Design Document

for

MapGo

Version 1.1

Prepared by

Group #: 6 **Group Name: Wall-E(s)**

Abhinav Garg	**210029**	**abhinavg21@IITK.ac.in**
Chouhan Jayesh	**210296**	**chouhanj21@IITK.ac.in**
Harshini Dola	**210418**	**harshini21@IITK.ac.in**
Mondem Shanwitha Yadav	**210628**	**shanwitha21@IITK.ac.in**
Nirmal Prajapati	**210735**	**nirmal21@IITK.ac.in**
Prachi Choudhary	**210732**	**prachic21@IITK.ac.in**
Priyanshu Meena	**210785**	**priyanshum21@IITK.ac.in**
Rohan Ravi	**210870**	**rohanr21@IITK.ac.in**
Sumit Kumar Bairwa	**211074**	**sumitkb21@IITK.ac.in**
Ujjwal Gautam	**211122**	**ujjwalg21@IITK.ac.in**
Yashas D	**211199**	**yashasd21@IITK.ac.in**

Mentor TA: ***Mr. Anirudh Nanduri***

Contents II

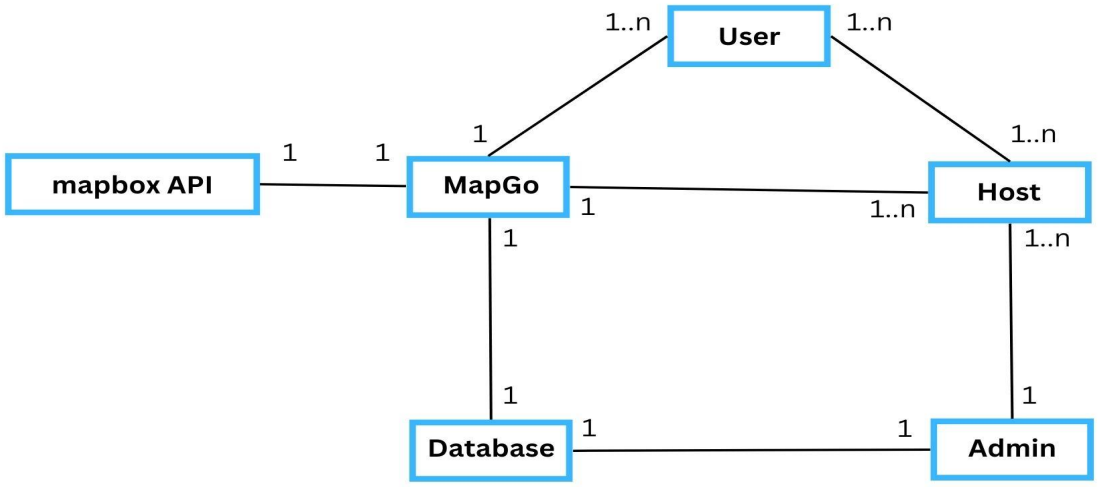
Revisions II

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Group#6 WallE(s)	This document holds details about the object oriented design.	10/02/23
1.1	Group#6 WallE(s)	Added description for the user interface and a few parts of the sequence diagrams	23/04/23

