



Daffodil
International
University

Committee Management System

By

Md. Kamrul Hasan

151-35-1049

&

Md. Nazmul Rubel

151-35-1069

A project submitted in partial fulfillment of the requirement for the degree
of Bachelor of Science in Software Engineering.

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Fall-2018

ACKNOWLEDGEMENT

First of all, we are grateful to The Almighty Allah for making us eligible to complete this project. Then we should like to thank my supervisor Syeda Sumbul Hossain, Lecturer, Department of Software Engineering. I am extremely grateful and indebted to her expert, sincere and valuable guidance and encouragement extended to us.

Beside my supervisor, we would like to express our profound gratitude to Syeda Sumbul Hossain, Lecturer, Department of Software Engineering for her patience, motivation, enthusiasm and immense knowledge.

we wish to express my sincere thanks to Dr. Touhid Bhuiyan, Professor & Head of Software Engineering department for his constant encouragement.

We should also like to thank (“Auto Address4,” MMBD Center”) two companies who helped us with their valuable time, resource and requirements for this project.

We take this opportunity to record our sincere thanks to all the faculty members of the Department of Software Engineering for their help and encouragement.

Last but not least, we should like to thank to my parents for their unconditional support and love.

APPROVAL

This thesis titled on “Committee Management System”, submitted by **Md. Kamrul Hasan, 151-35-917** and **Md. Nazmul Rubel, 151-35-1069** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor & Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Head

Examiner

Internal

Examiner

Internal

Examiner

External

DECLARATION

It hereby declares that this thesis has been done by us under the supervision of **Ms. Syeda Sumbul Hossain**, Lecturer, Department of Software Engineering, Daffodil International University. It is also declared that neither this thesis nor any part of this has been submitted elsewhere for award of any degree.

Md. Kamrul Hasan

Student ID: 151-35-1049

Batch: 16th

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Md. Nazmul Rubel

Student ID: 151-35-1069

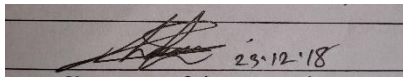
Batch: 16th

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:



Ms. Syeda Sumbul Hossain

Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Table Of Contents

ACKNOWLEDGEMENT.....	ii
APPROVAL	iii
DECLARATION	iv
Executive Summary.....	1
CHAPTER-1: INTRODUCTION	2
1.1. Project Overview	2
1.2. The Purpose of the Project	2
1.2.1 Background	2
1.2.3. Benefits & Beneficiaries.....	2
1.2.4. Goals	2
1.3. Stakeholders.....	3
1.4. Project Schedule	3
1.4.1. Gantt Chart	3
1.4.2. Milestone	4
CHAPTER-2: SOFTWARE REQUIREMENT SPECIFICATION.....	5
2.1 Functional Requirements	5
2.1.1. Registration (FRQ001).....	6
2.1.2. Login (FRQ002).....	6
2.1.3. Profile (FRQ003).....	6
2.1.4. Admin (User) (FRQ004).....	6
2.1.5. Staff (User) (FRQ005)	6
2.1.6. Create Committee (FRQ006)	7
2.1.7. Manage Committee (FRQ007)	7
2.1.8. Leader (FRQ008)	7
2.1.9. Member (FRQ009)	7
2.1.10. Manage Event (FRQ010)	7
2.1.11. Manage Post (FRQ011)	7
2.1.12 Slide (FRQ012)	7
2.2. Non-functional Requirements.....	8
2.2.1. Available for faculty members (NFRQ001)	8
2.2.2. Access requirements (NFRQ002)	8
2.2.3. Fast browsing (NFRQ003).....	8
2.2.4. Good UID(NFRQ004).....	8
2.2.5. Screen responsible (NFRQ005)	8
CHAPTER-3: SYSTEM ANALYSIS	9
3.1. Use Case Diagram	9
3.2 Use Case Description (for each use case):	10
3.2.1. Module-1: Admin	10
3.2.2. Leader:.....	13
3.2.3. Member:.....	16
3.3 Activity Diagram:	18

3.3.1. Activity Diagram Admin Perspective	18
3.3.2. Activity Diagram Leader and Member Perspective	19
CHAPTER-4: SYSTEM DESIGN SPECIFICATION	20
4.1 Sequence Diagram	20
4.1.1. Admin Perspective Sequence Diagram	20
4.1.2. Leader Perspective Sequence Diagram	20
4.1.3 Member Perspective Sequence Diagram	21
4.2. Class Diagram	22
4.3. Database Design Diagram	22
4.3.1. Entity Relationship Diagram (ERD)	23
4.4. Development Tools & Technology	26
4.4.1. User Interface Technology	26
4.4.2. Implementation Tools & Platforms	26
CHAPTER-5: SYSTEM TESTING	27
5.1. Testing Features	27
5.1.1. Features to be tested	27
5.1.2. Features not to be tested	27
5.2. Testing Strategies	27
5.3. Testing Environment (hardware/software requirements)	27
5.4. Test Cases	28
5.4.1. Test Case Module-1: Registration	28
5.4.1.1. State Transition Diagram	28
5.4.2. Test Case Module-2: Login	29
5.4.3. Test Case Module-1: Committee creation	30
5.4.4. Test Case Module-4: modify member	31
CHAPTER-6: USER MANUAL	33
6.1. Registration Page	33
6.2. Home Page	33
6.3. Profile	34
6.4. Committee List	35
6.5. Event detail view	35
CHAPTER-7: PROJECT SUMMARY	36
7.1. GitHub Link	36
Link: [1] https://github.com/kamrulhasanp/committee_management	36
7.2. Critical evolution	36
7.3. Limitations	36
7.4. Obstacles & Achievements	36
7.5. Future Scope	36
References	37

List of tables

<u>Table 1 .1:Project Milestone</u>	7
<u>Table 2 1:Functionalities</u>	8
<u>Table 2.2:Non-functional Requirement</u>	12
<u>Table 5. 1: Test Case Registration</u>	41
<u>Table 5 2:</u>	41
<u>Table 5 3: Test Case Login</u>	42
<u>Table 5 4</u>	43
<u>Table 5 5: Test Case Committee creation</u>	43
<u>Table 5 6:</u>	44
<u>Table 5 7: Test Case modify member</u>	44
<u>Table 5 8:</u>	45

List of Figure

<u>figure 1. 1:Block Diagram</u>	6
<u>figure 1. 2: Gantt Chart</u>	7
<u>Figure 3. 1:Use Case Diagram</u>	14
<u>Figure 3. 2: Admin Use Case Diagram</u>	15
<u>Figure 3. 3: Use Case Diagram for Admin</u>	16
<u>Figure 3. 4: Use Case Diagram for Admin Manage Committee</u>	17
<u>Figure 3. 5: Use Case Diagram for Admin Create Slider</u>	17
<u>Figure 3. 6: Use Case Diagram for Admin Modify Member</u>	18
<u>Figure 3. 7: Use Case Diagram for Event of Admin</u>	18
<u>Figure 3. 8: Use Case Diagram for Admin Manage Post</u>	19
<u>Figure 3.2. 1: Use Case Diagram for Committee Leader</u>	21
<u>Figure 3.2. 2: Use Case Diagram for Leader Authentication</u>	22
<u>Figure 3.2. 3: Use Case Diagram for Modify Member</u>	22
<u>Figure 3.2. 4: Use Case Diagram for Leader Events</u>	23
<u>Figure 3.2. 5: Use Case Diagram for Leader Manage Post</u>	24
<u>Figure 3.2. 6: Use Case Diagram for Member</u>	25
<u>Figure 3.2. 7: Use Case Diagram for Member Authentication</u>	25
<u>Figure 3.2. 8: Use Case Diagram for Admin Manage Post</u>	26
<u>Figure 3.3. 1: Activity Diagram for Admin Perspective</u>	28
<u>Figure 3.3. 2: Activity Diagram for Leader and Member Perspective</u>	30
<u>Figure 3.4. 1: System Sequence Diagram for Admin Perspective</u>	32
<u>Figure 3.4. 2: System Sequence Diagram for Leader Perspective</u>	33
<u>Figure 3.4. 3: SystemSequence Diagram for Member Perspective</u>	34
<u>Figure 4.1. 1 : Sequence Diagram for Admin</u>	33
<u>Figure 4.1. 2 : Sequence Diagram for Leader</u>	34
<u>Figure 4.1. 3 : Sequence Diagram for Member</u>	35
<u>Figure 4.2. 1 : Class Diagram</u>	36
<u>Figure 4.3 1 : Class Diagram</u>	38
<u>Figure 5.4. 1 : State Transition Diagram for Registration</u>	42
<u>Figure 5.4. 2: State Transition Diagram for Login</u>	43
<u>Figure 5.4. 3 : State Transition Diagram for Committee Creation</u>	44
<u>Figure 5.4. 4 : State Transition Diagram for Modify Member</u>	45
<u>Figure 6. 1Registration Page</u>	50
<u>Figure 6. 2: Home Page</u>	50
<u>Figure 6. 3: Profile</u>	51
<u>Figure 6. 4: Committee list</u>	52
<u>Figure 6. 5: Event detail</u>	53

Executive Summary

We create Committee Management System Based on our software department committees. Now when one committee held an event it clashes with another committee event. So Committee Management System' is very helpful for Software Engineering department to keep their activity record so easily. So that all committee member and leader can manage and also track other event. Student and teacher can see every activity under a committee. Other all user can create post every event. So that a committee will be more active, and every event will be success and efficient. We use analysis of field data for making this system. We find 60 percent of people are facing problems of event clash when they are member of different committee and committees hold two different event same time. Committee Management System will help member to track and find all event of different committee and when leader create an event he/she cannot select same time when another event selected this time already. This report will be very helpful for developers if the system is going for further development in future.

CHAPTER-1: INTRODUCTION

1.1.Project Overview

‘Committee Management System’ is very helpful for Software Engineering department to keep their activity record so easily. Here, admin and staff user can create our departmental committee and event. Student and teacher can show every activity under a committee. Other all user can create post every event. So that a committee will be more active, and every event will be success and efficient.

1.2. The Purpose of the Project

Software Engineering department has many several committees. Every committee has organized many events. But currently they are working randomly, as a result many conflicts has occurred. Committee Management System will help maintain this problem conveniently without any clash.

1.2.1 Background

In the software engineering department has many different committees. But they are working randomly, as a result many teacher and student do not know how much committee have in our department, who is the committee member or leader and what is they are activity. Our honorable lecturer Ms. Syeda Sumbul Hossain thought we must solve the current problem to handle different internal departmental committees. Then she gives me that responsibility to build a system for managing our departmental committee. Since then I'm still trying to develop this system.

1.2.3. Benefits & Beneficiaries

The committee management system is developed for our faculty members and students. If we use this system to manage our committees than we can see all committee in page. We also can see all activities of a certain committee. It will help to increase activities of a committee. Students and teachers can to know who the committee leader is and who are the committee members. Student easily can to know about an event, so an event will be more efficient and success. After all we can say Benefit is manage to easy the department committee and Beneficiaries are students and teachers.

1.2.4. Goals

The basic functionality of this system is managing the departmental committee and making the process user friendly. And to reach with every student about a committee. It helps admin to control the committee management system easily.

1.3. Stakeholders

A stakeholder is either group, an individual, organization or a person who is impacted by the effect of a project. customer also a stakeholder. In our project stakeholders are teachers, student, admin and software engineering department. Here stakeholders are Software engineering department: - Software engineering department is a stakeholder because this system will use our department.

Teacher: - In the system will use teachers and leader and member will be from teacher, so teacher is stakeholder.

Student: - Since student will get benefit from the system, so student is also stakeholder.

Admin: - Admin also a stakeholder, because admin will manage and control this system.

1.4. Project Schedule

Project scheduling is a mechanism to about what tasks need to get done in the certain time.

