Design Document

for

mIIT-Kute

Version 1.0

Prepared by :

Group #: 8   **Group Name: Developing Decuple**

Name	Roll Number	Email id
Vedasree Tatimanu	201049	vedasreetatimanu@gmail.com
Gutta Raghavendra Chowdary	200396	grc1933@gmail.com
Akshat Garg	200084	akkigarg989@gmail.com
Janhvi Rochwani	200467	rochwani.janhvi@gmail.com
Deepak Sangle	200860	deepaksangleok@gmail.com
Yeginati Vinay Teja	201161	vinayteja686@gmail.com
Bugada Ashritha	200285	bugadaashritha2002@gmail.com
Jayaprakash Napa	200622	jayaprakashnapa2003@gmail.com
Kembasaram Nitin	200505	kembasaramnitin@gmail.com
Sai Charan Modem	200586	saicharansai2570@gmail.com

Course: **CS253- Software Development and Operations**
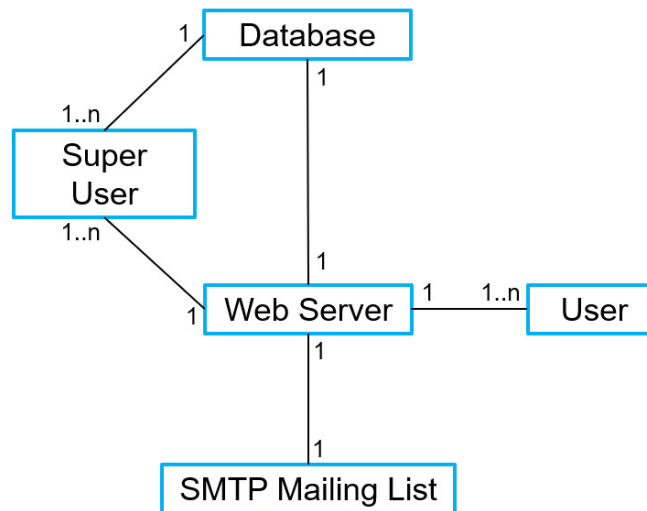
Mentor TA: **Mr. Sri Madhan**

Date: **15/02/2022**

Contents

Revisions

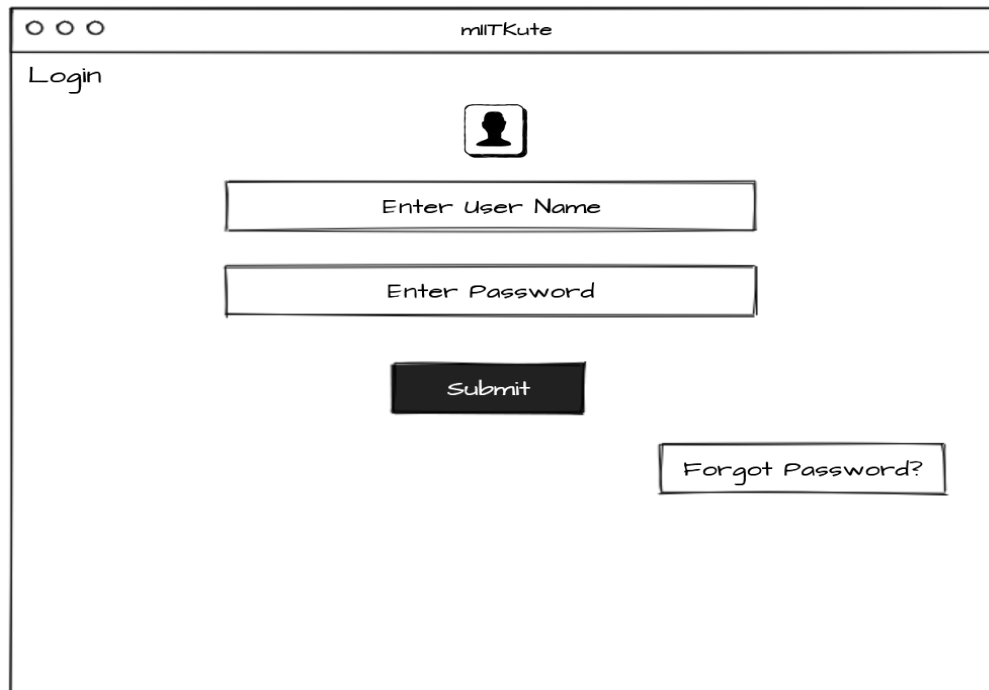| Version	Primary Author(s)	Description of Version	Date Completed
1.0	Vedasree Tatimanu Gutta Raghavendra Chowdary Akshat Garg Janhvi Rochwani Deepak Sangle Yeginati Vinay Teja Bugada Ashritha Jayaprakash Napa Kembasaram Nitin Sai Charan Modem	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	15/02/22

1 Context Design

1.1 Context Model

```
                              1 ┌──────────────┐
                               ┌┤   Database   │
                               │└──────────────┘
                               │       │ 1
                          1..n │       │
                    ┌──────────┴───┐   │
                    │    Super     │   │
                    │    User      │   │
                    └──────────┬───┘   │ 1
                          1..n │       │
                               │       │
                             1 │ ┌─────┴────────┐  1    1..n ┌──────────┐
                               └─┤  Web Server  ├───────────┤   User   │
                               1 └──────┬───────┘           └──────────┘
                                        │ 1
                                        │
                                        │ 1
                              ┌─────────┴──────────┐
                              │ SMTP Mailing List  │
                              └────────────────────┘
```

        The Context Model consists of a database system, user and super user, webserver and SMTP Mailing List. This components interacts with each other.

        The Database system mainly stores the information regarding the user Credentials, the different events he/she subscribed to as well as the information regarding his/her future events. The Super user have different privileges like deleting and creating interests, events, feedbacks, etc. The Web Server is used to easily control over the whole system and maintain flexibility with respect to different components of the application.

1.2 Human Interface Design

If OTP is valid:

If OTP is invalid:

Here primarily there are two types of users:

1. If a new user, then the user has to register in the portal with their name, IITK roll number, and mail ID for the mIIT-Kute portal. Then the user gets a mail with an OTP to the specified ID for verification.
2. If the user is already registered,
    1. Enter the credentials (username and password), then the login page will direct them to the home page.
    2. If the user forgets the credentials, they have to click "forgot password". The user can then retrieve their details by using his mail ID. OTP will be sent to their mail for confirmation.

