcement
View Joined Societie's Events
Joined Soceity's Anouncem
View Announcement
Edit Announcement
Edit Event


Return
Create New Announcement
Student ID
Society ID
Title
Description
Submit


Return
Create Event
Student ID
Society ID
Event Name
Location
Description

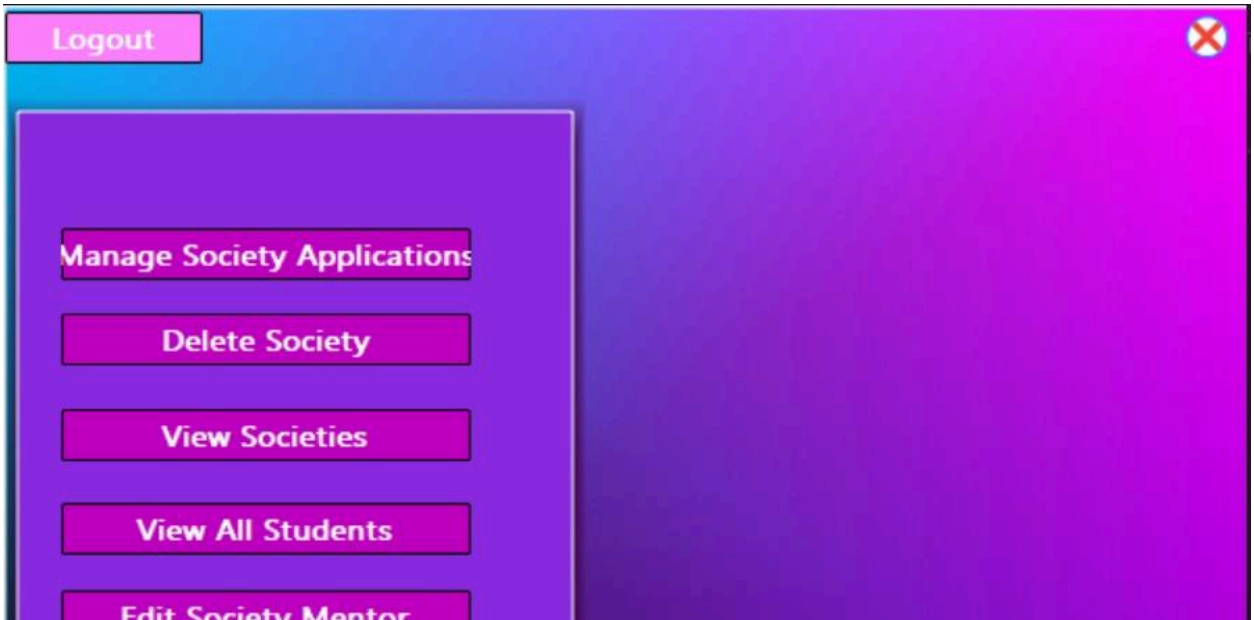

Return
Logout
Exit
Viewing All Events made be you
Event Name
Location
Date of event
Monday , April 22, 2024
Description
Edit
EventID
SocietyID
EventName

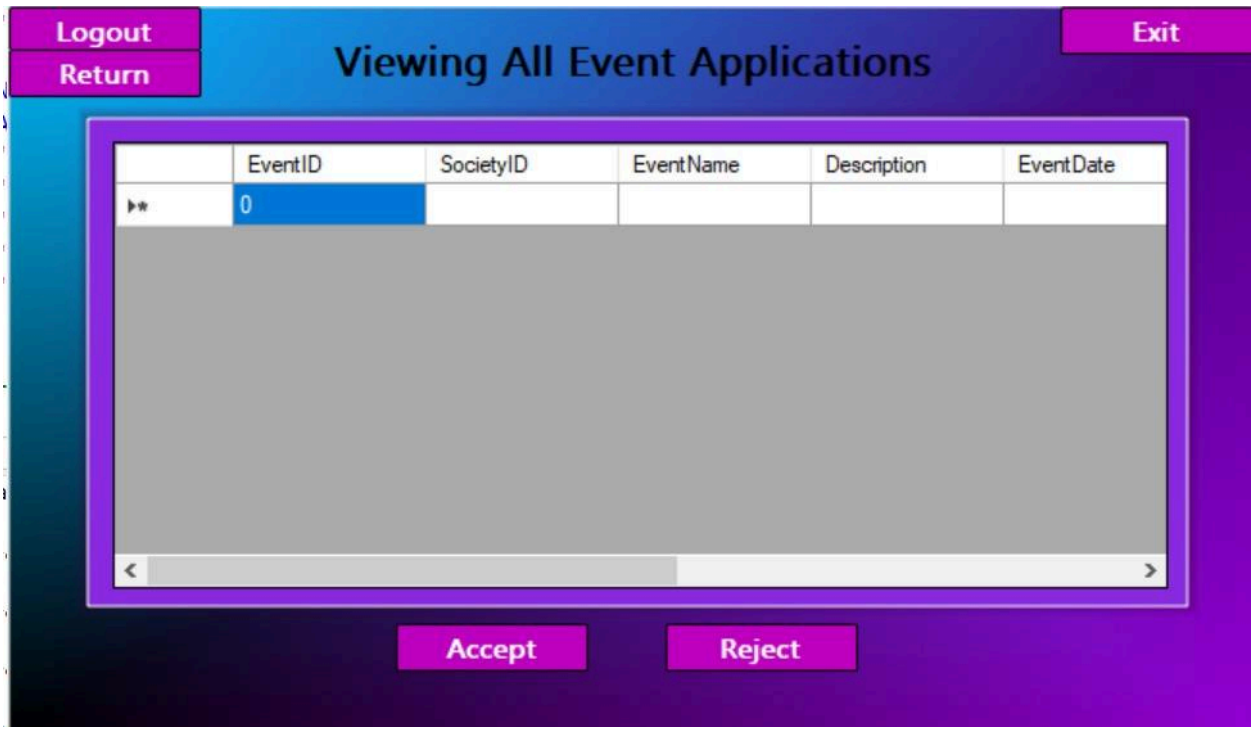

Return
Logout
Exit
Viewing All Announcements made be you
Title
Description
AnnouncementID
SocietyID
Title
Description
0