1.4.1. Gantt Chart

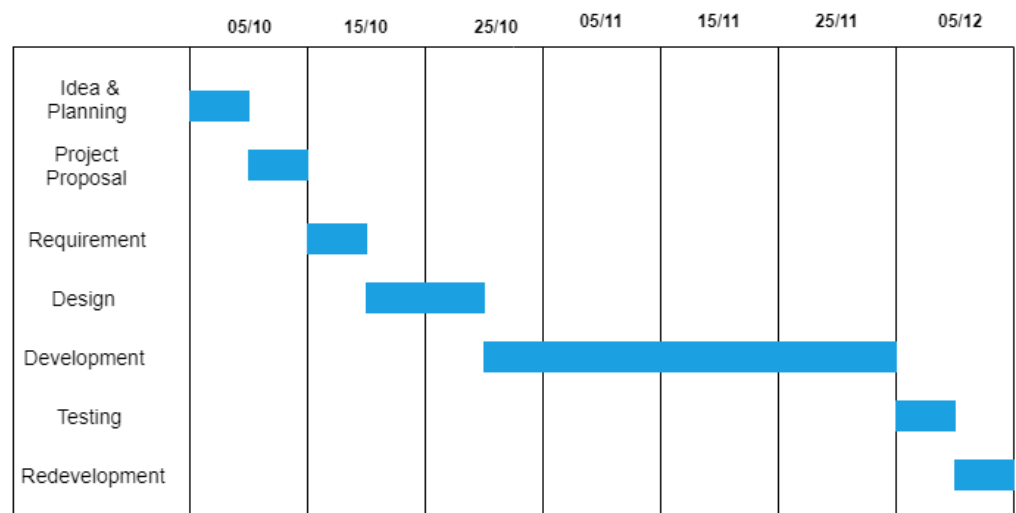


figure 1. 1: Gantt Chart

1.4.2.Milestone

Table 1.1:Project Milestone

Milestone	Descriptions	Milestone criteria	Planned date
M1	Proposal the Project description detail.	Proposal the Project description detail.	10/09/2018
M2	Requirement analysis, Collect all functional requirement and non-functional requirements.	Specify and finalized requirements.	15/09/2018
M3	System design, Design all user interface.	Finalized all design, Architecture reviewed	25/09/2018
M5	Implementation, Coding of functionalities	Buildup System, Architecture reviewed	08/10/2018
M6	Testing, System test Pass or fail	System tested.	25/11/2018
M7	Reimplementation, Coding of error functionalities	Buildup System, Architecture reviewed	02/12/2018
M8	Implementation, <milestone description> Overview of the project.	Document reviewed.	08/12/2018

CHAPTER-2: SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requirements

Functional requirements are expressed in the form “specifies something the system must do”. Functional requirements drive the application architecture of a system. Requirements from the user or stakeholders. The functionality they wanted.

Table 2 1:Functionalities

FRQ ID	FRQ Name	Descriptions	Constraint
FRQ001	Registrations	The module will collect data from users for registration.	N/A
FRQ002	Login	This module will provide the report only for registered users	FRQ001
FRQ003	Profile	This module will be addressing a single user.	FRQ001
FRQ004	Admin (User)	The admin is all in all.	FRQ001
FRQ005	Staff (User)	The staff users have less access than admin user	FRQ001
FRQ006	Create Committee	The module will be able to create a committee	FRQ004, FRQ005
FRQ007	Slide	This module will represent a committee	FRQ006
FRQ008	Manage Committee	The module will provide for manage a committee	FRQ004, FRQ005

FRQ009	Leader	Leader will lead a committee	FRQ001
FRQ0010	Member	Member will be adviser a committee	FRQ001
FRQ011	Manage Event	This module will be able to create event under a committee.	FRQ009, FRQ0010
FRQ012	Manage Post	This module will be able to manage a post under an event.	FRQ009, FRQ0010

2.1.1. Registration (FRQ001)

Feature:

The module will collect data from users for registration. Add new user. Detect this providing email is match with '@diu.edu.bd' domain.

2.1.2. Login (FRQ002)

Feature:

This module will provide the report only for registered users. When a user successfully registered then he or she can login the system.

2.1.3. Profile (FRQ003)

Feature:

This module will be addressing a single user. Here every user will be teacher that's Why their profile will contain Teacher ID, Teacher Designations etc.

2.1.4. Admin (User) (FRQ004)

Feature:

The admin is all in all. Admin can control everything in system. Admin can create one or more staff user for moderating this system. And the other think is admin only can create committee and select leader and member.

2.1.5. Staff (User) (FRQ005)

Feature:

The staff users have less access than admin user. Staff user not a like admin user. Staff user makes by admin for help him or her. Staff will get a power for creating committee and select leader and member.

2.1.6. Create Committee (FRQ006)

Feature:

The main purpose of this system to creating Committee. The committee contain some leader and some and some members from users. Admin and Staff user only can create a committee.

2.1.7. Manage Committee (FRQ007)

Feature:

It is necessary to management after Creating a committee. Admin and Staff user only can manage a committee like change, add or remove committee leader or member. Leader also can moderate members. They can also create, delete or update an event and post under a committee.

2.1.8. Leader (FRQ008)

Feature:

Every committee has one or more leader. A leader can moderate member in this committee. He or she also can create a or more Event and can create post under an event.

2.1.9.Member (FRQ009)

Feature:

Every committee has more than one member. A member can create a or more Event and can create post under an event.

2.1.10. ManageEvent (FRQ010)

Feature:

Every committee will organize many events in several time. This Event can manage admin, staff user, leader and member.

2.1.11. Manage Post (FRQ011)

Feature:

Under an Event in a committee every user can post. And user also can post, delete and update.

2.1.12 Slide (FRQ012)

Feature:

Every committee page has a header slide view. There will scrolling some photos of committee related with title and description of photos.

2.2. Non-functional Requirements

Table 2.2:Non-functional Requirement

NFRQ_ID	NFRQ_Name	Constraint
NFRQ001	Available for faculty members	N/A
NFRQ002	Access Requirements	N/A
NFRQ003	Fast browsing	N/A
NFRQ004	Good UID	N/A
NFRQ005	Screen responsible	N/A

2.2.1. Available for faculty members (NFRQ001)

The system will be only for faculty members of software engineering department.Only faculty members can manage and control this system.

2.2.2. Access requirements (NFRQ002)

Admin and super user will can access everything in this system. But Staff user has a specific access power to create committee and modify members and leader can also modify members. Other all users can only view this system.

2.2.3.Fast browsing (NFRQ003)

The client site server must be faster and smother in this system. If client site server is not faster than client will be bored. But it is not our business. That's why client site server must be faster.

2.2.4. Good UID(NFRQ004)

At present user interface design (UID) is the most important thing in the software engineering world. If your UID is not good looking than your system users will not accept this easily. Nowadays users want a smart design, the system may not so good. That's why the UID must be looking good for a system.

2.2.5. Screen responsible (NFRQ005)

Nowadays users use different device. as a result, users facing a problem about user interface. Now we should develop a system that will maintain all device screen responsible accurately.

CHAPTER-3: SYSTEM ANALYSIS

3.1. Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.[A]

In this system has three actors, there are 1. admin2. Leader and 3. member. this is the main use case diagram. I will describe more detail in below.

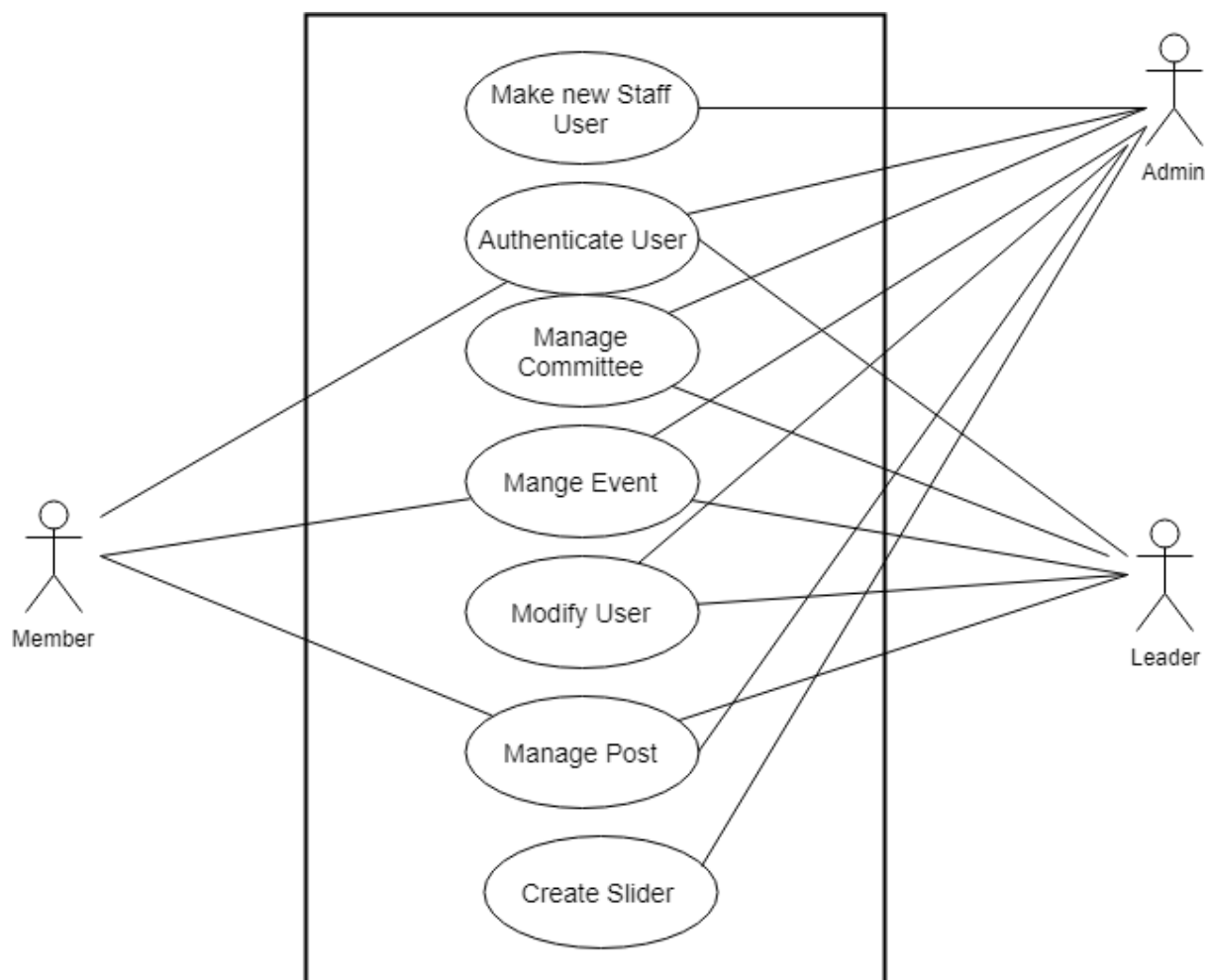


Figure: Use Case Diagram

Figure 3. 1:Use Case Diagram

3.2 Use Case Description (for each use case):

3.2.1.Module-1: Admin

A use case defines the relation between external user and system user. Actors must be able to make decisions, but actor need not to be a human. A user or actor might be a person, company, organization, computer system or a computer program hardware, software or both of hardware and software. Obviously, an actor are always stakeholders. Admin actor might be a computer program or human. But major time human is a admin. Below is a detailed discussion of the admin.