Finally the user will be directed to the homepage.
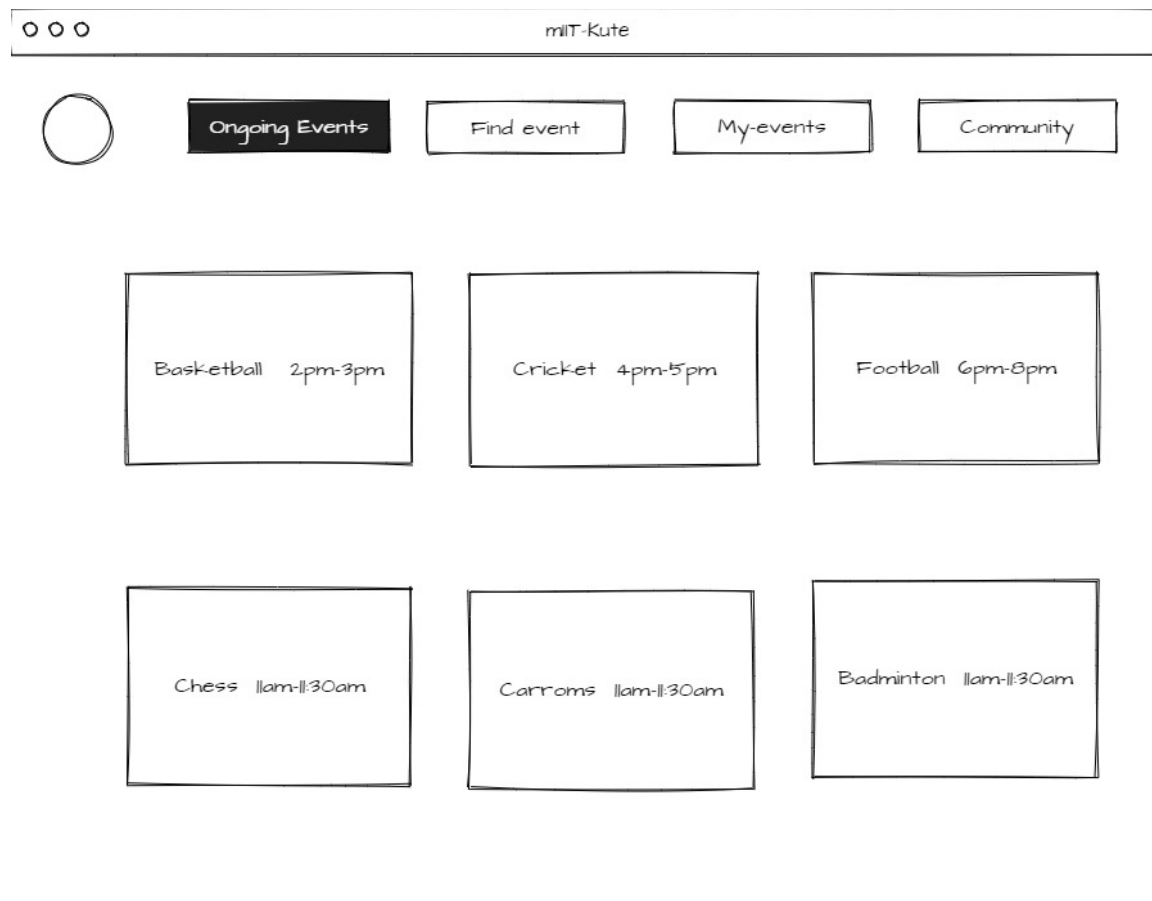
Here, the display shows the ongoing events that are purely based on the interests that were filled by the user while registering for the portal. By clicking on one of the ongoing events that is desirable for the user, they will be directed to the display of the event template.
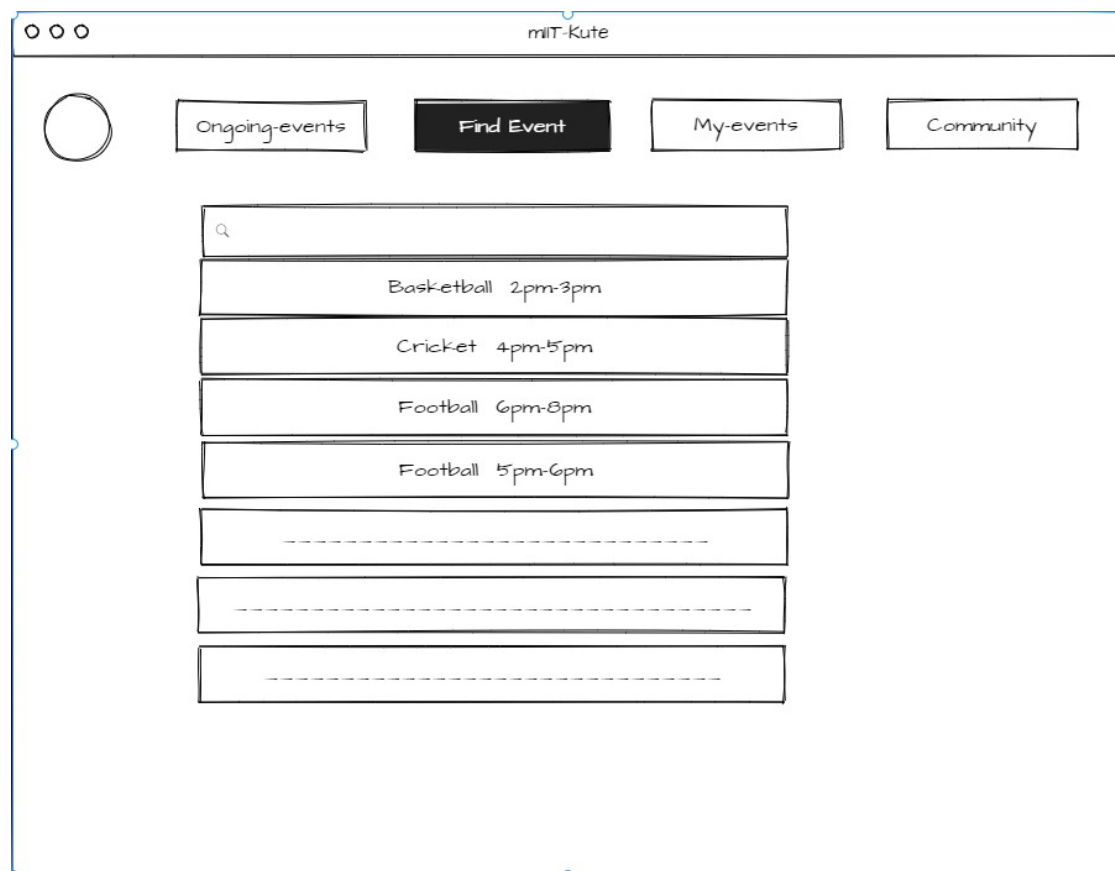
The module "Find Event" is for searching an event, and if there exists an event with desired interest and time for the user, they can join the event. If there doesn't exist any desired event then the user has to create an event.
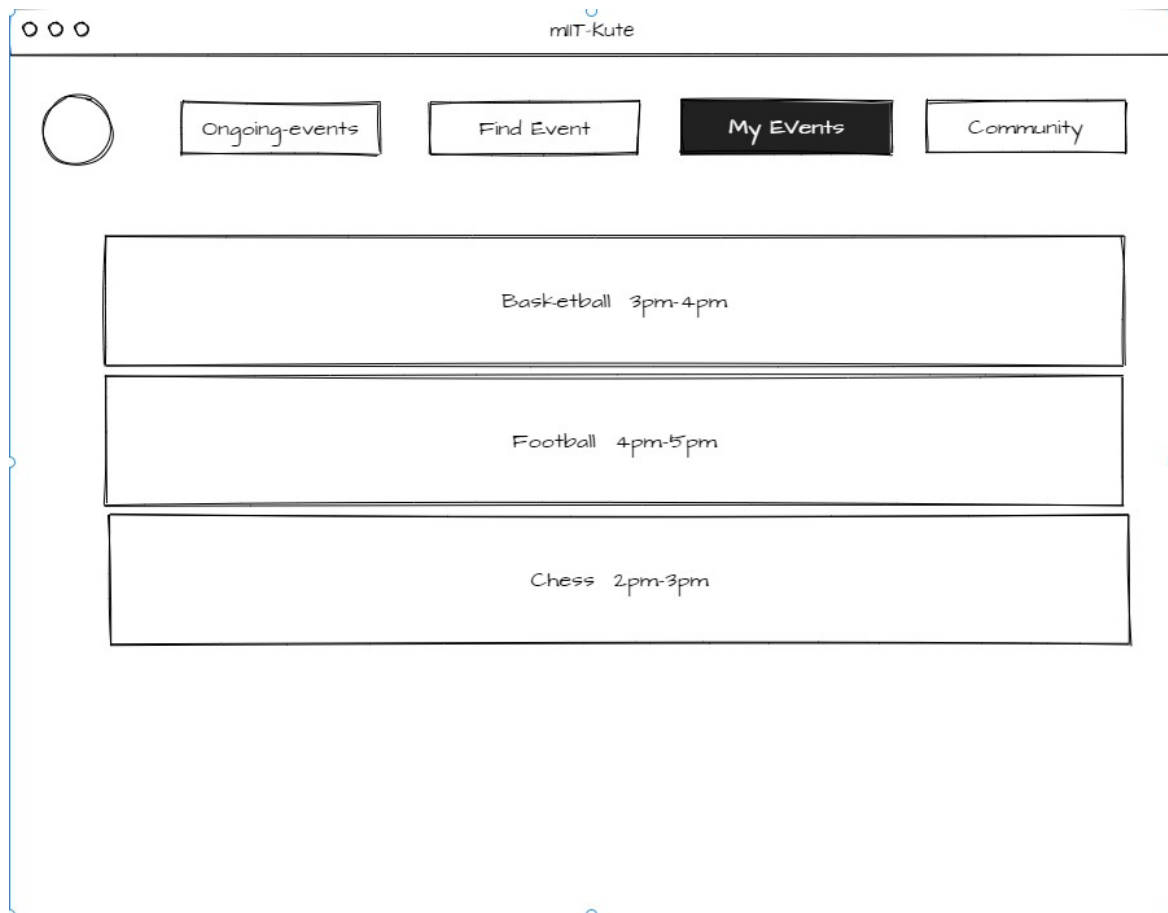
Here the screen shows the events that the user has created/participated in.
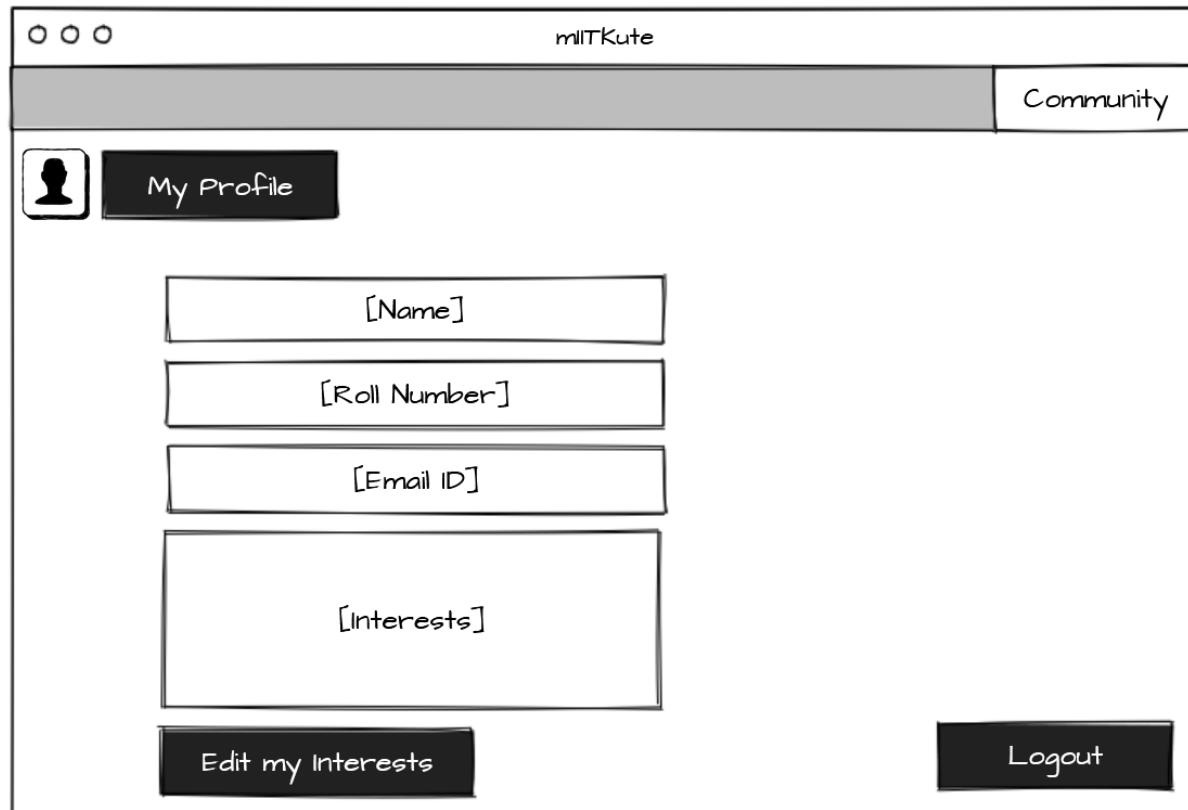
Here the screen shows the template of one of the events that the user has been in and the chat box corresponding to that event.
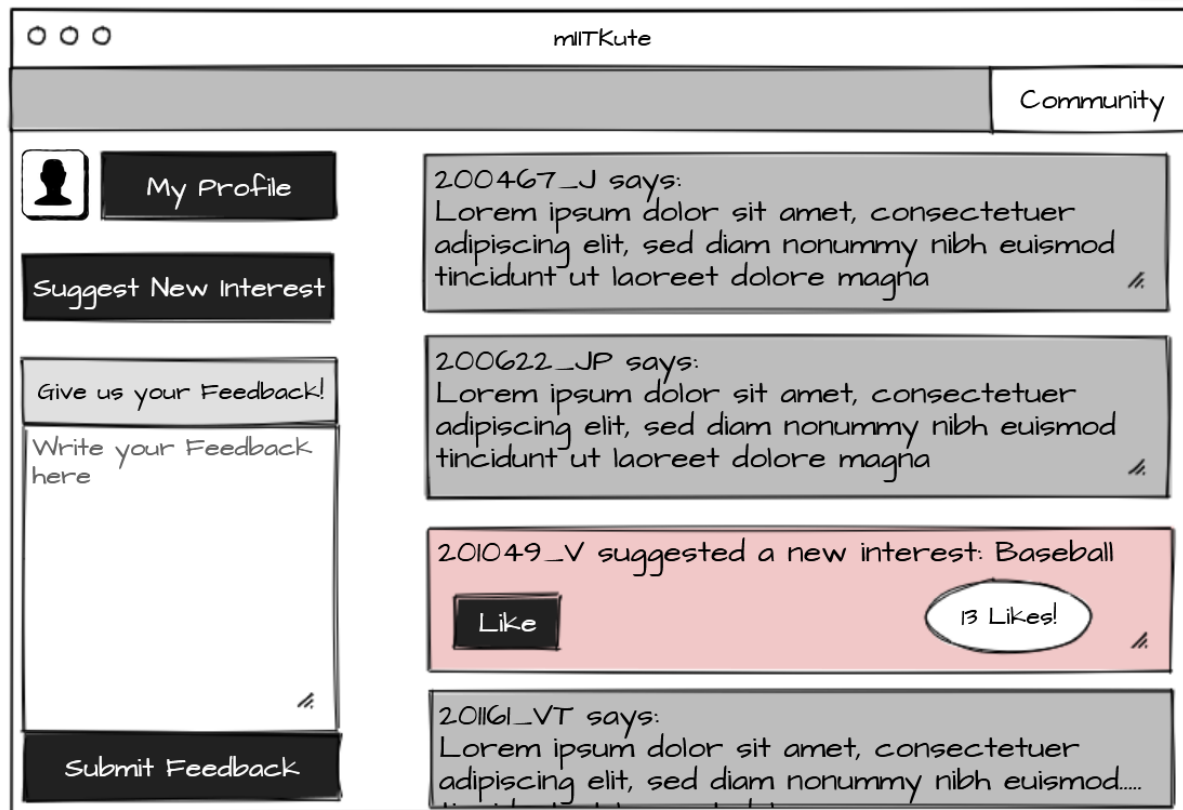
The module "Community" contains classes such as profile, feedback ,suggestions about interests etc,. The user can access their profile, suggest a new interest and give feedback.