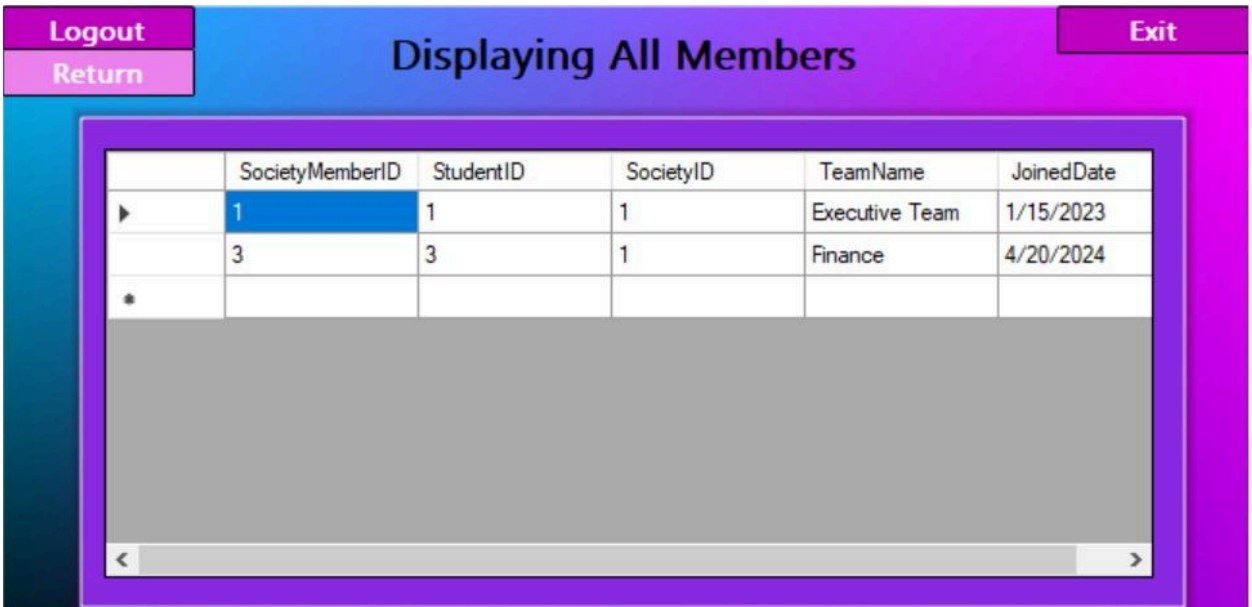

Exit
Welcome to Society Management System
Choose User Type
Admin
Student
Mentor


Return
Displaying All societies
Select A society to delete
SocietyID
SocietyName
Description
CreatedByStudentII
DepartmentN
1
Chess Club
Professional ches...
1
Computer Scie
2
Arts Society
Creative and Des...
2
Electrical Engi

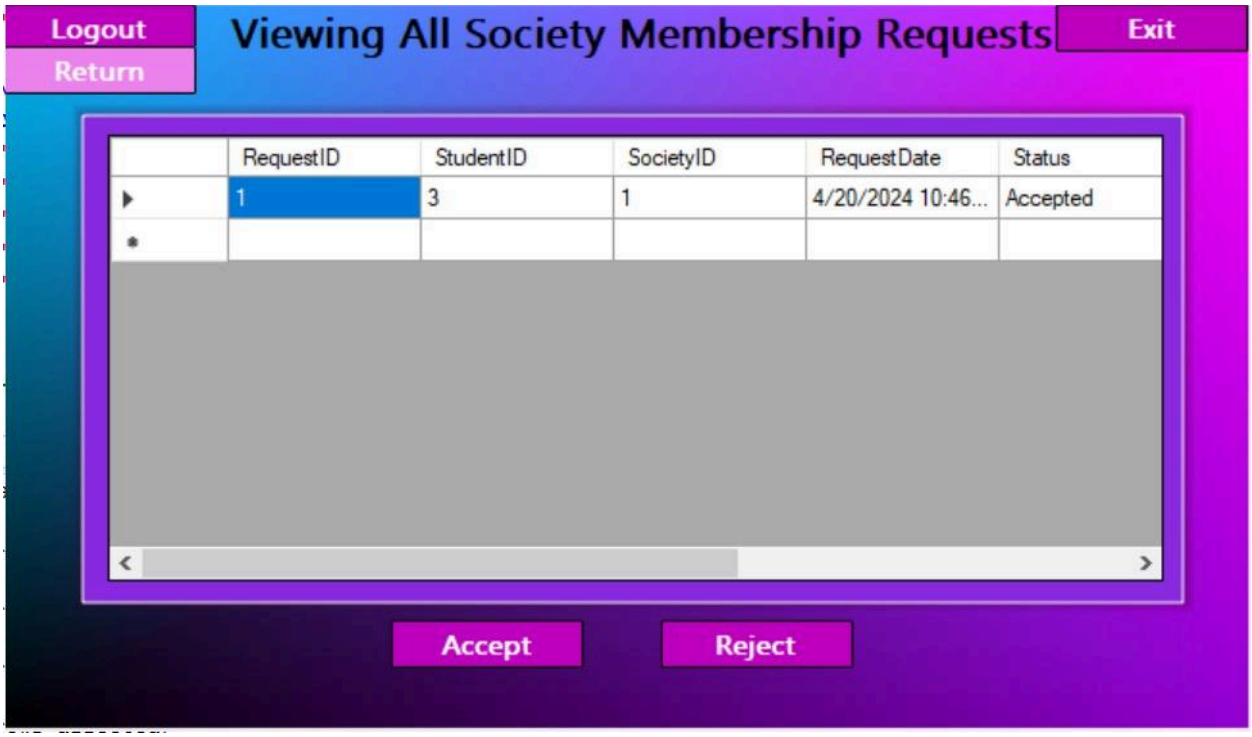

Logout
Return
Exit
Welcome Mentor John
Approve Member
Approve Event
Delete Member