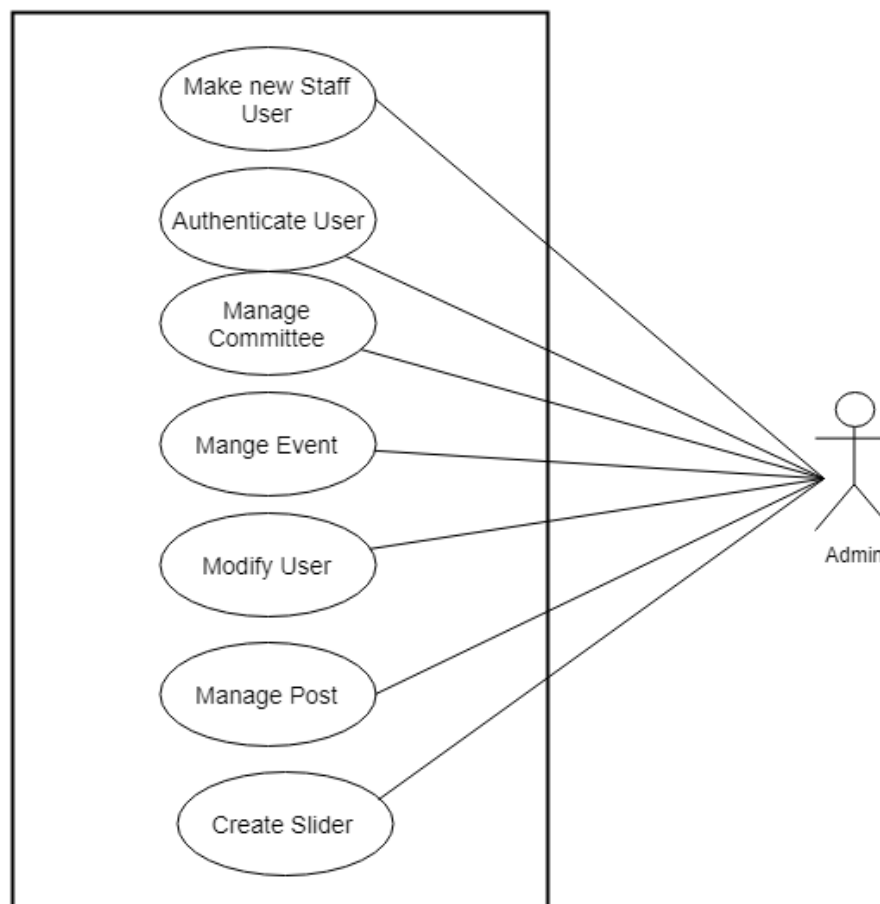


Figure: Use Case Diagram

Figure 3. 2: Admin Use Case Diagram

3.2.1.1. Admin Authentications:

An admin user must be an authenticated user. That means admin must be registered user. Every registered user might not be an admin. In this system admin is who operate the computer system for managing the committee management system. The admin registration system not an external option it is totally internal option.

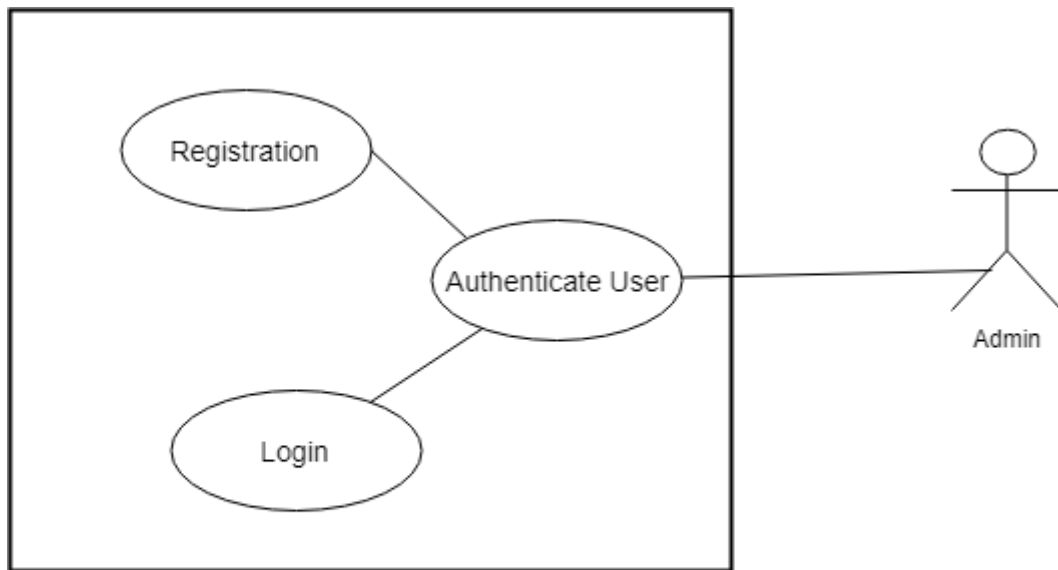


Figure: Use Case for Admin

Figure 3. 3: Use Case Diagram for Admin

3.2.1.2 Admin Manage Committee:

The main business of this system is creating a committee and manage it. Only admin can able to create a committee and manage it. Committee manage means create a committee. After creating a committee admin can be update and delete it.

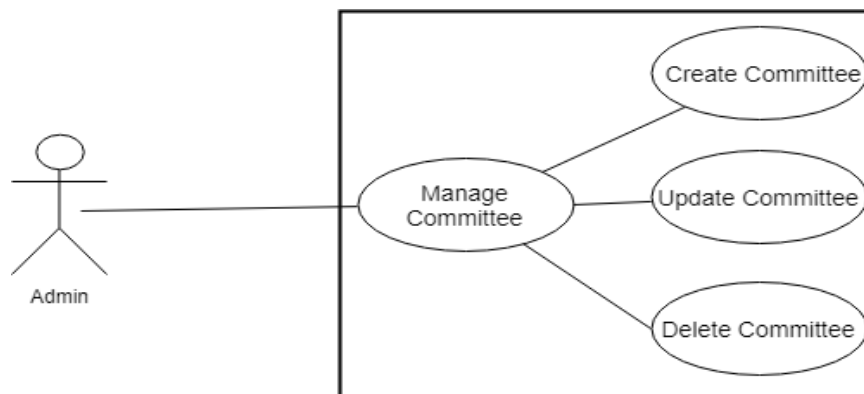


Figure: Use Case for Admin

Figure 3. 4: Use Case Diagram for Admin Manage Committee

3.2.1.3.Create Slider:

Slider is not main business in this system. But it will help to good user interface design. The slider will be scrolling image in the committee header site. The admin can create and update a slider. The admin actor can add infinity image in the slider option.

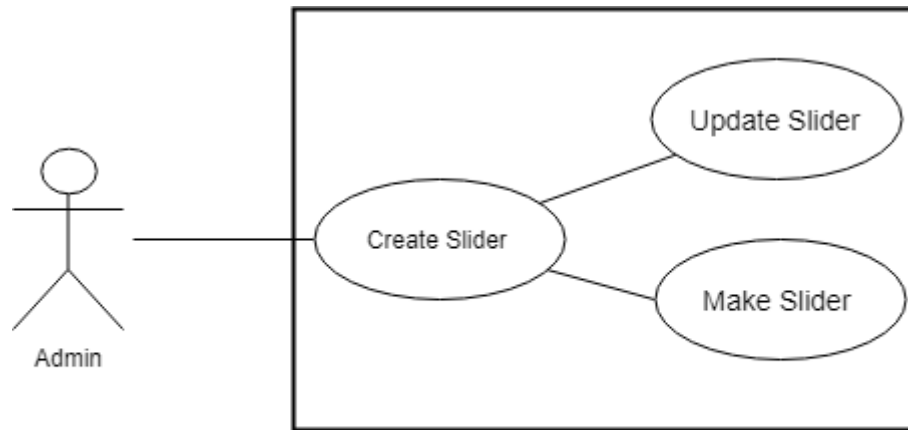


Figure: Use Case for Admin

Figure 3. 5: Use Case Diagram for Admin Create Slider

3.2.1.4. Modify member

Every committee has some committee members. But sometimes need to change or modify member. The admin will modify members

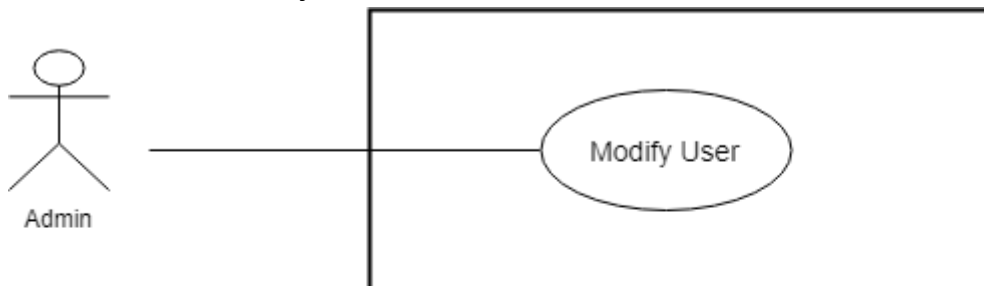


Figure: Use Case for Leader

Figure 3. 6: Use Case Diagram for Admin Modify Member

3.2.1.5. Admin Event:

In the several time committees has organize many different event or workshop. An admin can create an event under committee. In the event model has image, title and description attribute.

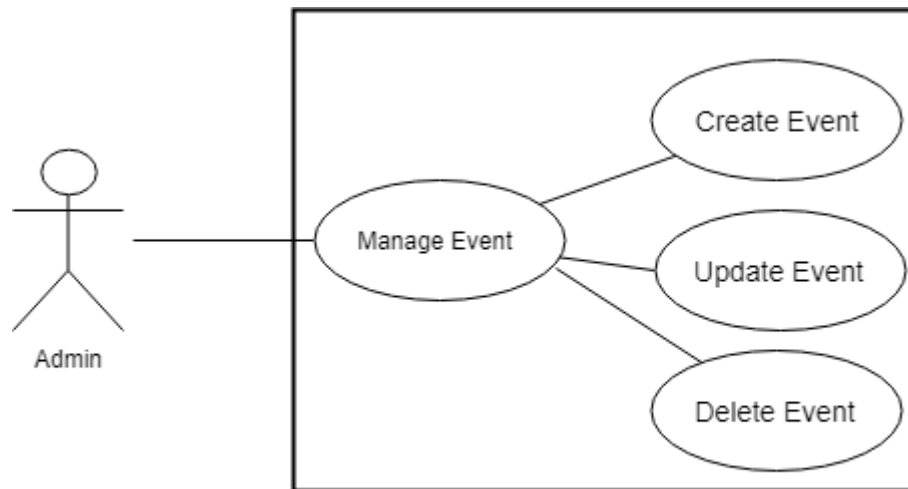


Figure: Use Case for Admin

Figure 3. 7: Use Case Diagram for Event of Admin

3.2.1.6 Admin Manage Post:

Another option is managed post for admin site in this system. we saw the previous module admin created an event. But that was an empty event. Need to create post for fulfilling the event. Admin can create post.admin also can update and delete a post.

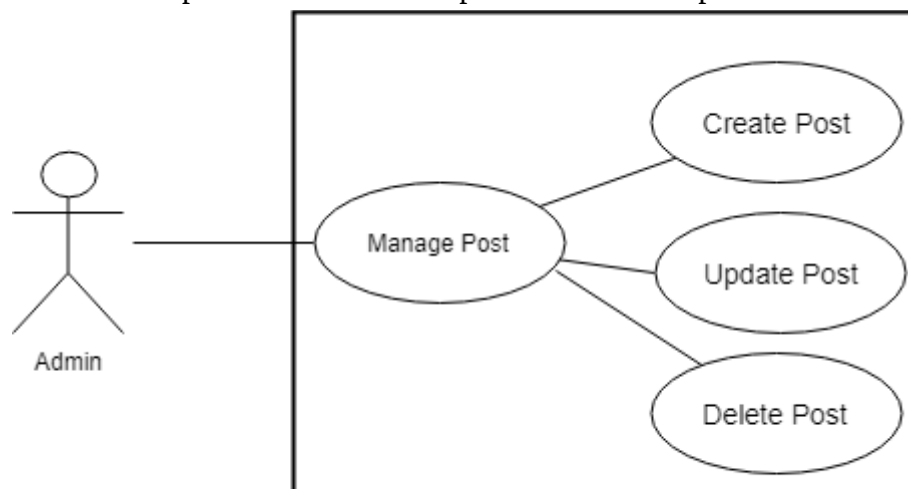


Figure: Use Case for Admin

Figure 3. 8: Use Case Diagram for Admin Manage Post

3.2.2. Leader:

We saw the previous section the admin can create a committee. But a committee has one or more leader. Leader is an external user and he is also an actor. An actor represent relation with a system. A system might be computer program, computer system etc. Below is a detail discussion of the Leader actor.