On clicking "My Profile":

```
 ○ ○ ○                          mIITKute

                                                    Community

  [👤]    My Profile


              ┌─────────────────────────────────┐
              │            [Name]               │
              └─────────────────────────────────┘
              ┌─────────────────────────────────┐
              │         [Roll Number]           │
              └─────────────────────────────────┘
              ┌─────────────────────────────────┐
              │          [Email ID]             │
              └─────────────────────────────────┘
              ┌─────────────────────────────────┐
              │                                 │
              │                                 │
              │          [Interests]            │
              │                                 │
              │                                 │
              └─────────────────────────────────┘
    ┌─────────────────────┐         ┌─────────────────────┐
    │  Edit my Interests  │         │       Logout        │
    └─────────────────────┘         └─────────────────────┘
```

On clicking "Suggest New Interest":

```
 ○ ○ ○                        mIITKute

                                                    Community



                    ┌─────────────────────┐
                    │                     │
                    │                     │
                    │  Type Interest Here │
                    │                     │
                    │                     │
                    └─────────────────────┘

                       ▐ Suggest New Interest ▌
```

If Interest already exists, i.e, it is predefined by the system:

```
 ○ ○ ○                        mIITKute

                                                    Community



                    ┌─────────────────────┐
                    │                     │
                    │ This Interest already exists! │
                    │                     │
                    └─────────────────────┘

                       ▐ Suggest New Interest ▌
```

If the suggestion already exists, i.e, it has already been suggested by some other user:
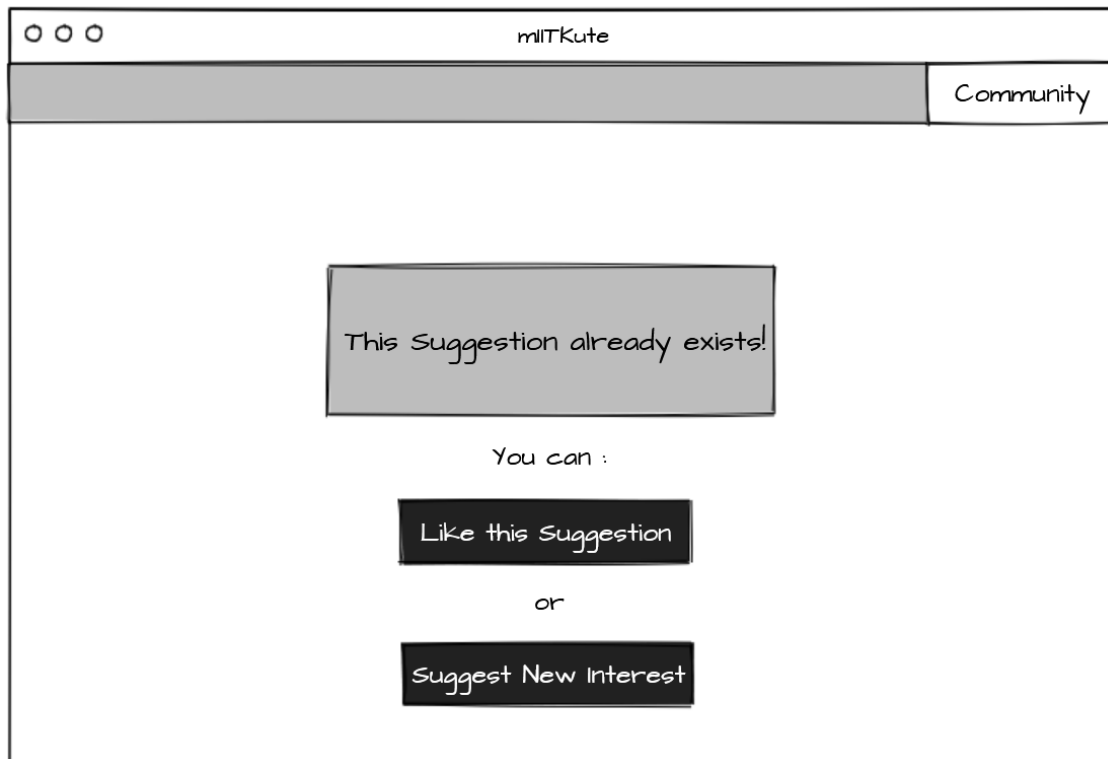
If suggestion (is new) and accepted:

mIITKute

Community

Suggestion successfully added!
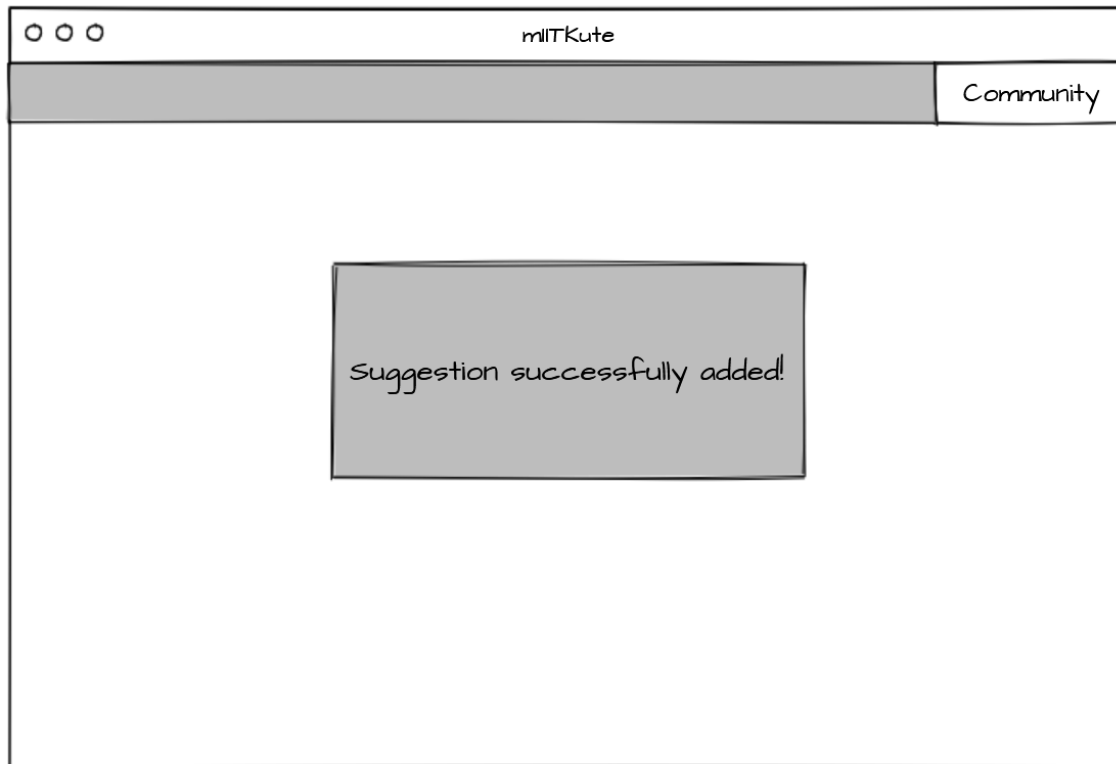
2 Architecture Design