Logout
Manage Society Applications
Delete Society
View Societies
View All Students
Edit Society Mentor

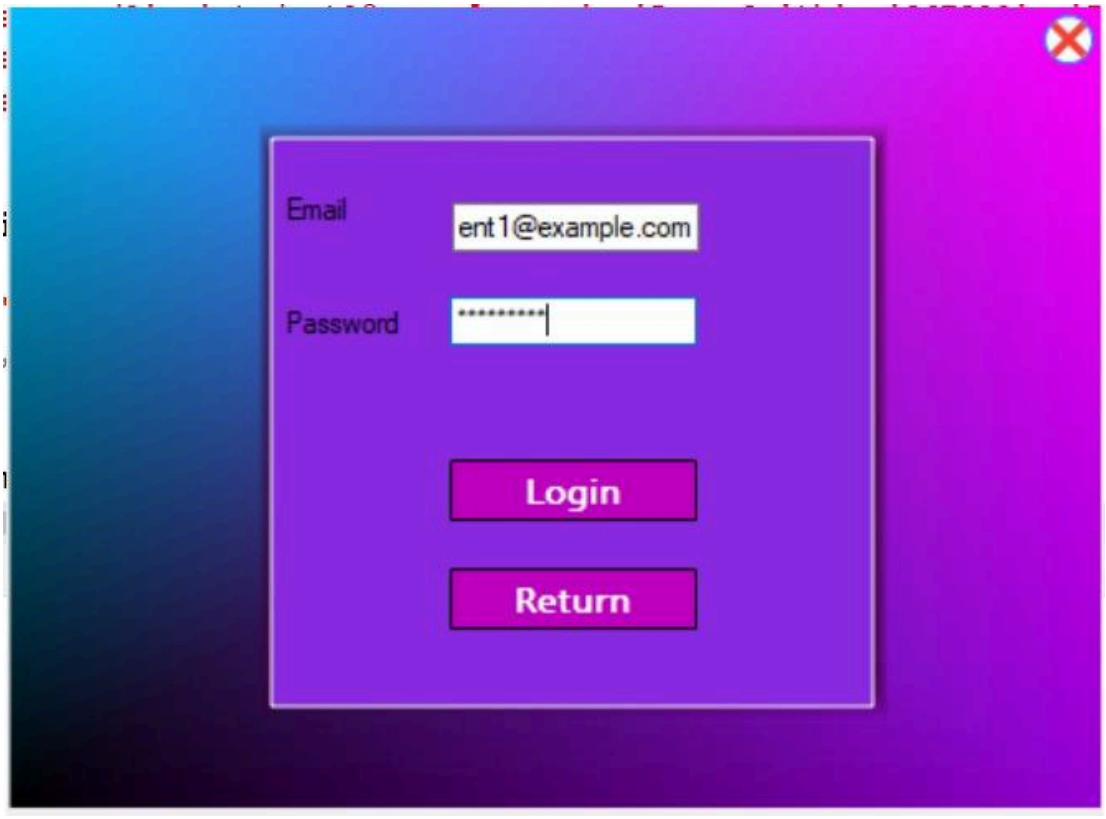

Logout
Return
Viewing All Event Applications
Exit
EventID
SocietyID
EventName
Description
EventDate
0
Accept
Reject