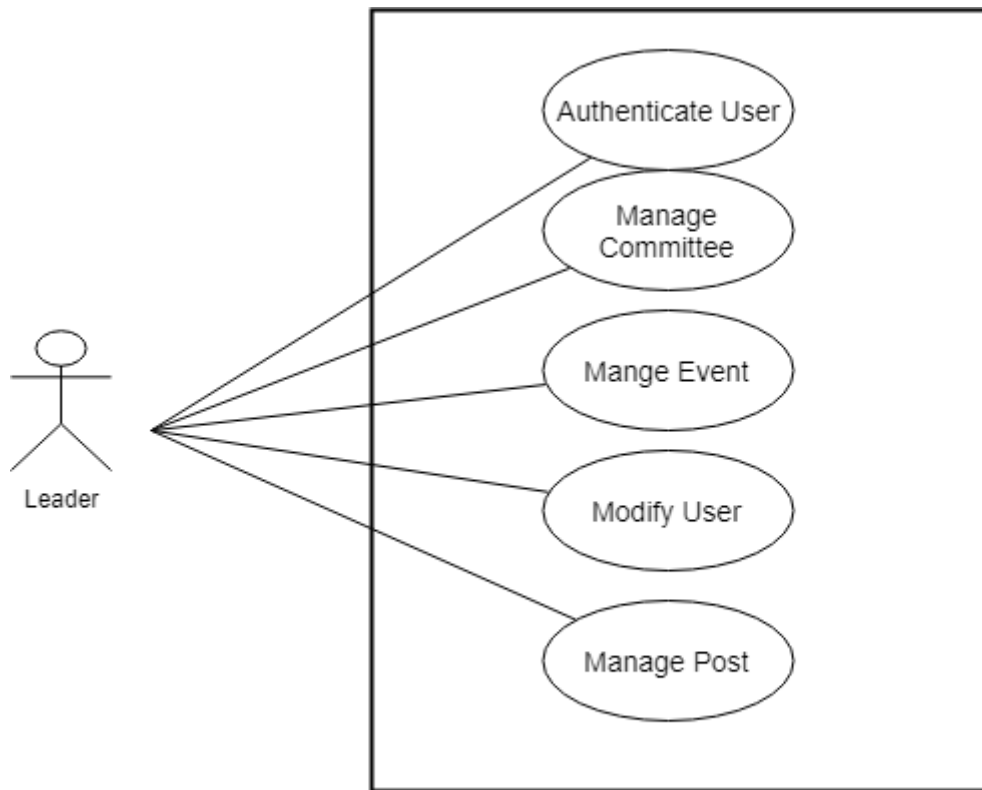


Figure: Use Case for Leader

Figure 3.2. 1: Use Case Diagram for Committee Leader

3.2.2.1. Leader Authentications:

A leader user must be an authenticated user. That means leader must be a registered user. Every registered user might not be a leader. In this system leader is who can lead a committee in managing the committee management system. The leader must register with daffodil email. A leader will be authenticated after registration and login

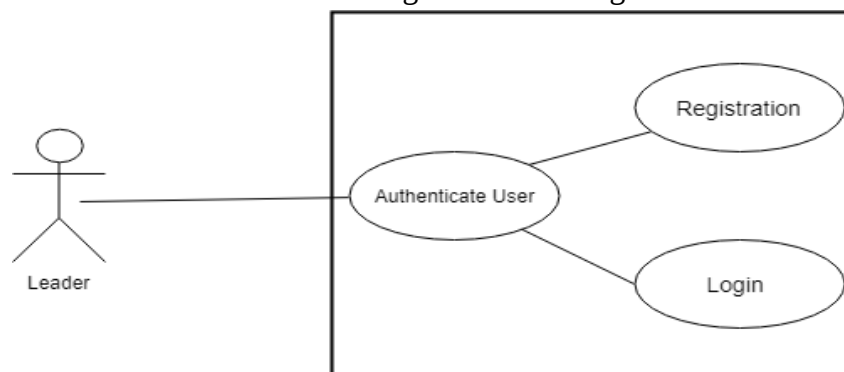


Figure: Use Case for Admin

Figure 3.2. 2: Use Case Diagram for Leader Authentication

3.2.2.2. Modify Member:

Every committee has some members. But sometimes need to change or modify member. The leader can modify members which committee's leader he or she. He cannot modify another committee member.

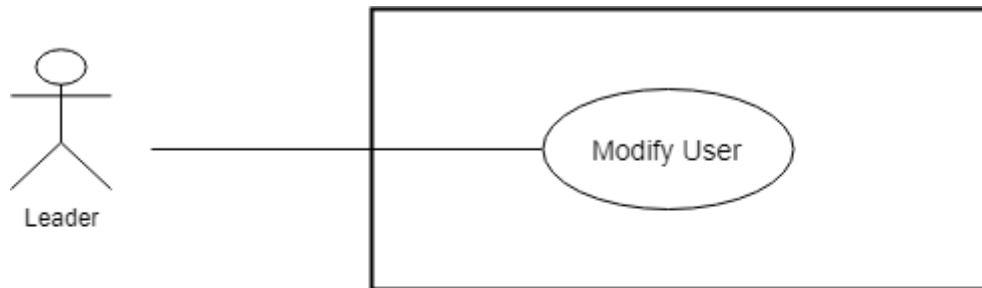


Figure: Use Case for Leader

Figure 3.2. 3: Use Case Diagram for Modify Member

3.2.2.3. Leader Events:

In the several time committees has organize many different event or workshop. A leader can create an event under committee which committee belong he. In the event model has image, title and description attribute.

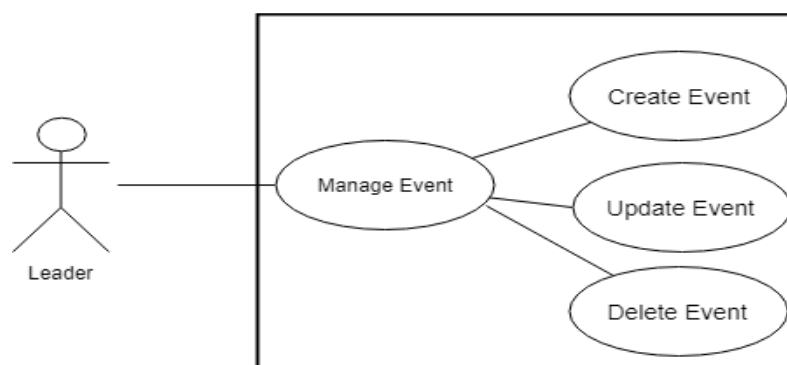


Figure: Use Case for Admin

Figure 3.2. 4: Use Case Diagram for Leader Events

3.2.2.4. Leader-Manage Post

Another option is managed post for leader in this system. we saw the previous module leader created an event. But that was an empty event. Need to create post for fulfilling the event. Leader can create post. leader also can update and delete his ownpost which

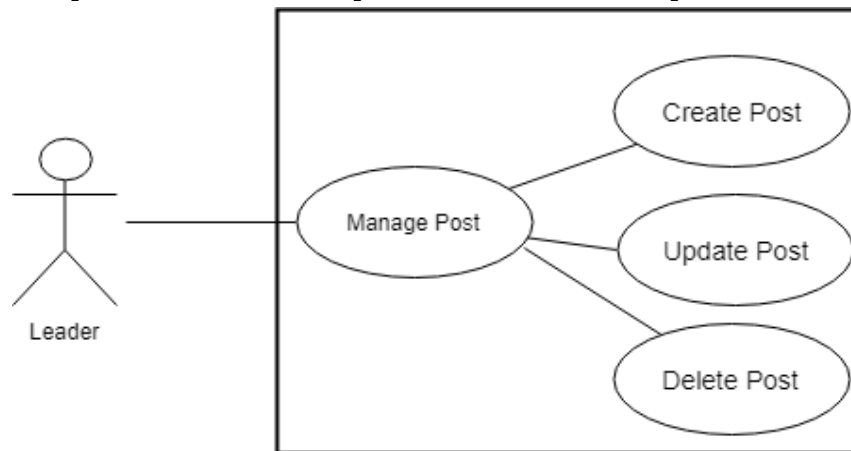


Figure: Use Case for Admin

Figure 3.2. 5: Use Case Diagram for Leader Manage Post

3.2.3. Member:

Member is another actor in this system. Member actor will describe relation between member actor and the system. In a committee can be many members. Below is a detailed discussion of the member

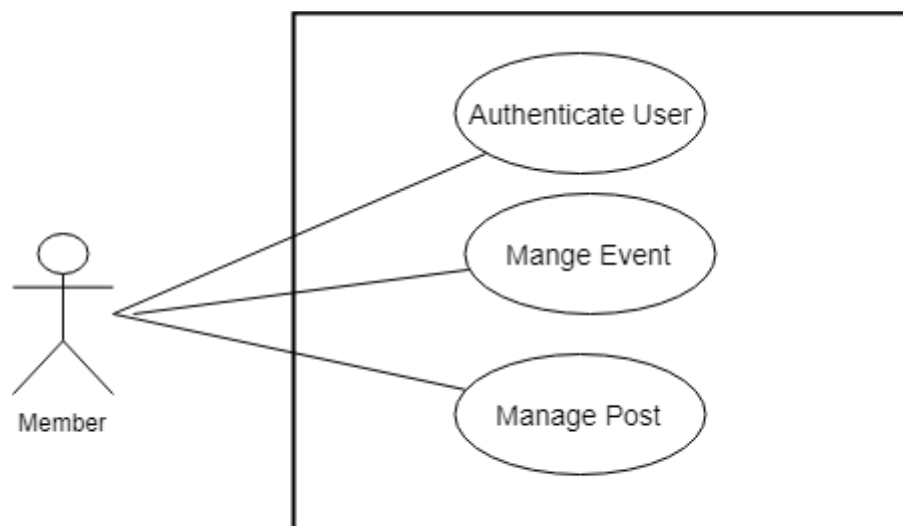


Figure: Use Case for Member

Figure 3.2. 6: Use Case Diagram for Member

3.2.3.1. Authentication:

A member user must be an authenticated user. That means member must be a registered user. Every registered user might not be a member. In this system member is who is gain the membership in a committee in managing the committee management system. The member must register with daffodil email. A member will be authenticated after registration and login

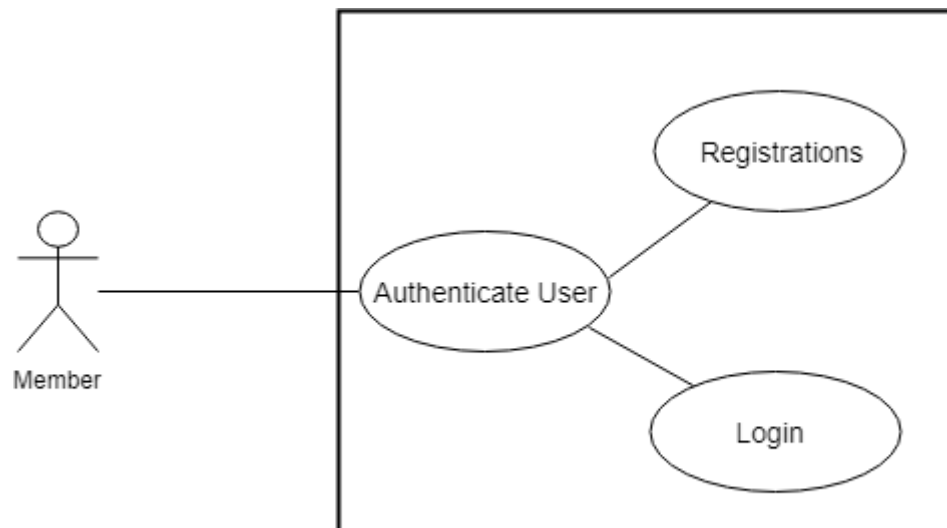


Figure: Use Case Diagram

Figure 3.2. 7: Use Case Diagram for Member Authentication

3.2.3.2. Manage Post:

Another option is managed post for member in this system. we saw the previous module leader and admin created an event. But that was an empty event. Need to create post for fulfilling the event. Member can create post. Member also can update and delete his own post. Member cannot do other work without the manage post.