Layered Architecture

Web Browser Interface

User Login	Ongoing Events	MyEvents	Feedback

Events Management	Interest Editing	Chat Synchronization	Feedback Monitoring

User Details	Chat History	Feedbacks

Database
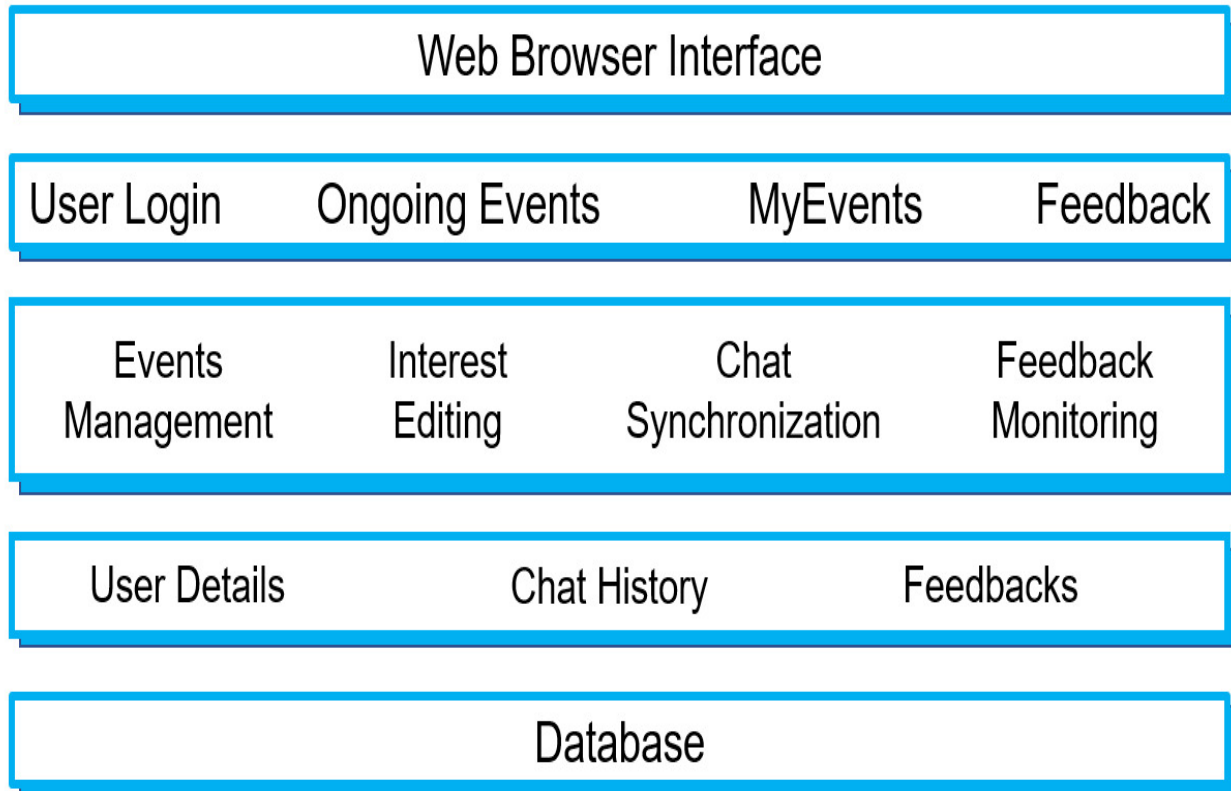
3 Object Oriented Design

3.1 Use Case Diagrams
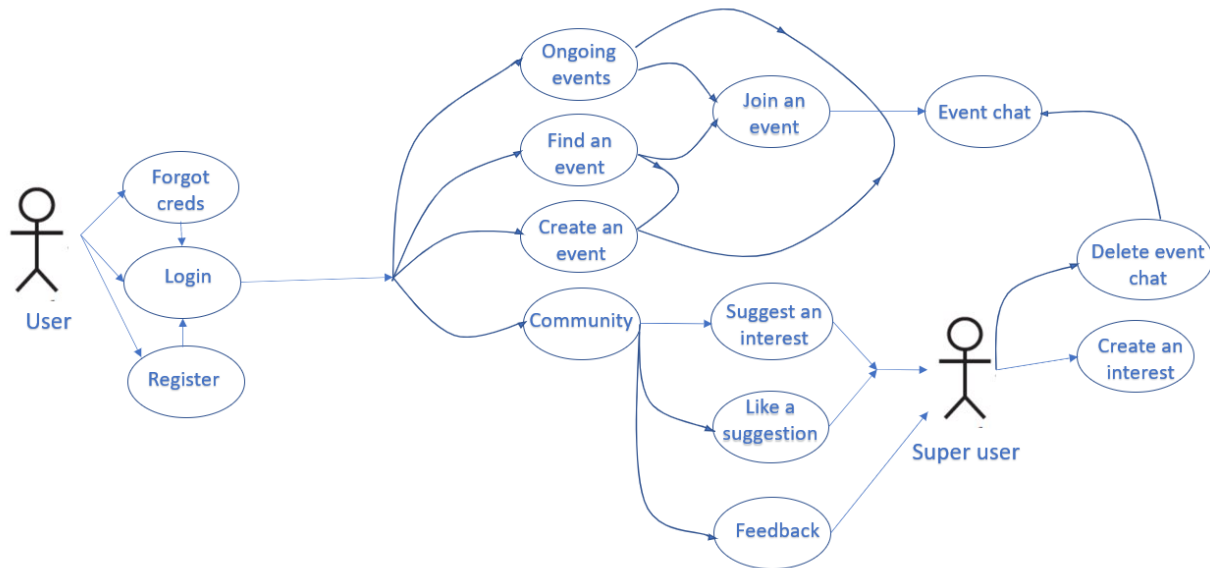
We have 3 different use case diagrams in our application.

1. The first use case diagram explains the structure of how the user will login to the application or register if it is the first time visiting the application. It also deals if one forgets his/her password

2. The second use case diagram mainly deals with the joining, creating different events and interacting with it along with the chatbox system to interact among different users.

3. The third use case diagram focuses on providing suggestion, feedbacks to the application regarding different aspects of the application as well as regarding creating new interest in the Server.

3.1.1 Use Case #1
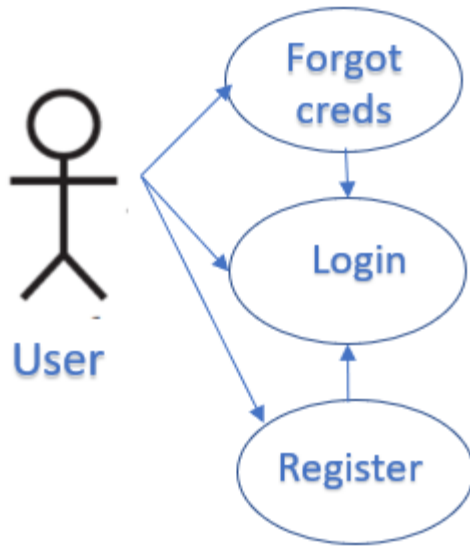
Use Case S.no	Use Case #1
Purpose	User registration or login
Requirements Traceability	IIT-K mail id for new users and login info for registered users.
Priority	High
Preconditions	None
Post conditions	User logs in
Actors	User
Exceptions	None
Includes	None
Notes/Issues	Assumed that the user belongs to IIT-K.

3.1.2 Use Case #2
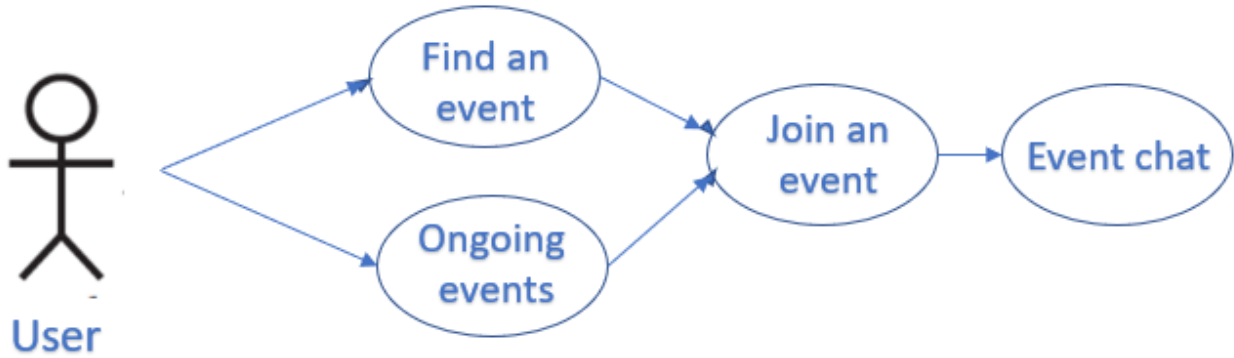
Use Case S.no	Use Case #2
Purpose	Join an event
Requirements Traceability	User id, user interests,event schedule
Priority	high
Preconditions	User is logged in
Post conditions	User joins an event
Actors	User
Exceptions	Required event is not available
Includes	Use case #1
Notes/Issues	All the events should be visible.

3.1.3 Use Case #3
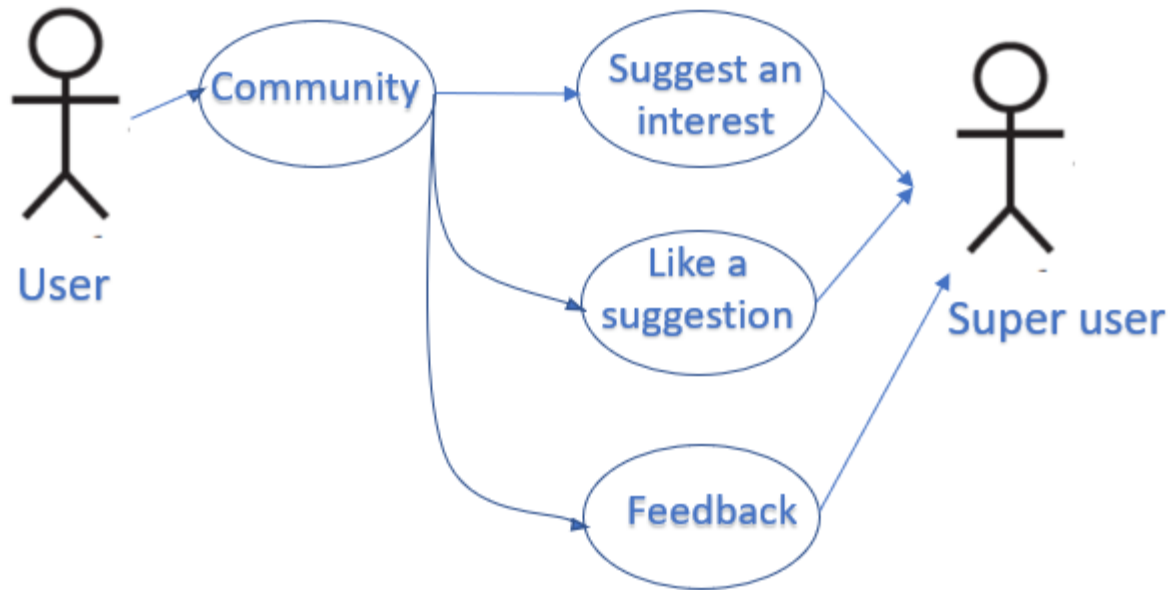
Use Case S.no	Use Case #3
Purpose	To take feedback and create new interest
Requirements Traceability	User ID, Existing requests
Priority	Medium
Preconditions	User must be logged in
Post conditions	New interest will be created if there are enough likes,a new interest request is made.
Actors	User,Super user
Exceptions	Existing request
Includes	Use Case #1
Notes/Issues	None