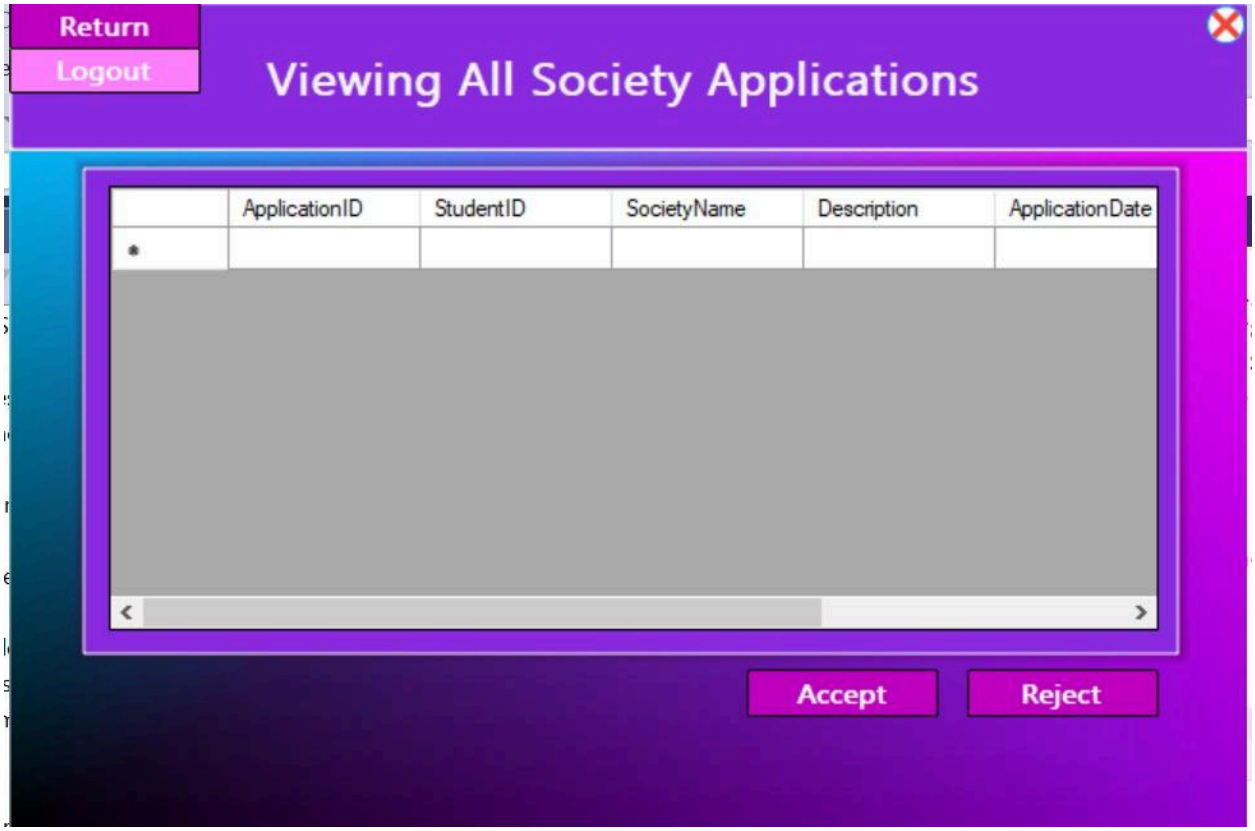

Logout
Return
Displaying All Members
Exit
SocietyMemberID
StudentID
SocietyID
TeamName
JoinedDate
1
1
1
Executive Team
1/15/2023
3
3
1
Finance
4/20/2024


Logout
Exit
Welcome John Doe As Society Executive
Create Event
View your Event Requests
View Scheduled Events
Create Announcement
View Joined Societie's Events
Joined Soceity's Anouncem
View Announcement
Edit Announcement
Edit Event
Viewing all your Schedued Events
EventID
SocietyID
EventName
Description

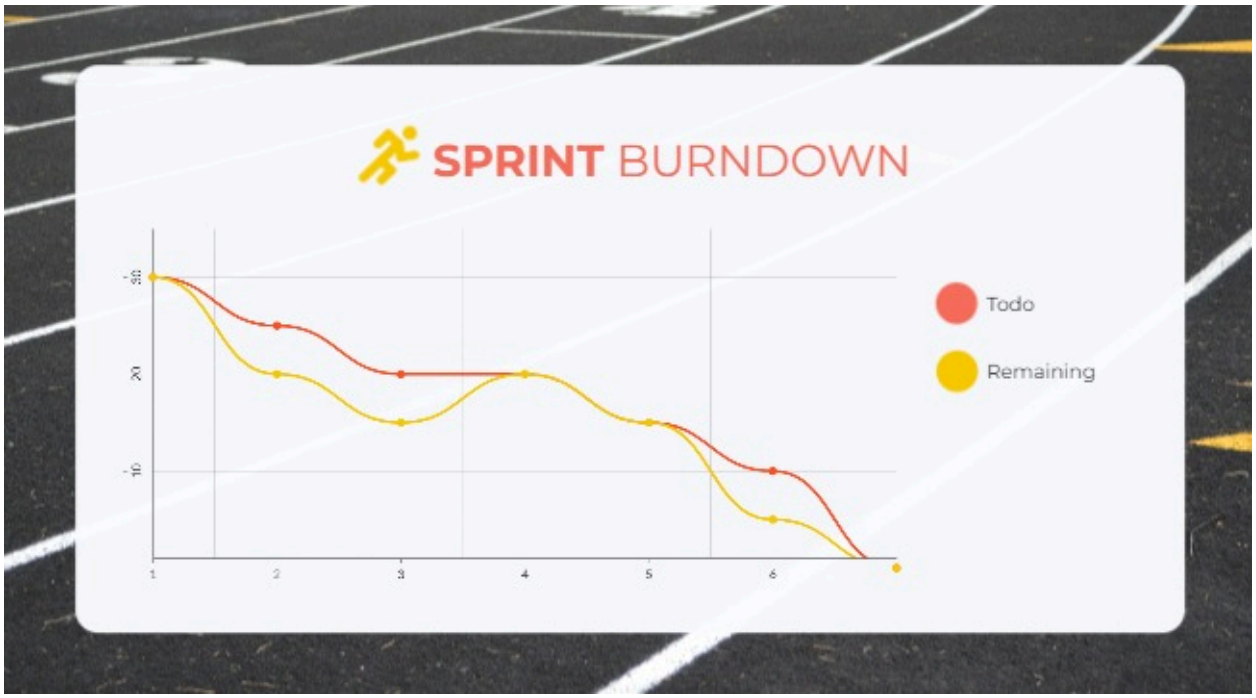

Logout
Return
Viewing All Society Membership Requests
Exit
RequestID
StudentID
SocietyID
RequestDate
Status
1
3
1
4/20/2024 10:46...
Accepted
Accept
Reject


Logout
Exit
Welcome John Doe As Society Executive
Create Event
View your Event Requests
View Scheduled Events
Create Announcement
View Joined Societie's Events
v Joined Soceity's Anouncem
View Announcement
Edit Announcement
Edit Event
Viewing all your Event Requests
EventID
SocietyID
EventName
Description