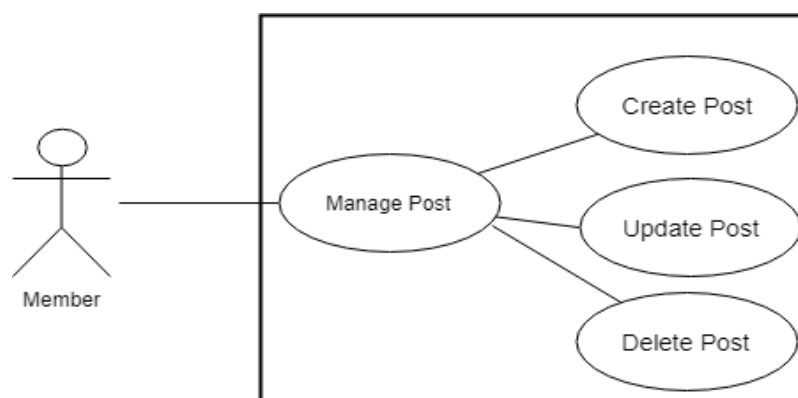


Figure: Use Case for Admin

Figure 3.2. 8: Use Case Diagram for Admin Manage Post

3.3 Activity Diagram:

Activity diagram is the most important diagram in UML to describe the dynamic aspect of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. That's why Activity Diagram is very important. The activity can be described as an operation of the system.[B].

3.3.1. Activity Diagram Admin Perspective

For admin perspective when system starts admin try to login this system. System check is the user an admin or not. If it is admin it will show home page otherwise it will send back to login. If admin login to this system admin can create committee and view all committee. When admin create a new committee, admin can assign leader and member. Then create event under a committee. Admin can post in an event modify an event delete an event, also update a post delete a post under an event. An admin also modify member and create slider.

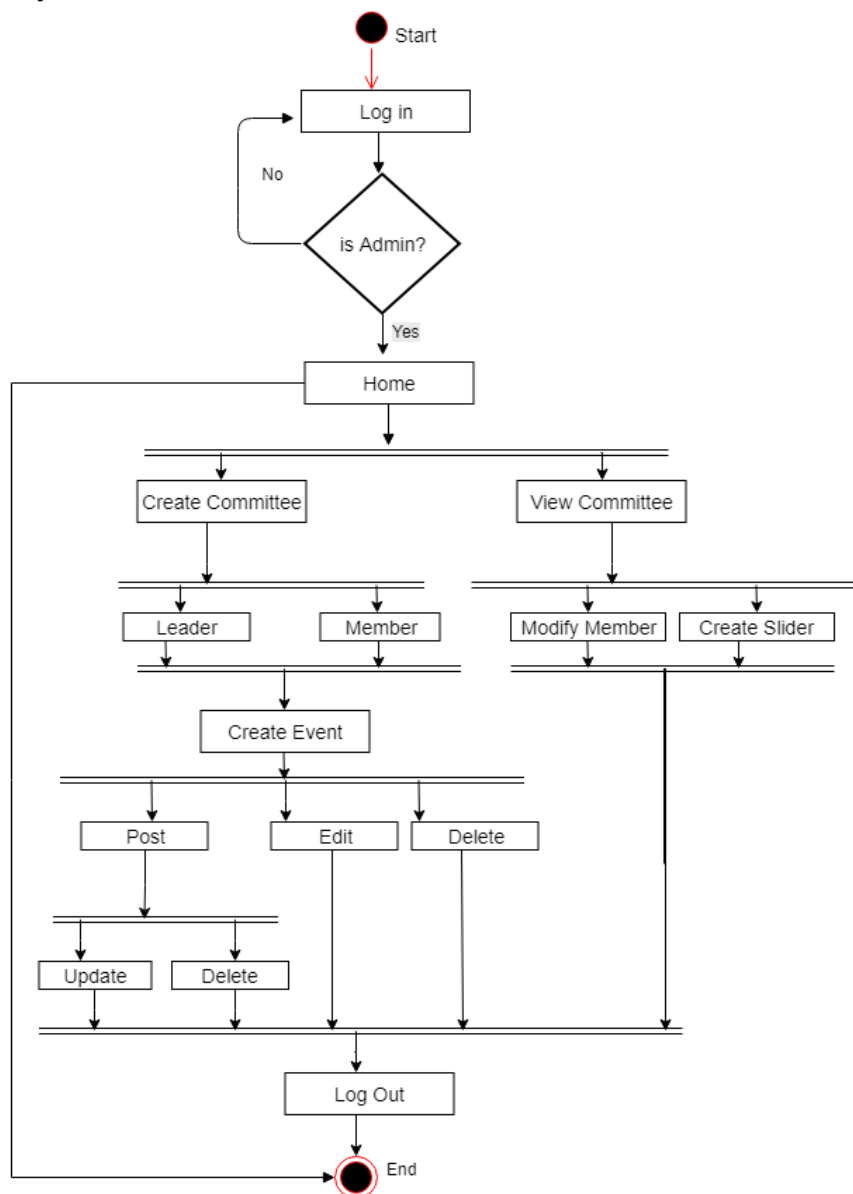


Figure 3.3. 1: Activity Diagram for Admin Perspective

3.3.2. Activity Diagram Leader and Member Perspective

For leader and member purpose when user open this system it ask user if he/she is a existing user or new user. If new user it requires to Register first. On registration when he enter his/her email first it checked by the system if it is diu email or not. If this is diu email user can register otherwise he/she cannot use this system. When a user registered successfully it takes him/her to the home page. User can view all committee. If he/she is a leader he/she can modify member, create event. Under an event he/she can post, update a post, delete a post. If he/she is a member he/she can create event. Under an event he/she can post, update a post, delete a post. When user is done he/she can log out from this system.

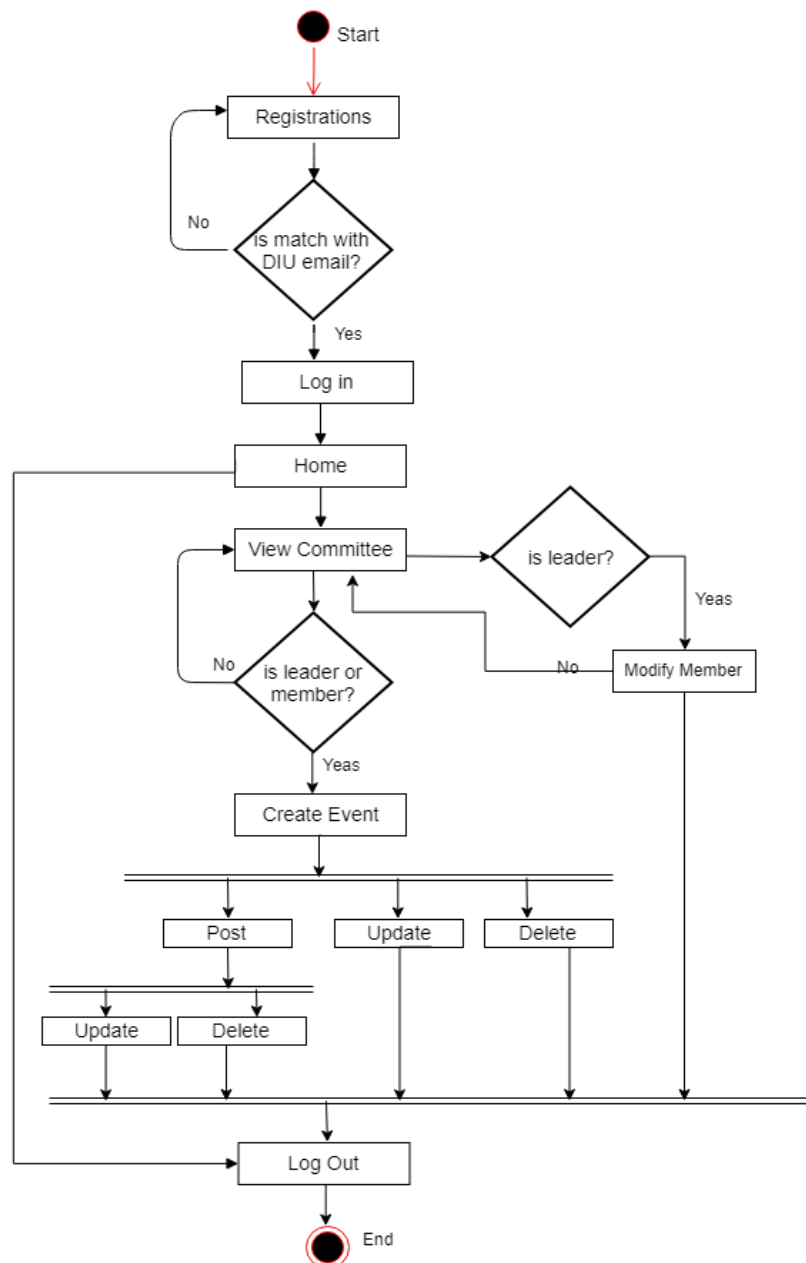


Figure 3.3. 2: Activity Diagram for Leader and Member Perspective

CHAPTER-4: SYSTEM DESIGN SPECIFICATION

4.1 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system. In simpler words, a sequence diagram shows different parts of a system work in a 'sequence' to get something done.[C]

4.1.1. Admin Perspective Sequence Diagram

Admin have access to Registration, Login, Home, Create Committee, Select Leader, Select Member, Modify Member, Create Event, Create Post, Create Slide.

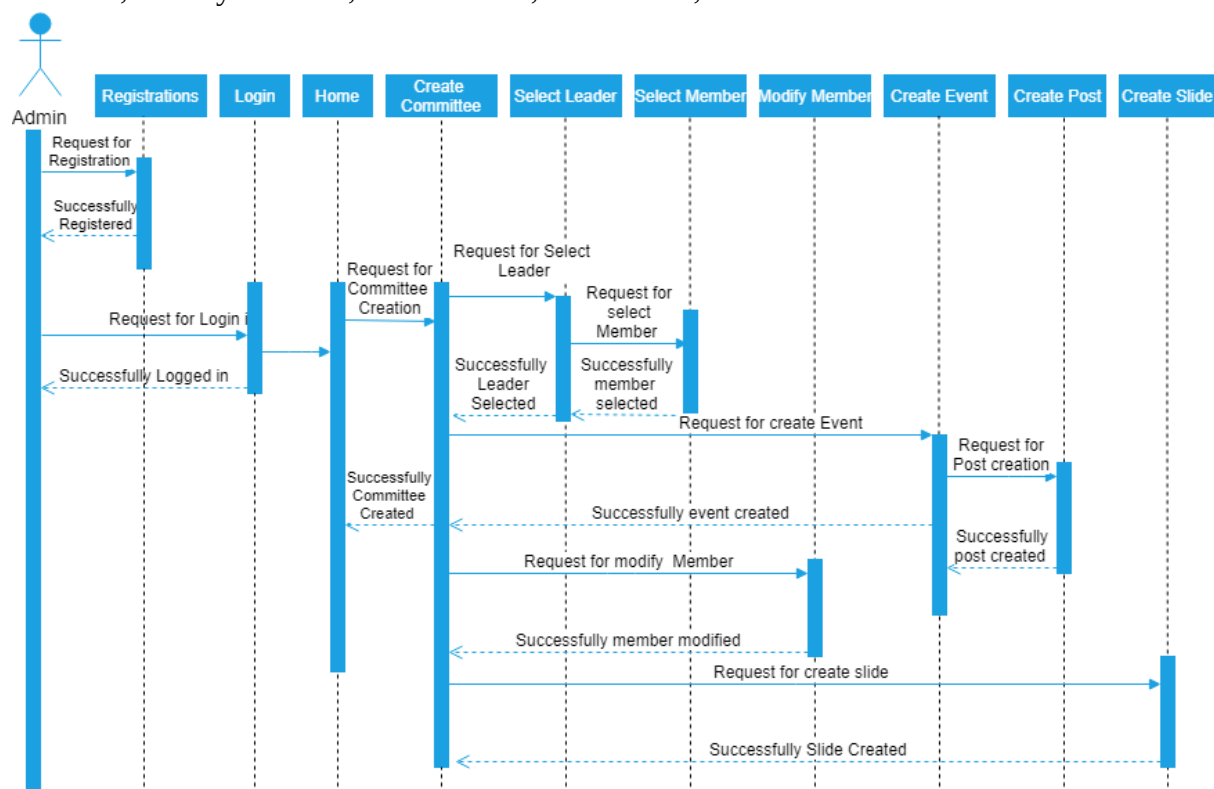


Figure 4.1. 1 : Sequence Diagram for Admin

4.1.2. Leader Perspective Sequence Diagram

Leader have access to Registration, Login, Home, Committee, Modify Member, Create Event, Create post.