3.2 Class Diagrams

Class Diagram for the interaction between the user and different classes of the webapp is given below.
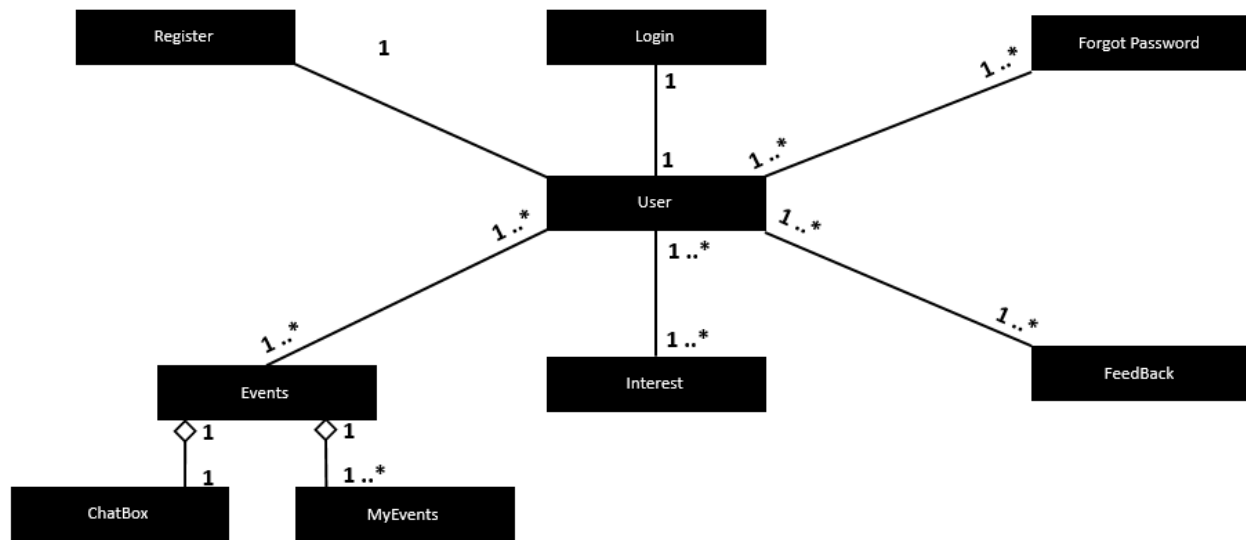
Details of the main classes used in the system:

User

Name:	String
Email ID:	String
Roll no.:	String
Password:	String

Login()
Register()
Forgot_Password()

Register

Name:	String
Email Id:	String
Roll no.:	String
Username:	String
Password:	String

SendOTP()
VerifyOTP()
NewPassword()

Login

Username:	String
Password:	String

Check()
Forgot_Password()

Forgot Password

Email Id:	String

SendOTP()
VerifyOTP()
NewPassword()

Events

Name:	String
Interest:	String

SearchEvent()
CreateEvent()
JoinEvent()

Interest

Name:	String

append_interest()

FeedBack

Name:	String
Likes:	Integer
Feedback:	String

update_feedbacks()
display_feedbacks()
get_old_feedbacks()

ChatBox

Event Name:	String
Messages:	String

get_previous_chat()
display_chat()
update_chat()
enter_text()

MyEvents

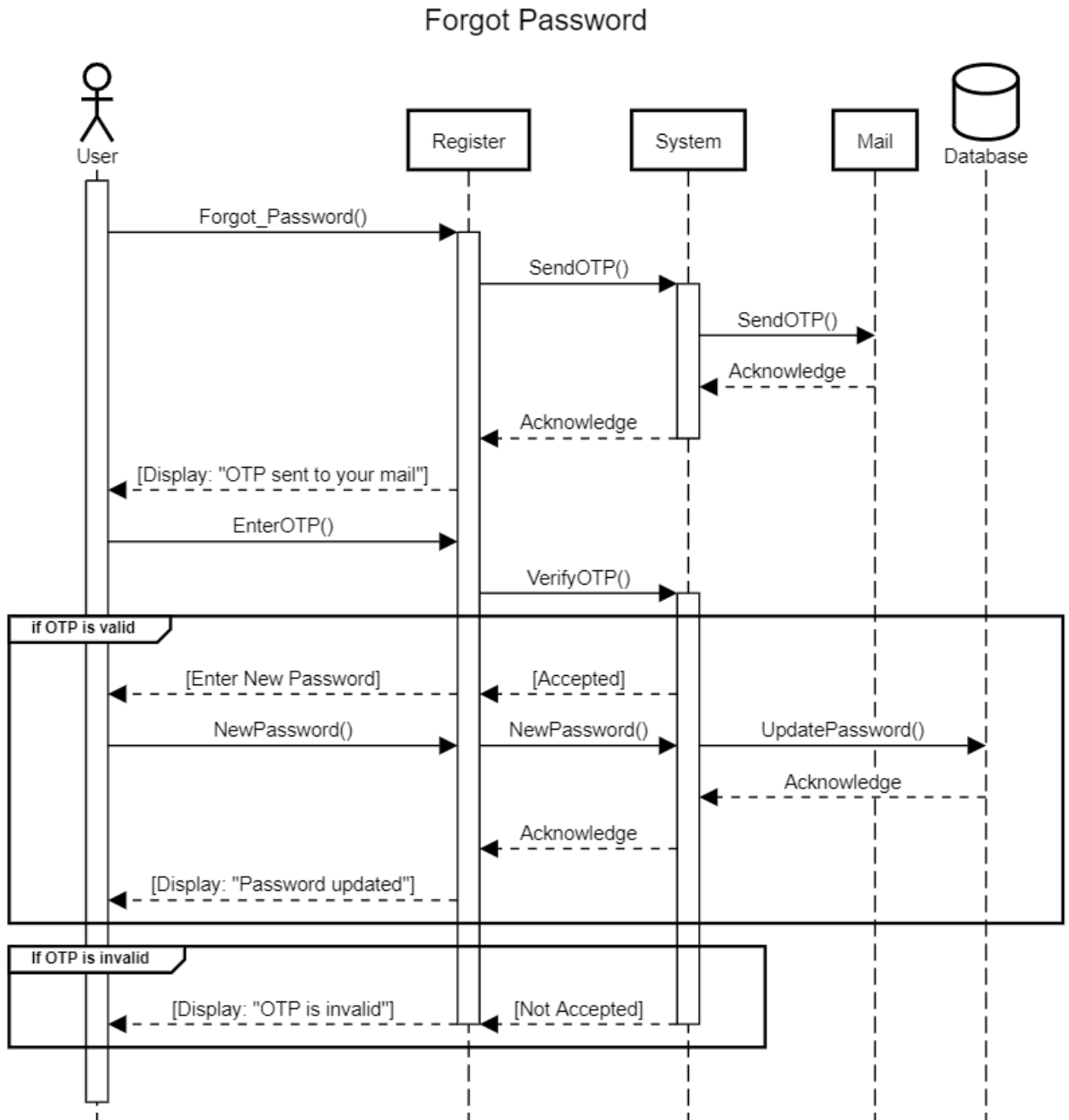|	
Name:	String
Date/Time:	String
Location:	String

Pop_out_ChatBox()

3.3 Sequence Diagrams

1. Sequence Diagram for Register Functionality on the Login Page is given below.

Register
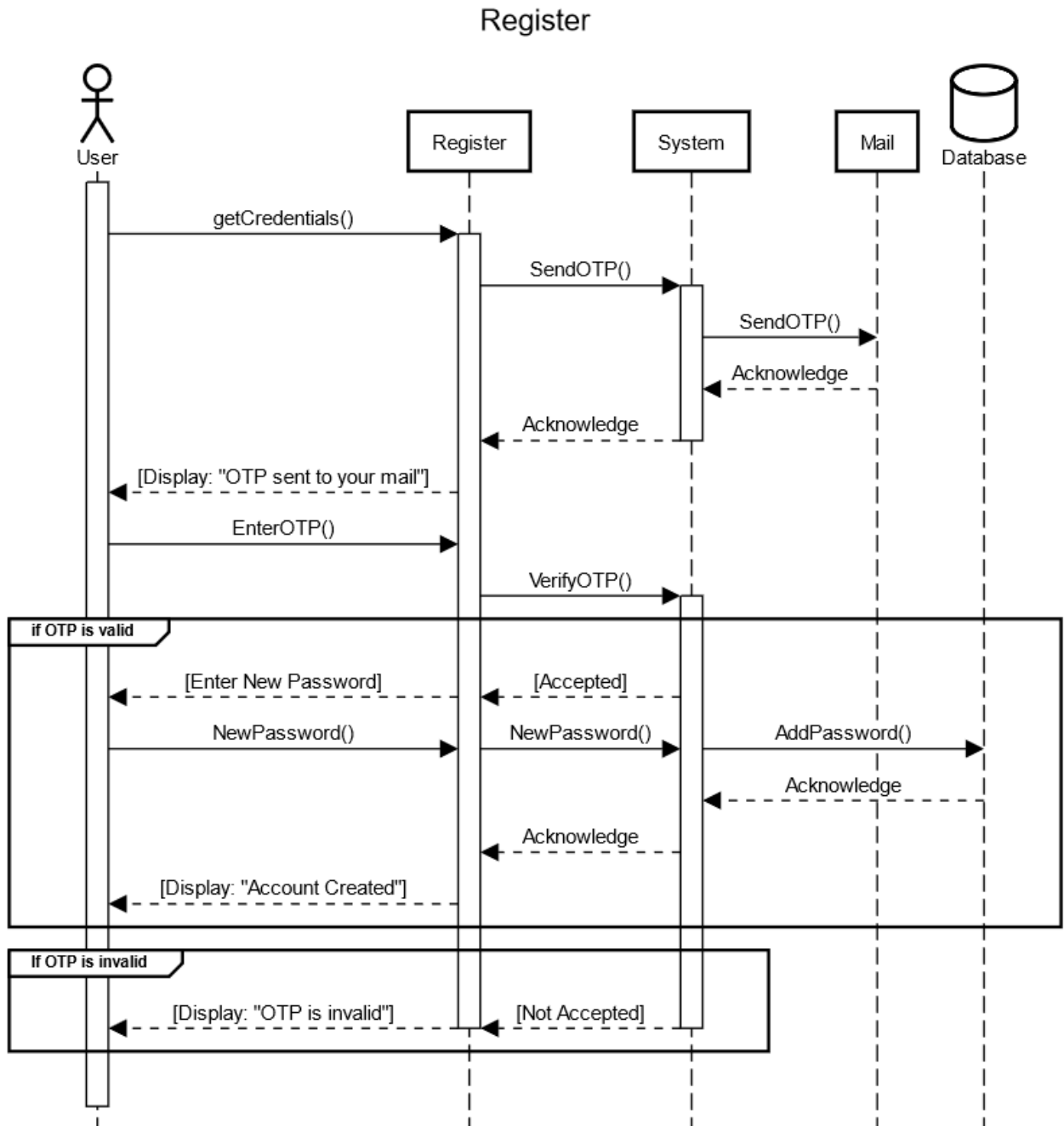
2. Sequence Diagram when the user Forgot the Password on the Login Page is given below.