Email
ent1@example.com
Password
Login
Return

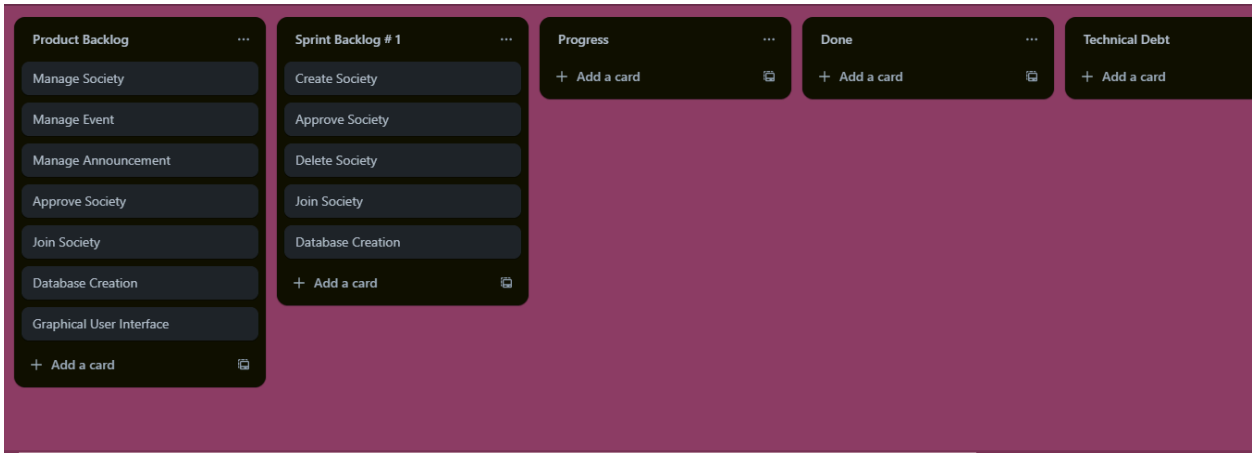

Return
Logout
Viewing All Society Applications
ApplicationID
StudentID
SocietyName
Description
ApplicationDate
Accept
Reject

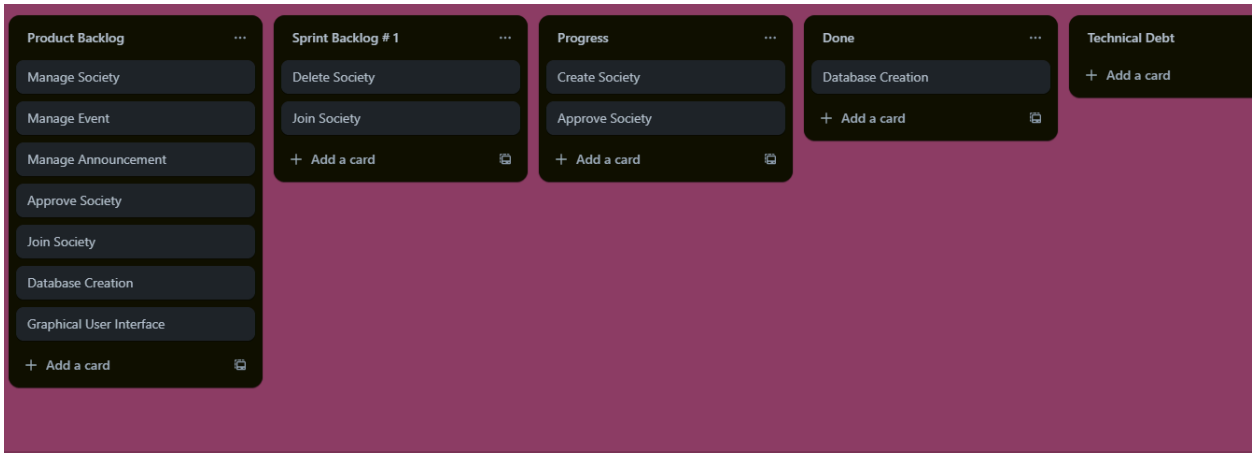
Project BurnDown Chart:

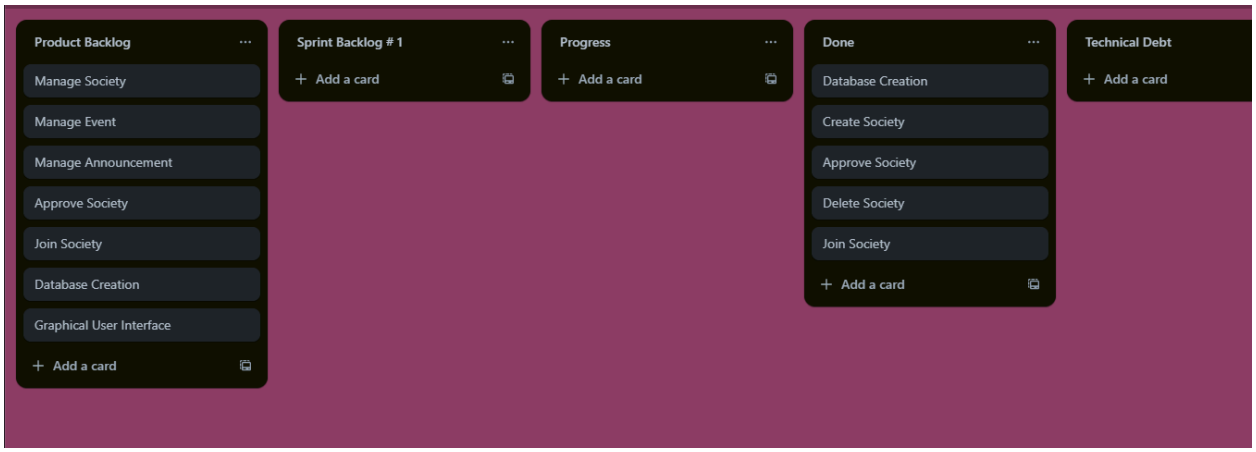


Trello Board Screenshots:

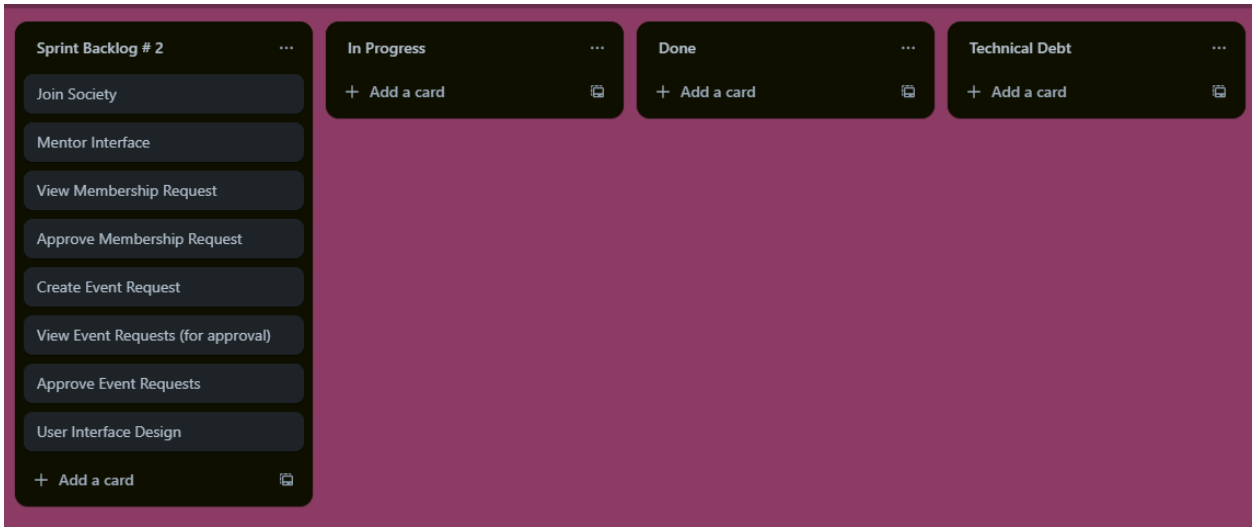
Iteration 1:

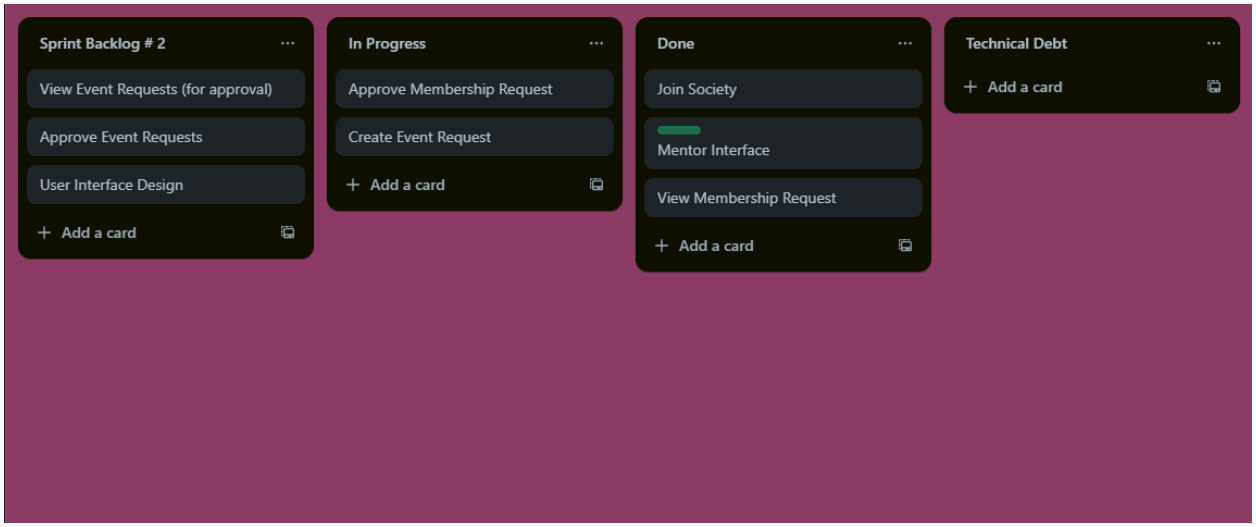


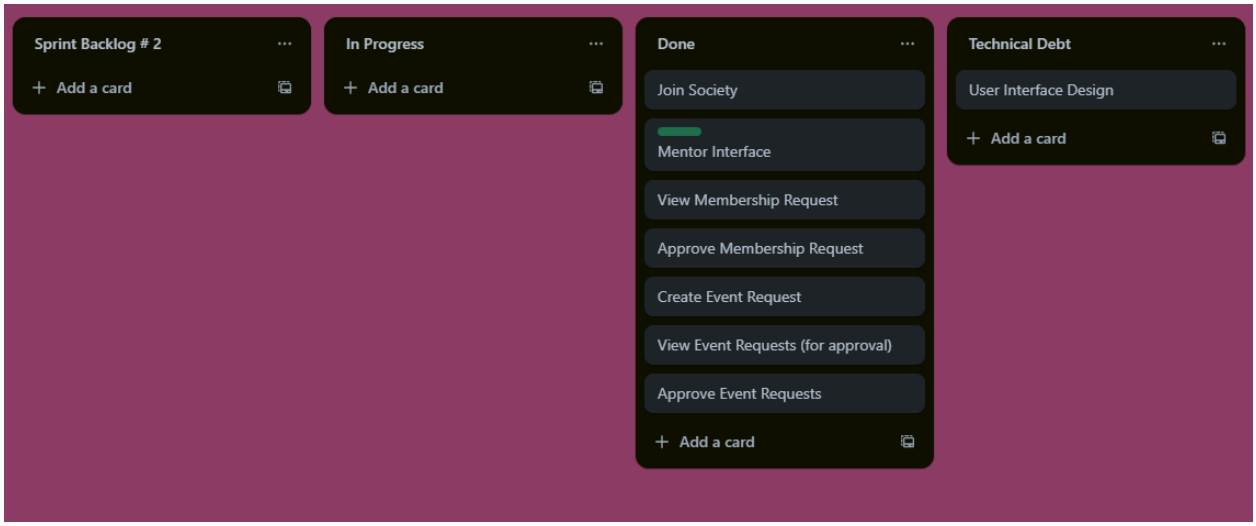




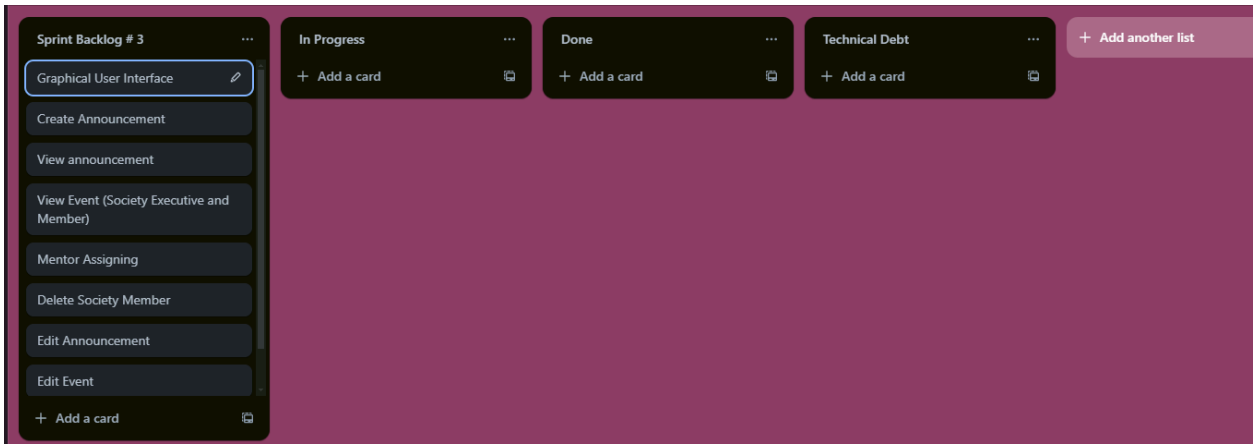
Iteration 2:

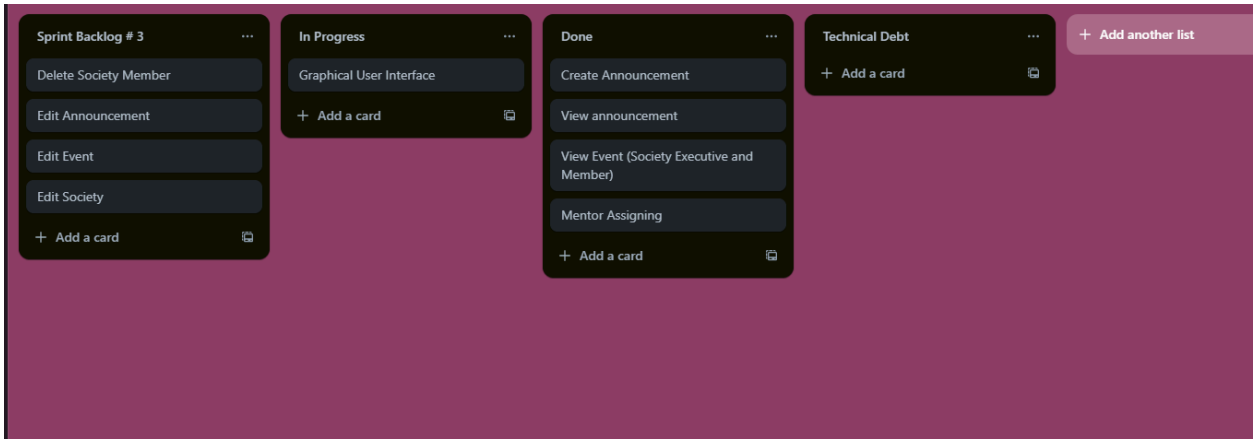

Sprint Backlog # 2
Join Society
Mentor Interface
View Membership Request
Approve Membership Request
Create Event Request
View Event Requests (for approval)
Approve Event Requests
User Interface Design
Add a card
In Progress
Add a card
Done
Add a card
Technical Debt
Add a card