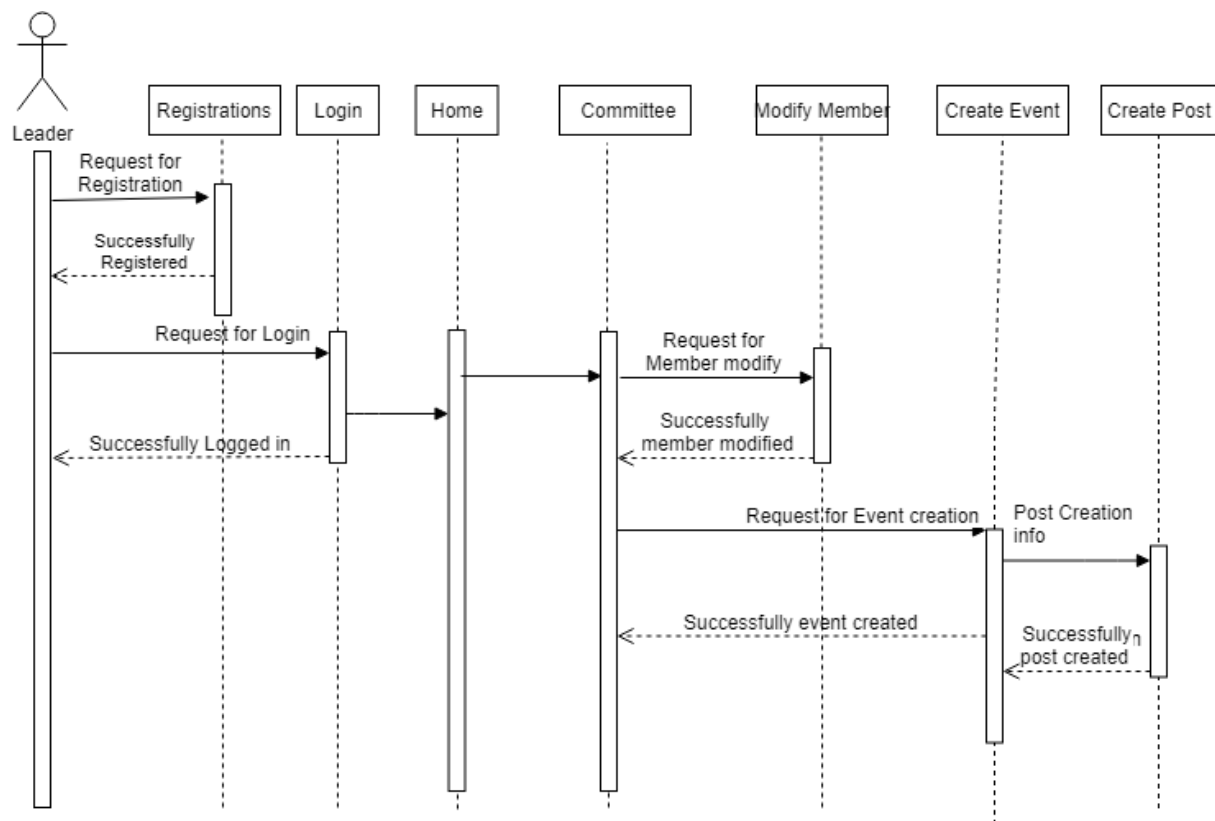


Figure 4.1. 2 : Sequence Diagram for Leader

4.1.3 Member Perspective Sequence Diagram

Member have access to Registration, Login, Home, Committee, Create Event, Create Post.

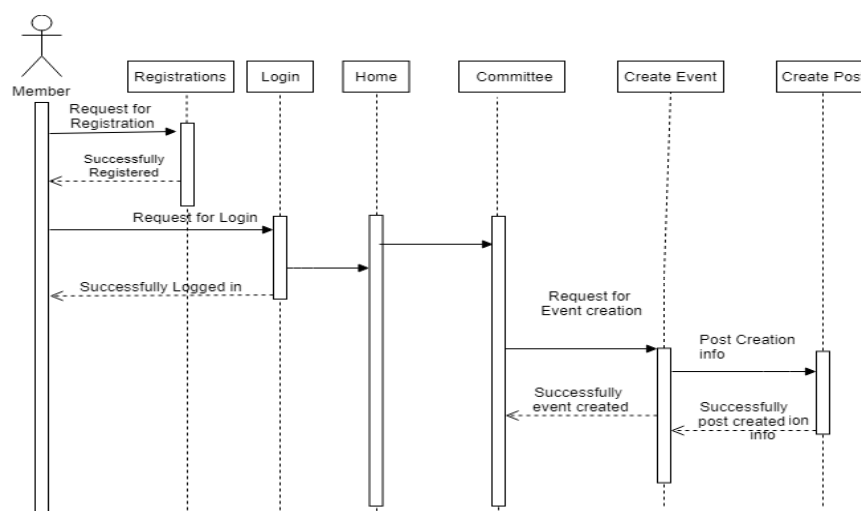


Figure 4.1. 3 : Sequence Diagram for Member

4.2. Class Diagram

Class Diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods). Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a system by modeling its classes, attributes, operations, and relationships between objects.[D]

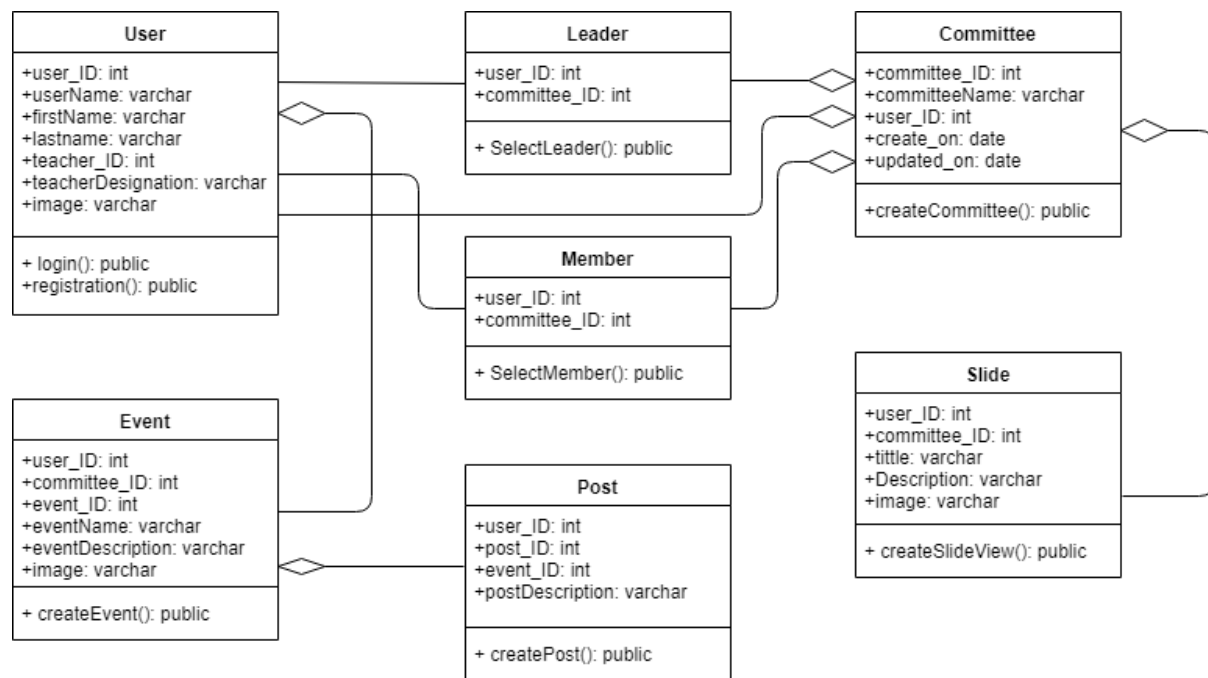


Figure 4.2. 1 : Class Diagram

4.3. Database Design Diagram

4.3.1.Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD), also called as an entity relationship model. An ERD is a data modeling technique that can help identify business processes and be used as the basement for a relational database. Now describing entity of this project.

4.3.1.1. User(model):

User model is the prime model of this system. User model has different entity.

- user_ID
- teacher_ID
- userName
- Designation
- Image
- Email
- Password

4.3.1.2. Committee(model)

Committee model is the second prime model for this system. The prime work of this project is create committee. Committee model has different entity...

- user_ID
- committee_ID
- committeeName
- Description
- Created_on
- Updated_on
- selectLeader
- selectMember

4.3.1.3. Leader(model)

In every committee has one or more leader. The leader model has two entity.

- user_ID
- committee_ID
- 4.3.1.4. Member(model)

In every committee also has some member. The member model has two entity.