Forgot Password

3. Sequence Diagram for Searching any event on Homepage is given below.

Search Event
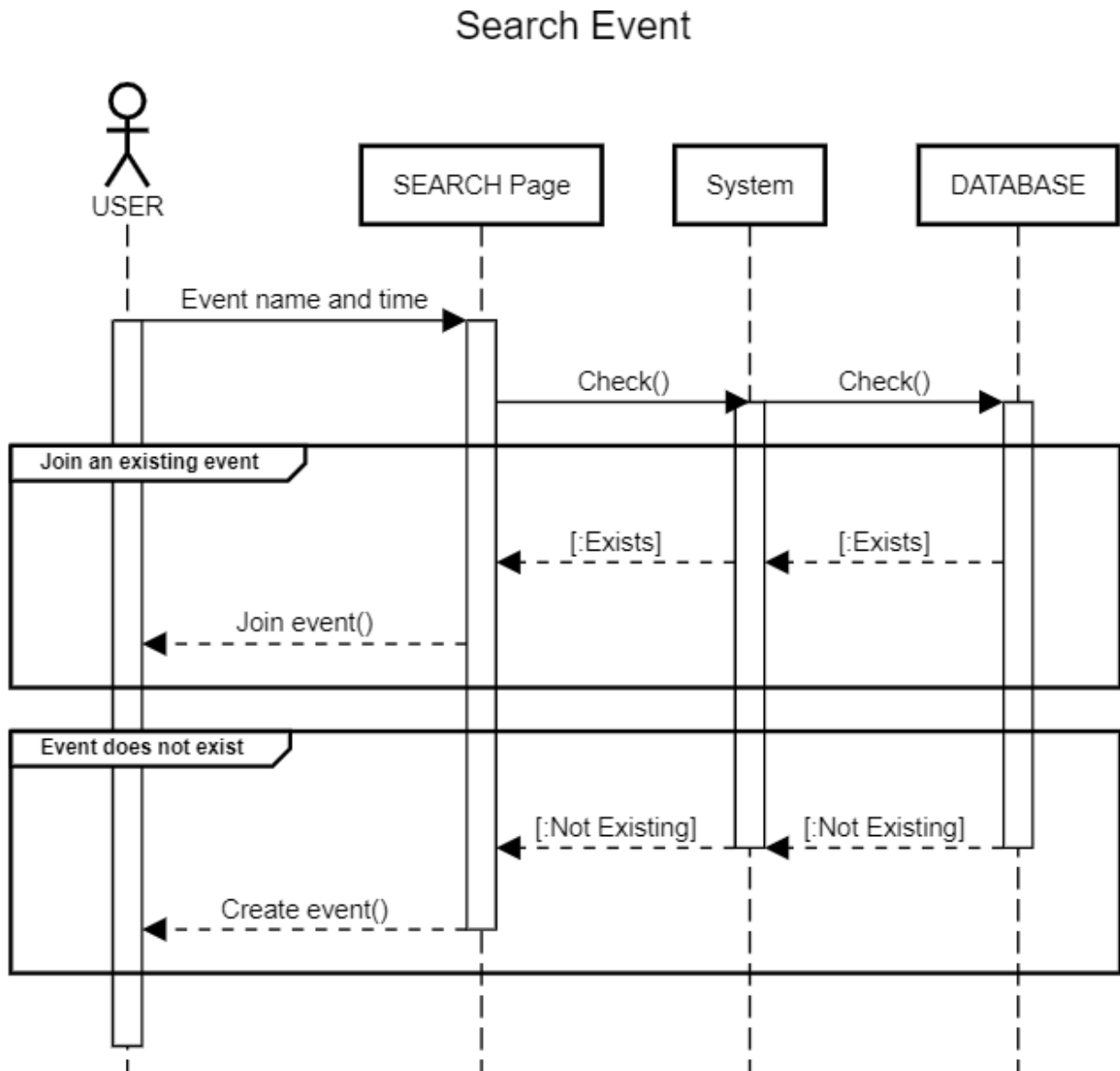
4. Sequence Diagram for providing Feedback and suggestion by the users is given below.

Feedback Page
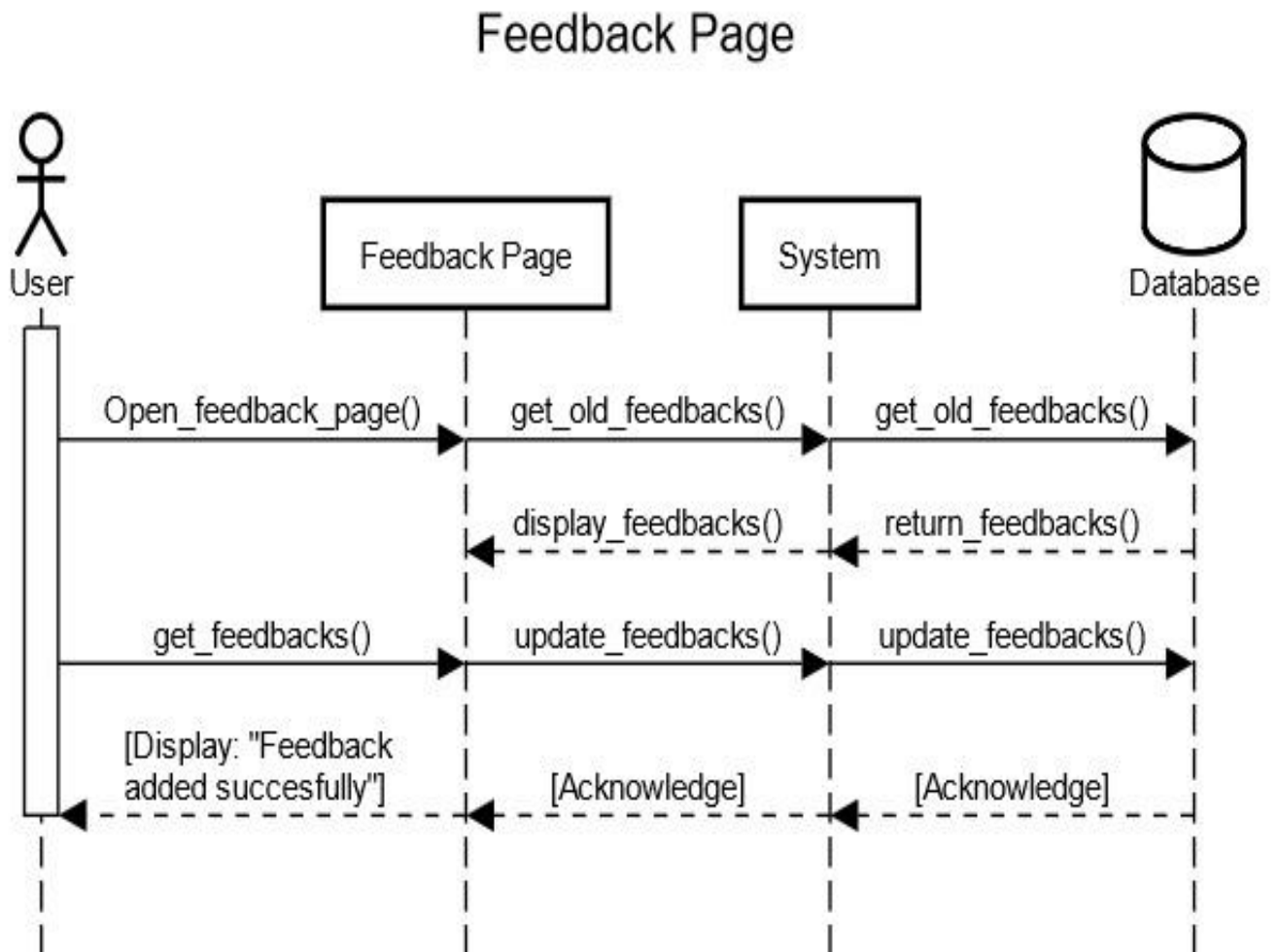
5. Sequence Diagram for Chat Box Functionality on the Event Page is given below.
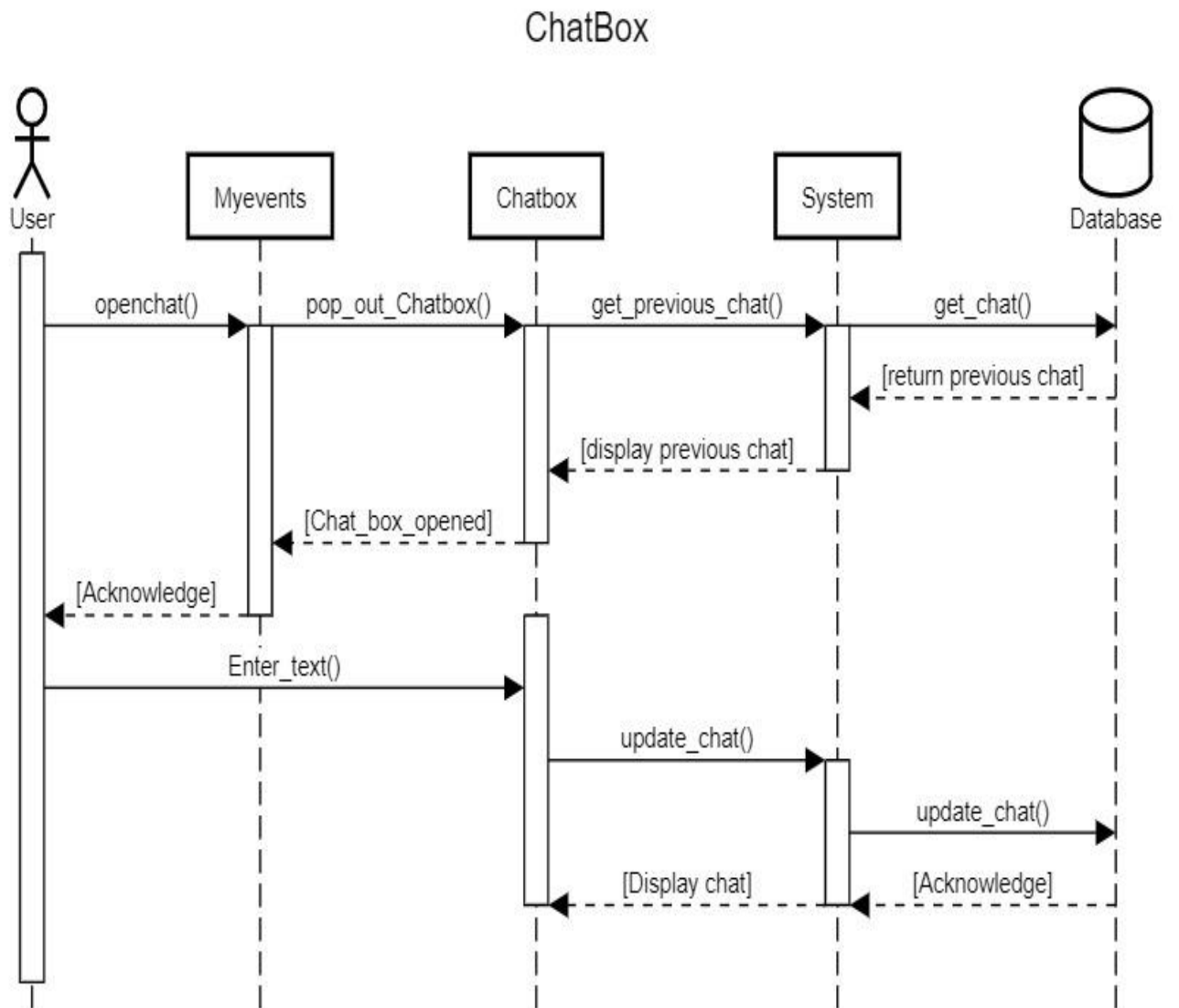
6. Sequence Diagram for Creating new Interest by the super user is given below.