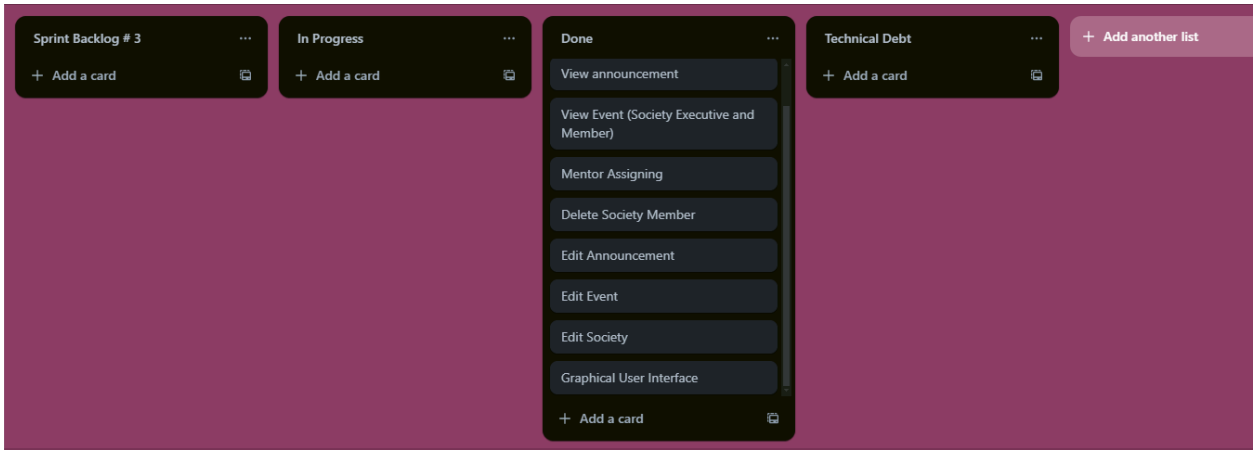

Sprint Backlog # 2
View Event Requests (for approval)
Approve Event Requests
User Interface Design
Add a card
In Progress
Approve Membership Request
Create Event Request
Add a card
Done
Join Society
Mentor Interface
View Membership Request
Add a card
Technical Debt
Add a card


Sprint Backlog # 2
Add a card
In Progress
Add a card
Done
Join Society
Mentor Interface
View Membership Request
Approve Membership Request
Create Event Request
View Event Requests (for approval)
Approve Event Requests
Add a card
Technical Debt
User Interface Design
Add a card

Iteration 3:


Sprint Backlog # 3
Graphical User Interface
Create Announcement
View announcement
View Event (Society Executive and Member)
Mentor Assigning
Delete Society Member
Edit Announcement
Edit Event
Add a card
In Progress
Add a card
Done
Add a card
Technical Debt
Add a card
Add another list


Sprint Backlog # 3
Delete Society Member
Edit Announcement
Edit Event
Edit Society
Add a card
In Progress
Graphical User Interface
Add a card
Done
Create Announcement
View announcement
View Event (Society Executive and Member)
Mentor Assigning
Add a card
Technical Debt
Add a card
Add another list



Boundary Value Analysis Testing:

1) Login screen:

Valid Classes:

Email: student1@example.com

Password: pass1

Invalid Classes:

Email:
@example.com

student@exampleexample.com

Password:

P

pass

Student Login:

Email	Password	Outcome
student@exampleexample.com	pass	Invalid credentials
student1@example.com	password1	Login successful
@example.com	P	Invalid credentials

Text Case 1: (Login page student)

Inputs:

Email: student1@gmail.com

Password: pass1

Output: logged in successfully

Outcome: student logged in to the app successfully

Text Case 2: (Login page student)

Inputs:

Email: student@@example.com

Password: pass1

Output: incorrect credentials

Outcome: Student unable to login

Test Case 3: (Login page student)

Inputs:

Email: student@exampleexample.com

Password: password123@

Output: incorrect credentials

Outcome: Student unable to login

Mentor Login:

Email	**Password**	**Outcome**
mentor@example.com.com	pass	Invalid login
mentor1@example.com	pass1	Login sucessfully
@example.com	pass	Invalid Login

Admin Login:

Email	**Password**	**Outcome**

admin@example.com	pass	Login sucessful
admin@example.com.com	pass1	Invalid Login
@example.com	pass	Invalid Login

2) Create Society page:

Valid classes:

Student ID:

- Matches the current student
- Is not empty

Society Name:

- Contains only alphabets
- Is not empty

Department:

- Computer Science, Electrical Engineering, Business, Artificial Intelligence, Data Science, Cyber Security, Software Engineering
- Is not empty

Description:

- Is not empty

Invalid classes:

Student ID:

- Does not Match the current student
- Is empty

Society Name:

- Contains numbers
- Contains Characters
- Is empty

Department:

- Is empty

Description:

- Is empty

Weak equivalence:

Society Name	**Student ID**	**Department**	**Description**	**Outcome**
Arts	3	Business	Good society	Created successfully
Arts123	1	“ “	“ “	Invalid entries
Arts@	2	“ “	“ “	Invalid entries

Work Division:

1) Hassan Raza:

Documentation (Minor), Design (Use Case Diagram, Class Diagram), Implementation (backend), Scrum Board

2) Haider Rizvi:

Documentation (Major), Design (Activity Diagrams), Implementation (frontend), Presentation

3) Maaz Khalid:

Implementation (frontend), Design (Sequence Diagrams), Testing and Validation

Lesson Learnt:

Throughout our project of developing a university society management system, we learnt many lessons along the way. Before the project, we were not aware of the extensive documentation, and how vital documentation is. Like writing user stories of each module, designing the architecture of the whole system and stuff like sprint backlogs are important too. We learnt that doing all this work aside from the implementation, is vital as it organizes the project really well, helping not only the user but also the developers as well in understanding the scope of the project, the requirements and it helps in the implementation itself. To summarize, we learnt that documentation is crucial when developing a software as it guides the project's direction effectively.

Furthermore, this is the first time that we worked on the code along with its test cases from which we learnt that writing test cases for your implementation is integral in building a software.

Also for the first time we used trello to manage our sprint backlogs and made burndown charts, which was a learning experience too.

Lastly, the project taught us time management and working together as a team. This was the first time we worked in an agile development methodology, working on sprints and iterations which will aid us in our future jobs and career as well.