- user_ID
- committee_ID

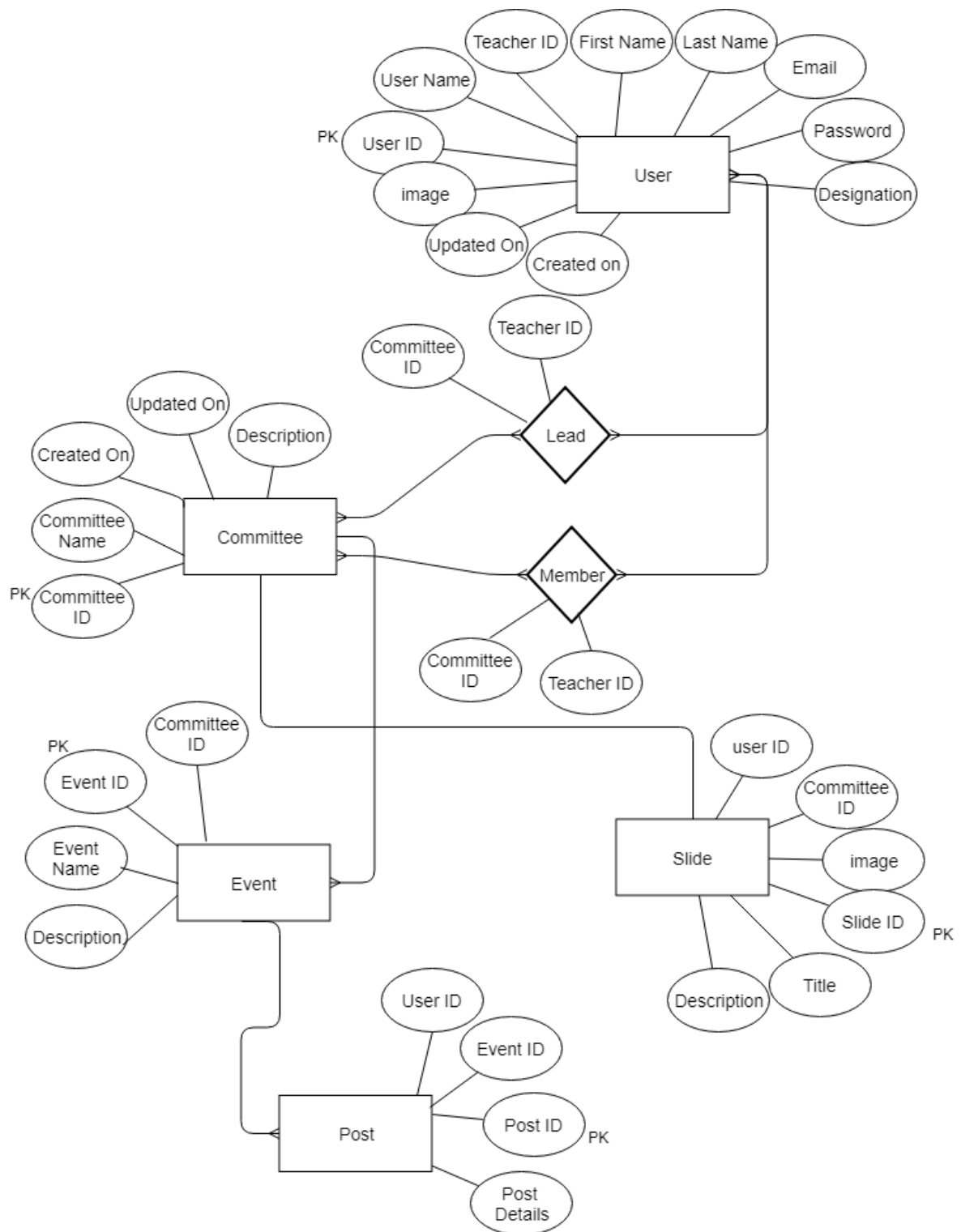


Figure: ER Diagram

Figure 4.3 1: Entity Relationship Diagram (ERD)

4.3.1.5. Event(model)

Event is another model of this system. Event model has different entity.

- committee_ID
- event_ID
- user_ID
- eventName
- Description
- image

4.3.1.6. Post(model):

Post is the last model of my system. Post model has many entities.

- event_ID
- postId
- user_ID
- Description

4.3.1.6. Slide(model):

The slide view increase to smart the UID. this is the part of a committee. Slide model has many entities.

- committee_ID
- Descriptions
- Title
- image

4.4. Development Tools & Technology

4.4.1. User Interface Technology

- Python 3.6
Python is a programming language for developing apps. Python 3.6 is the newest major release of the Python language, and it contains many new features and optimizations. It's used for many different applications. [E]
- Django 2.1
Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better Web apps more quickly and with less code. [F]
- HTML5
HTML is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard and subsumes XHTML. HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. [G]
- CSS
CSS stands for Cascading Style Sheets which take care of presentation. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. [H]
- JavaScript 1.8.5
JavaScript is a programming language commonly used in web development. It is most commonly used as a client-side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it. JavaScript makes things happen. [I]
- AJAX
AJAX stands for Asynchronous JavaScript and XML. AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. [J]
- Bootstrap 4.1
Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixings, responsive grid system, extensive pre-built components, and powerful plugins built on jQuery. Bootstrap 4.1 is the latest release from bootstrap. [K]

4.4.2. Implementation Tools & Platforms

- SQLite 3.25.0
- Pycharm-professional-2018.1.2

CHAPTER-5: SYSTEM TESTING

5.1. Testing Features

5.1.1. Features to be tested

- Registration
- Login
- Committee Creation
- Modify member

5.1.2. Features not to be tested

- Manage profile
- View Committee
- View Event
- View Post
- View leader
- View member
- Create slide
- Create post
- Create event

5.2. Testing Strategies

Testing strategies is an outline that describes the testing approach of the software development cycle. Testing strategy is one of the most powerful factors in the success of the test effort and the accuracy of the test plans. Testing strategies helps us to define the approach to test our application. [L]

5.3. Testing Environment (hardware/software requirements)

- Test Data
- Web Application
- Database Server
- Operating System
- Browser

5.4. Test Cases

5.4.1. Test Case Module-1: Registration

5.4.1.1. State Transition Diagram

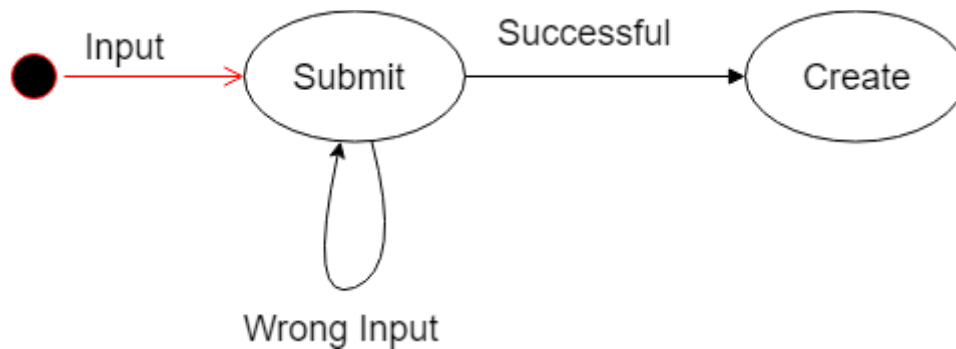


Figure 5.4. 1 : State Transition Diagram for Registration

5.4.1.2. Test Case Table

Table 5. 1: Test Case Registration

Test Case ID: TC_ASS_1	FRQ001.1
Test Design by: M. H. Masuk	RQ Name
Test Priority: High	Test Design Date: 06.12.2018
Module Name: Registration	Test Execute by: Md. Kamrul Hasan
Description: This section contain the functionality of registering a new user.	Test Execute Date: 06.12.2018

Precondition: User must be employee of Daffodil International University.

Dependencies: Enter input required information

Step	Test cases	Test Data	Expected result	P/f	Actual result
01	Enter empty value for field		Display error message	Pass	Display error message
02	Enter user name same as email	Kamrul	Your username similar with your email	Pass	Display error message
03	Enter domain except diu.edu.bd or daffodiliversity.edu.bd	Gmail.com, yahoo.com	Your email don't match with daffodil email	Pass	Display error message

04	Enter domain with daffodil email	kamrul@diu.edu.bd	View login page	Pass	Successfully login
05	New user registration		View login page	pass	View login page

5.4.2. Test Case Module-2: Login

5.4.2.1. State Transition Diagram

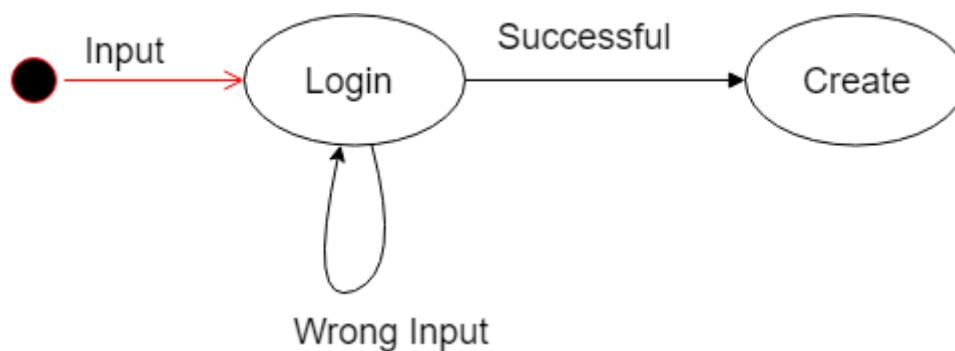


Figure 5.4. 2: State Transition Diagram for Login

5.4.2.2. Test Case Table

Table 5 2: Test Case Login

Test Case ID: TC_ASS_1	Test Design by: M. H. Masuk
Test Priority: High	Test Design Date: 06.12.2018
Module Name: Login	Test Execute by: Md. Kamrul Hasan
Description: This section contain the functionality of registering a new user.	Test Execute Date: 06.12.2018

Precondition: User must be registered

Dependencies: Input correct username and password

Step	Test cases	Test Data	Expected result	P/f	Actual result
01	Enter empty value		Display error	pass	Display error

	for user name & password		message		message
02	Enter correct value for user name & password	Username and password	Accept	Pass	Display error message
03	Enter wrong value for user name & password	Username and password	Display error message	Pass	Display error message
04	Click the login button after giving correct value.		View home page	Pass	View Home page

5.4.3. Test Case Module-1:Committee creation

5.4.3.1. State Transition Diagram

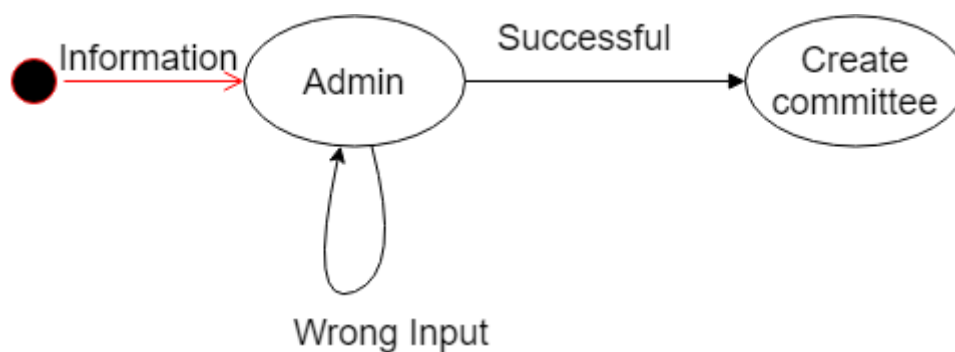


Figure 5.4. 3 : State Transition Diagram for Committee Creation

5.4.3.2. Test Case Table

Table 5 3: Test Case Committee creation

Test Case ID: TC_ASS_1	Test Design by: Tohinul Islam
Test Priority: High	Test Design Date: 07.12.2018

Module Name: Registration	Test Execute by: M.H. Masuk
Description: This section contain the functionality of registering a new user.	Test Execute Date: 07.12.2018

Precondition: User must be Admin or stuff user

Dependencies: Enter input required information

Step	Test cases	Test Data	Expected result	P/f	Actual result
01	Select leader twice a user	Kamrul, Kamrul	Display error message	pass	Display error message
02	Select member who is already in leader	Kamrul	Display error message	Pass	Display error message
03	Select member who is not in leader	Hasan	View committee list	Pass	View committee list
04	Click confirm button	Accept	View committee list	pass	View committee list

5.4.4. Test Case Module-4: modify member

5.4.4.1. State Transition Diagram

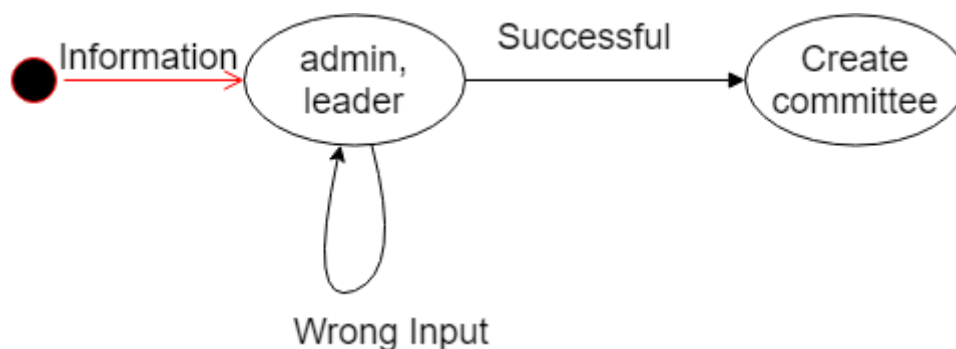


Figure 5.4. 4 : State Transition Diagram for Modify Member

5.4.4.2. Test Case Table

Table 5 4: Test Case modify member

Test Case ID: TC_ASS_1	Test Design by: Tohinul Islam
Test Priority: High	Test Design Date: 07.12.2018