Super User Creating New Interest

7. Sequence diagram for creating event is given below.

Create Event

3.4 State Diagrams

State diagram for user
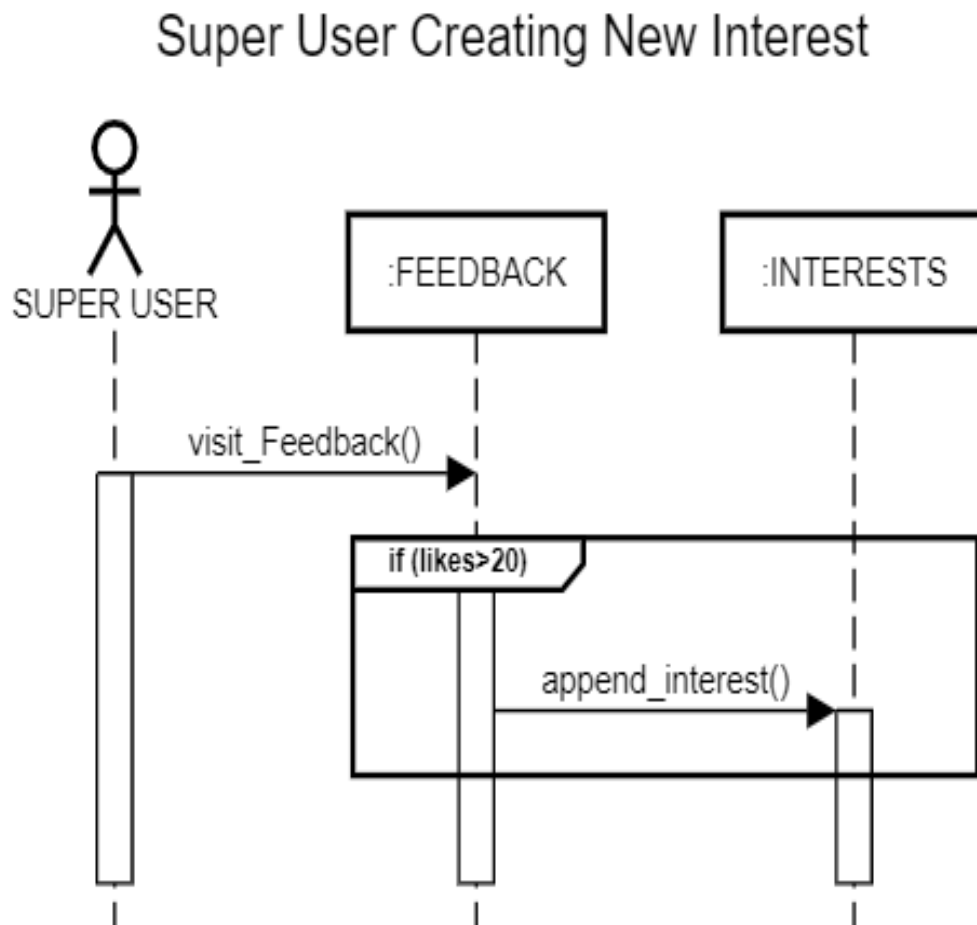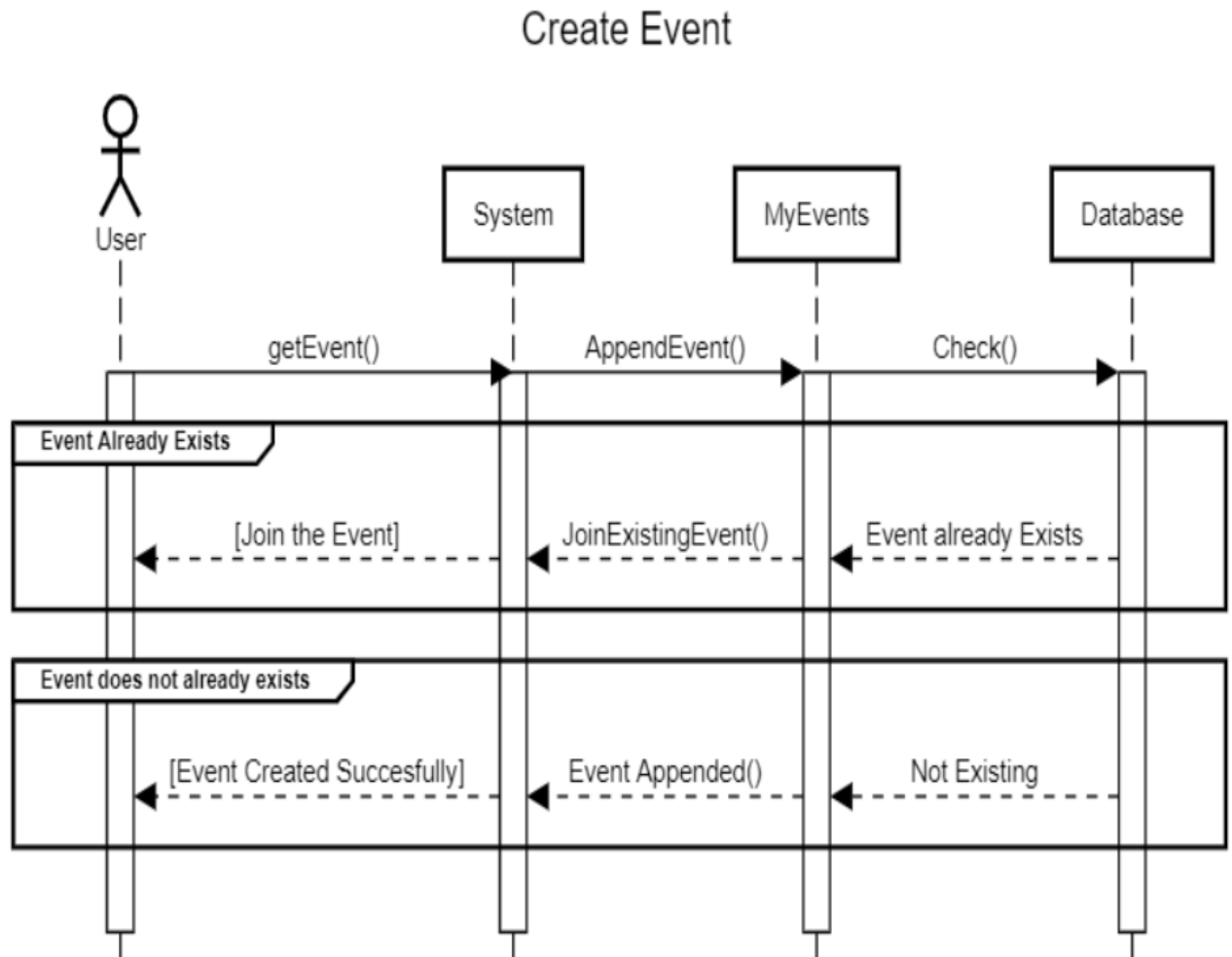