Module Name: Modify member	Test Execute by: M.H. Masuk
Description: This section contain the functionality of authentic admin and committee leader.	Test Execute Date: 07.12.2018

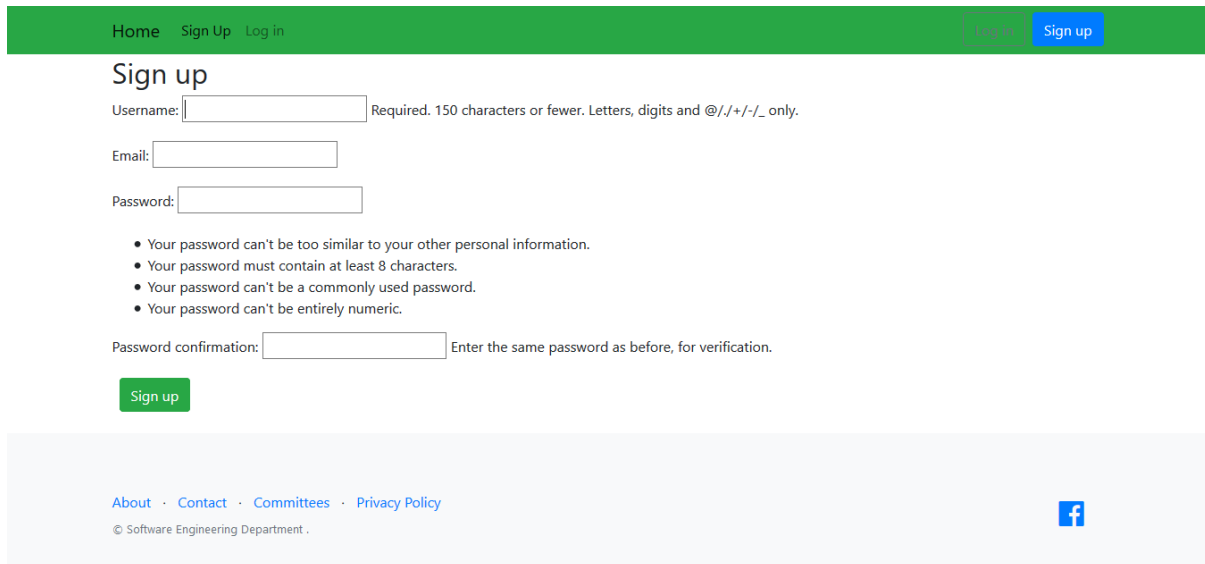
Precondition: User must be Admin or stuff user or committee leader.

Dependencies: Enter input required information

Step	Test cases	Test Data	Expected result	P/f	Actual result
01	Select member twice a user	Kamrul, Kamrul	Display error message	pass	Display error message
02	Select member who is already in leader	Kamrul	Display error message	Pass	Display error message
03	Select member who is not in leader	Hasan	View committee list	Pass	View committee detail
04	Click confirm button	Accept	View committee list	pass	View committee detail

CHAPTER-6: USER MANUAL

6.1. Registration Page.



The screenshot shows the registration page of the Software Engineering Department. It features a green header with navigation links: Home, Sign Up, and Log in. On the right side of the header, there are buttons for Log in and Sign up. The main content area is titled "Sign up" and contains a form with the following fields: Username (with a note: "Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only."), Email, and Password. Below the Password field, there are four bullet points providing password requirements: "Your password can't be too similar to your other personal information.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", and "Your password can't be entirely numeric." Below these requirements is a Password confirmation field with the instruction "Enter the same password as before, for verification." A green Sign up button is located below the form. At the bottom of the page, there is a footer with links: About, Contact, Committees, and Privacy Policy, followed by the copyright notice "© Software Engineering Department." and a Facebook icon.

Home Sign Up Log in Log in Sign up

Sign up

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Sign up

About · Contact · Committees · Privacy Policy

© Software Engineering Department .

f

Figure 6. 1Registration Page

6.2. Home Page.



Figure 6. 2: Home Page

6.3. Profile

[Home](#) [Log out](#) kamrulhasan ▾



● Name: Kamrul Hasan

Username

kamrulhasan

Email

hasan@diu.edu.bd

Teacher ID

12346789

Designation

Student

[About](#) · [Contact](#) · [Committees](#) · [Privacy Policy](#)

© Software Engineering Department .



Figure 6. 3: Profile

6.4. Committee List.

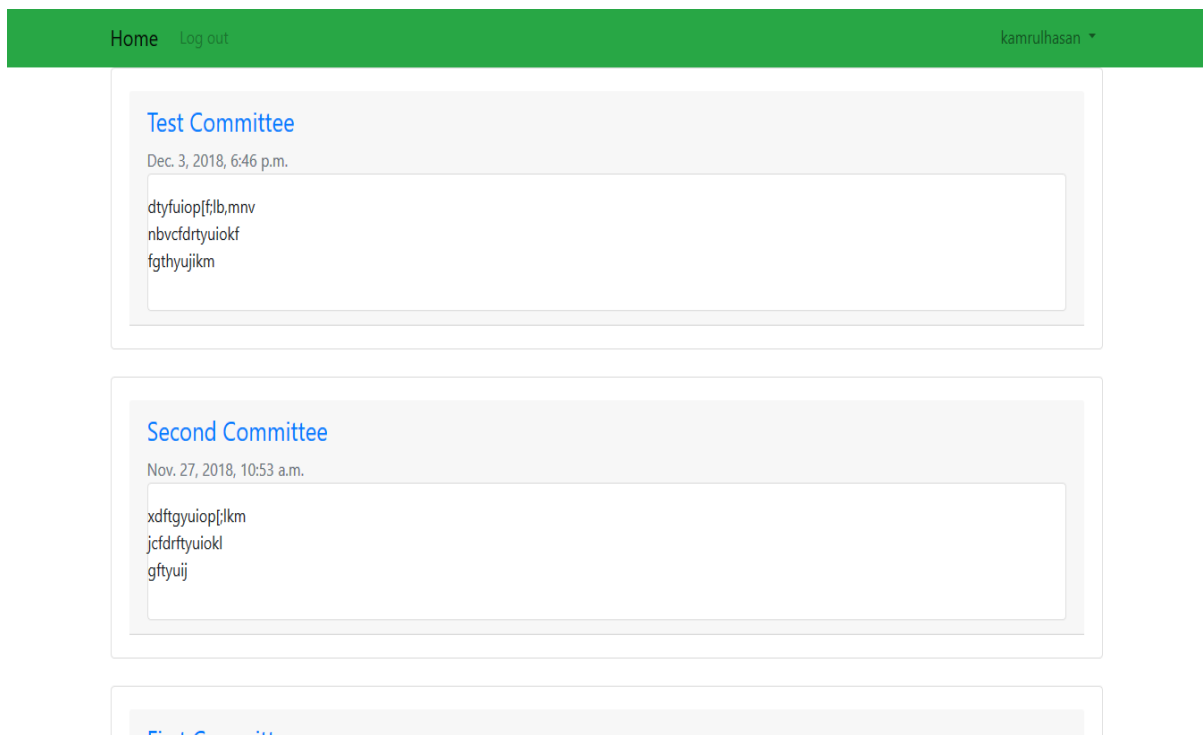


Figure 6. 4: Committee list

6.5. Event detail view

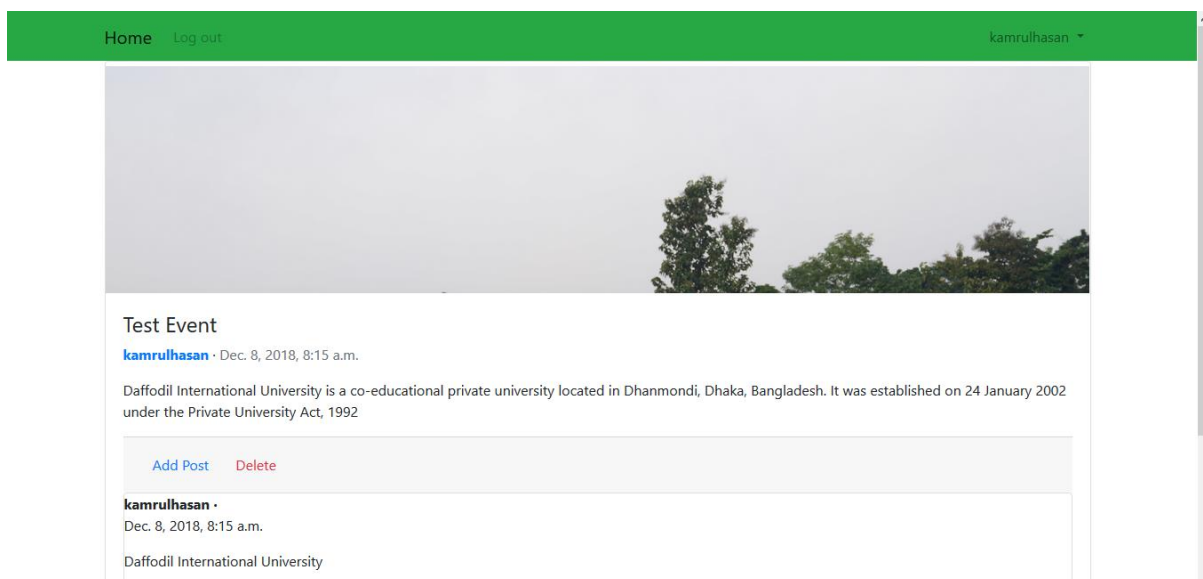


Figure 6. 5: Event detail

CHAPTER-7:PROJECT SUMMARY

7.1.GitHub Link

Link: [1]https://github.com/kamrulhasanp/committee_management

7.2. Critical evolution

Committee management system we had started Sep first week. When we are start in our project then some problem are facing badly. then we are gathering knowledge and applying in our project. In our project main problem is committee create and set member and leader. but one major problem was member is multiple time show but it's one time I have needed. Then we are many days works this problem and I find out the problem solution.

7.3. Limitations

We tried our best to implement this project, but this project has some limitations.

1. Wecannot implement leader verification to modify member.
2. Wecannot implement email validation with social login.
3. We can auto define committee name When select member and leader in a certain committee.
4. Wecannot implement to view the all attribute in a view form when want to create a committee.

7.4. Obstacles& Achievements

In our project main problem is committee create and set member and leader. but one major problem was member is multiple time show but it's one time I have needed. We are creating Database but we do not clear knowledge in database relation. We are many time waste this part. Then we are many days read more than 10 days. we are find out our every problem in our project. Then we are selected problem are solved by online sources. Our achievements are every problem are clearly solved and gather more knowledge.

7.5.Future Scope

This project created for software engineering department. Although we created only for software engineering department we can add more feature and more scope. We will think about all department and varsity including every club and group related every part management in future.

References

- [A] <https://whatis.techtarget.com/definition/use-case-diagram>
- [B] https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
- [C] <https://creately.com/blog/diagrams/sequence-diagram-tutorial/>
- [D] https://en.wikipedia.org/wiki/Class_diagram
- [E] <https://docs.python.org/3/faq/installed.html>
- [F] <https://pypi.org/project/Django/>
- [G] <https://en.wikipedia.org/wiki/HTML5>
- [H] https://www.w3schools.com/css/css_intro.asp
- [I] <https://en.wikipedia.org/wiki/JavaScript>
- [J] https://www.tutorialspoint.com/ajax/what_is_ajax.htm
- [K] <https://getbootstrap.com/>
- [L] https://en.wikipedia.org/wiki/Test_strategy