Log out

Enters login details

| Search |
| Do : Search for an event |

| Join |
| Do : Join a chat server |

Register

| Logged out |
| Do : Display login page |

| Logged in |
| Do : Display homepage with ongoing events |

Search

Event not found

Join

Join an ongoing event

Chat

| Create |
| Do : Create an event |

Create

Log out

Log out

Chat

| Chat |
| Do : Display chat box |

Feedback

Log out

| Feedback |
| Do : Take feedback |
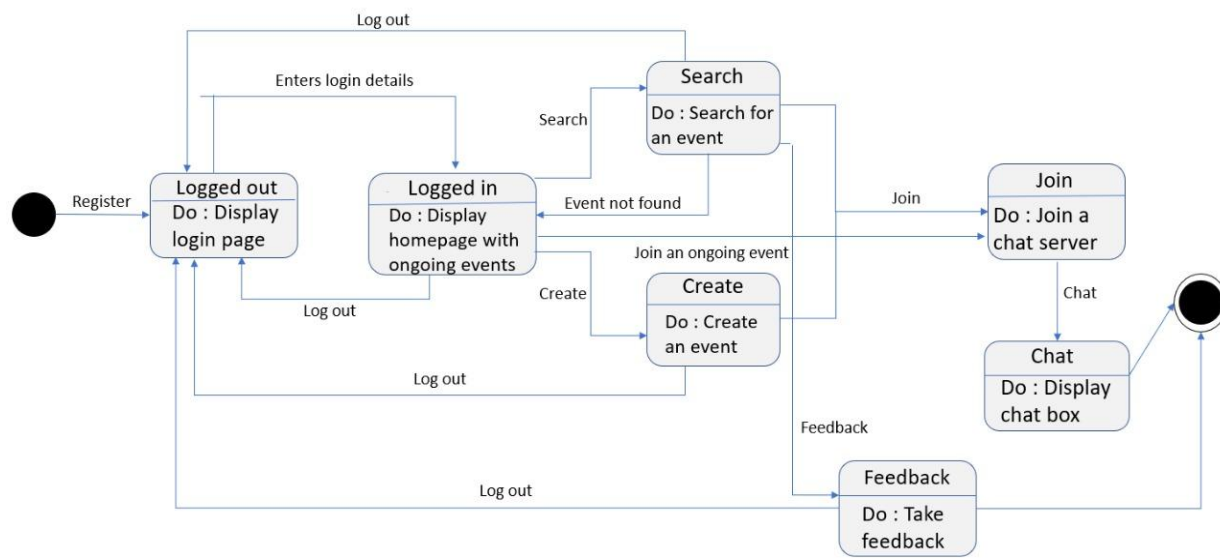
* Users can logout of their account at any stage in the above diagram

State diagram for super user

Create

| Create |
| Do : Create an interest based on feedback |

Register

| Logged out |
| Do : Display login page |

Enters login details

| Logged in |
| Do : Display feedback and interest suggestions |

Remove

| Remove |
| Do : Remove an event |

Log out

Delete

| Delete |
| Do : Delete a feedback |
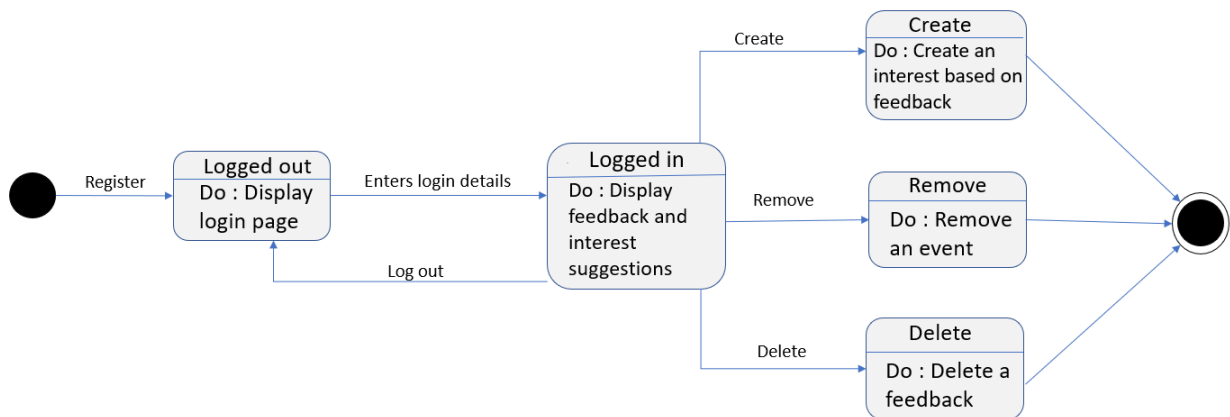
4 Project Plan

<Provide a detailed plan for the implementation and testing of the software. Identify the major tasks and their dependencies. Capture the plan in a Gantt Chart. Also, for each task, identify the team member who will be responsible for completing the task.>

We will be using **GitHub** to manage our codes and will be using **Visual Studio Code** as our IDE.

1. **Front End**

    a. **Wireframing** - Wireframing was used to identify different components such as navigation bar, elements, texts and images units,etc. The basic planned UI was thus first represented using lines, rectangles and blind texts. Such wireframing was achieved using https://www.mockflow.com/wireframing/ .

    b. **UI Design** - Will be designing the UI for the different webpages and their respective prompts to be displayed in our web application.

    c. **UI Development**:- Will write the actual code for UI Design implementation using HTML, CSS and JavaScript.

2. **Back End**

    a. **Database Management** - Storing and accessing user data, current events data, and user feedback.
    Language - SQL, RDMS- Sqlite

    b. **User Management** - Managing user login and registration.

    c. **System Design** - Defining all the functions and logic for system to be able to handle user requests/inputs

Tasks:

1. **Implementation** - Will be divided uniformly between all developers.

2. **Unit Testing** - In this task, individual units of software i.e. group of computer program modules, usage procedures and operating procedures will be tested to determine whether they are suitable for use or not. Every independent module will be tested by the developer themself to determine if there are any issues.
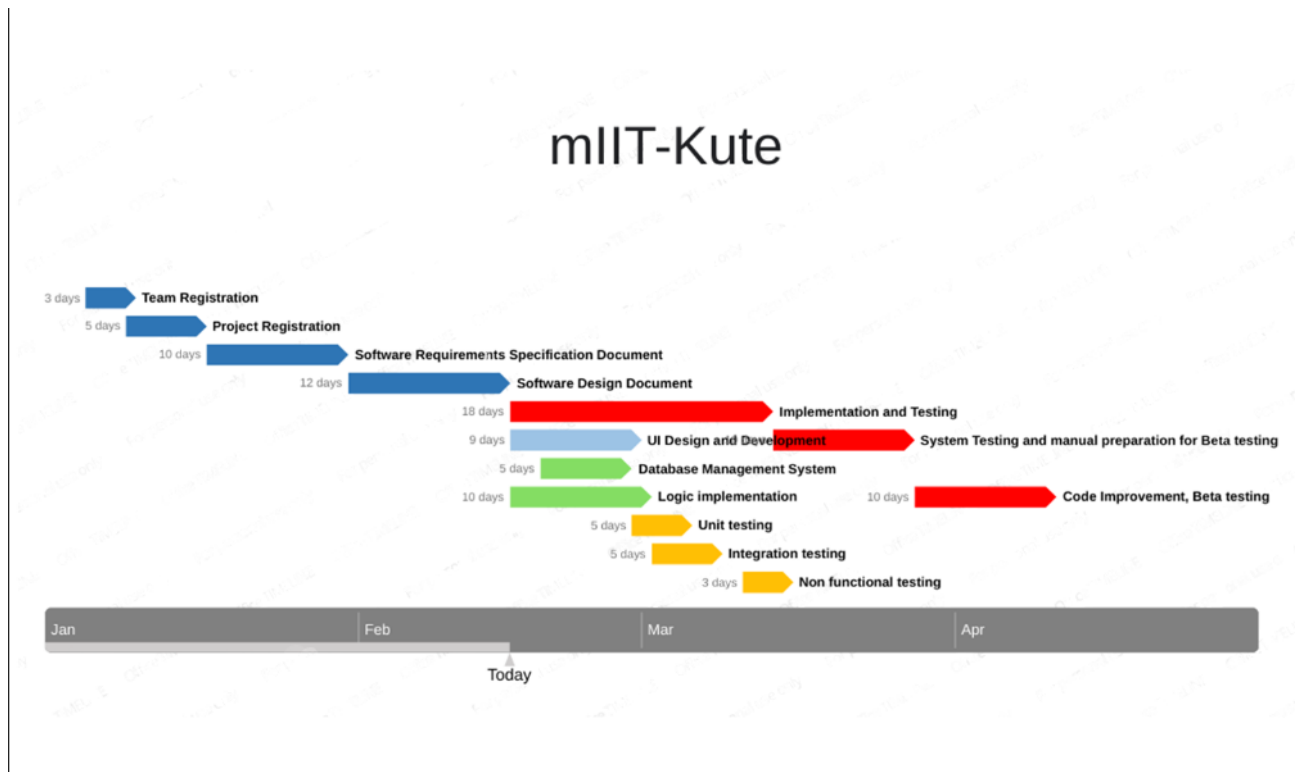
3. **Integration Testing** - In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.
   A. User registration and login (OTP mail)
   B. Search (New interests, events)
   C. Chat (Users synchronisation)
   D. Automated deletion of chat box after 30 minutes of event completion
   E. Feedback

4. **System Testing -** validates the complete and fully integrated software product
   A. Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called the End to End testing scenario.
   B. Verify thorough testing of every input in the application to check for desired outputs.
   C. Testing of the user's experience with the application.

5. **Manual Preparation for Beta Testing** - A clear set of instructions will be compiled in this manual so as to ensure easy understanding for the Testing team.

6. **Code Improvement** - Modifications made to the code after getting feedback from different clients and results of testing.

7. **Beta Testing of some other project**

S. No.	Team Member	Task
1	Ashritha Bugada	System Design (Back End)
2	Vedasree Tatimanu	Database Management
3	Jahnvi Rochwani	UI development
4	Akshat Garg	UI development
5	Deepak Sangle	UI development
6	Vinay Teja	System Design (Back End)
7	Jayaprakash Napa	System Design (Back End)
8	Raghavendra G	Database Management
9	Nitin Kembasaram	Manual Preparation for beta testing
10	Sai Charan	Manual Preparation for beta testing

*Any task not explicitly divided will be handled by the team as a whole

TIMELINE

The Gantt Chart for the project timeline is

5 Other Requirements

<This section is <u>Optional</u>. Please provide any other details that are suitable for being included in the design document .>

Appendix A - Group Log

February 5, 2022	2:30 PM - 5:00 PM	First meeting about the project. Division of work among the mates and planning the work distrubution among the individuals.
February 6, 2022	4:00-4:30	A small meet with T.A to get a brief on Software Design Document.`
February 7, 2022	4:30 PM - 7:00 PM	Completed the use cases and the UI diagrams with the necessary changes and description to each of the Human Interface Diagrams.
February 9, 2022	9:00 PM to 10:30 PM	Completion of state diagrams and the context model and allotting the remaining work to the teammates.
February 11, 2022	11:30 AM to 1:30 PM	Completion of class and sequence diagrams and making sure they refer to the same conclusion in the view of a user.
February 13, 2022	3:00 PM to 5:00 PM	Finalisation of the software design document so that we could present it to our TA.

February 15, 2022	11:00 AM to 12:30 PM	Had a meeting with our TA and found the necessary changes that need to be done in various sections in the software design document as instructed by our TA.
February 15, 2022	3:00 PM to 8:00 PM	Worked on the necessary changes that were instructed by our TA and made the final Software design document that needed to be submitted.