1. Context Design

1.1 Context Model

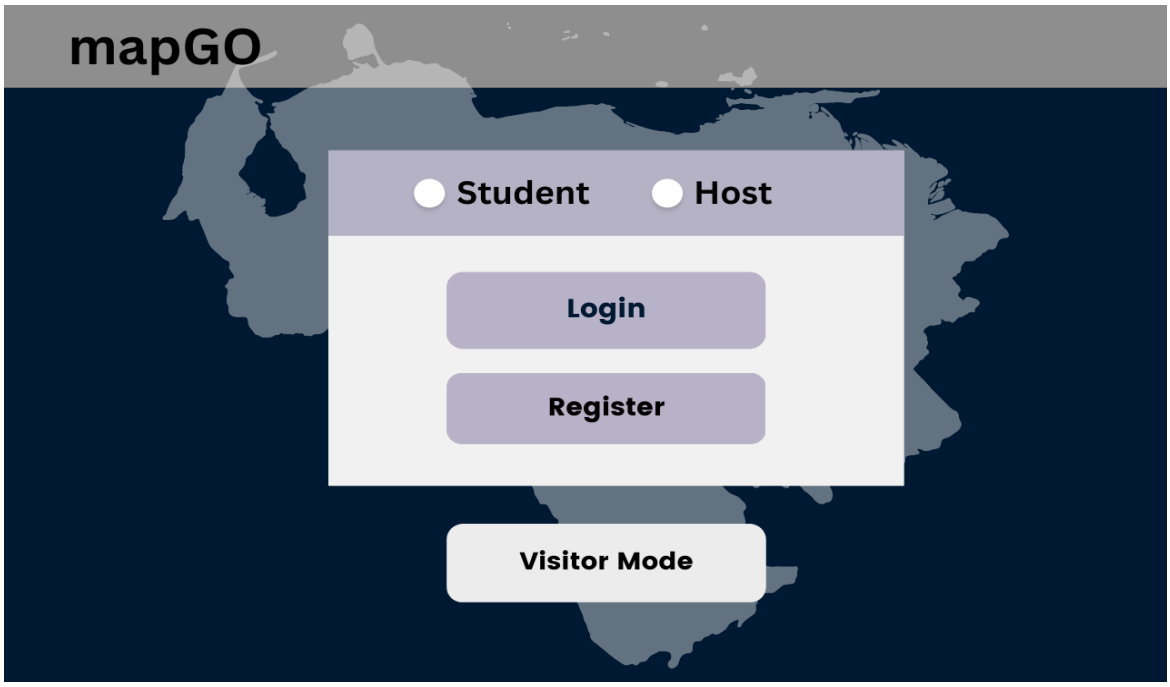


- The context model mainly consists of the
  - Users
  - Hosts
  - mapbox API
  - Admin
  - Database system

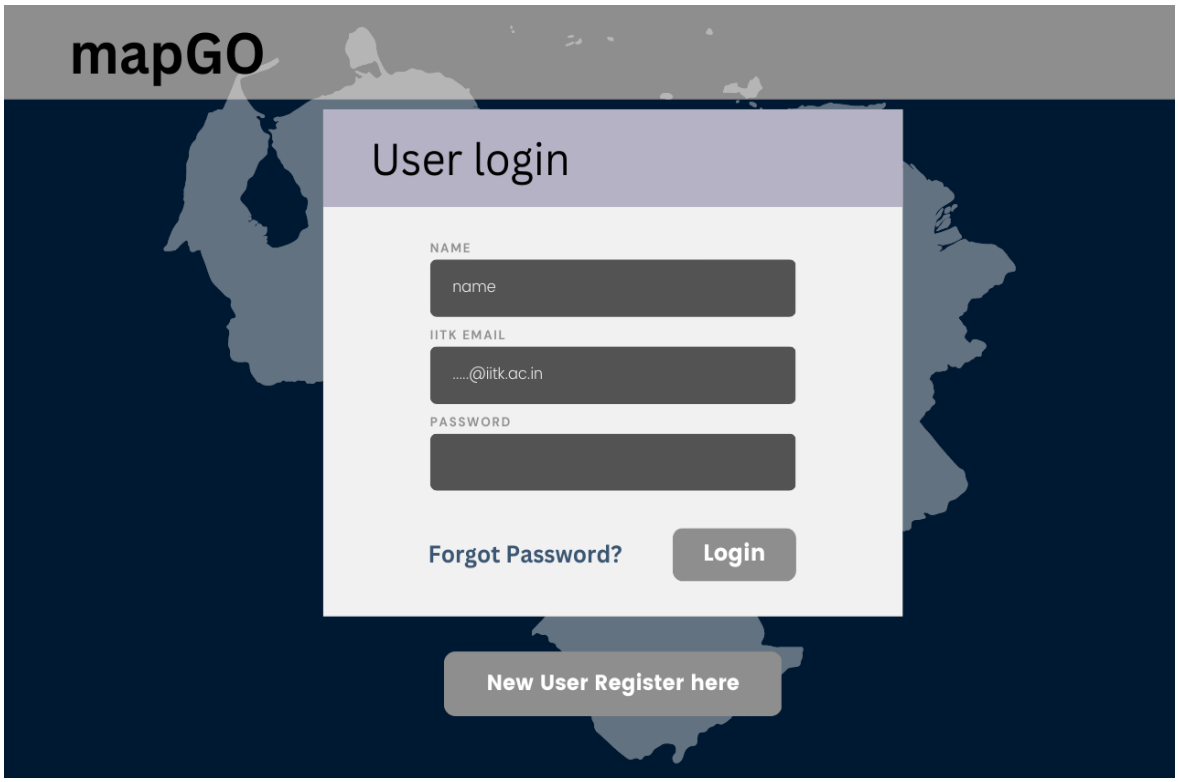
  These components interact with each other to make MapGo.
- The Database consists of the following:
  - User Information
  - Host credentials
  - List of channels
  - List of subscriptions
  - List of events and associated details
  - Log of forums
- It mainly interacts with our software and is used by the admin to verify channels being created by the hosts.
- The users and hosts interact with each where hosts create channels which could be joined by the users.
- MapGo uses mapbox API to get the map of our campus which is used by the users to navigate around the campus.

1.2 Human Interface Design

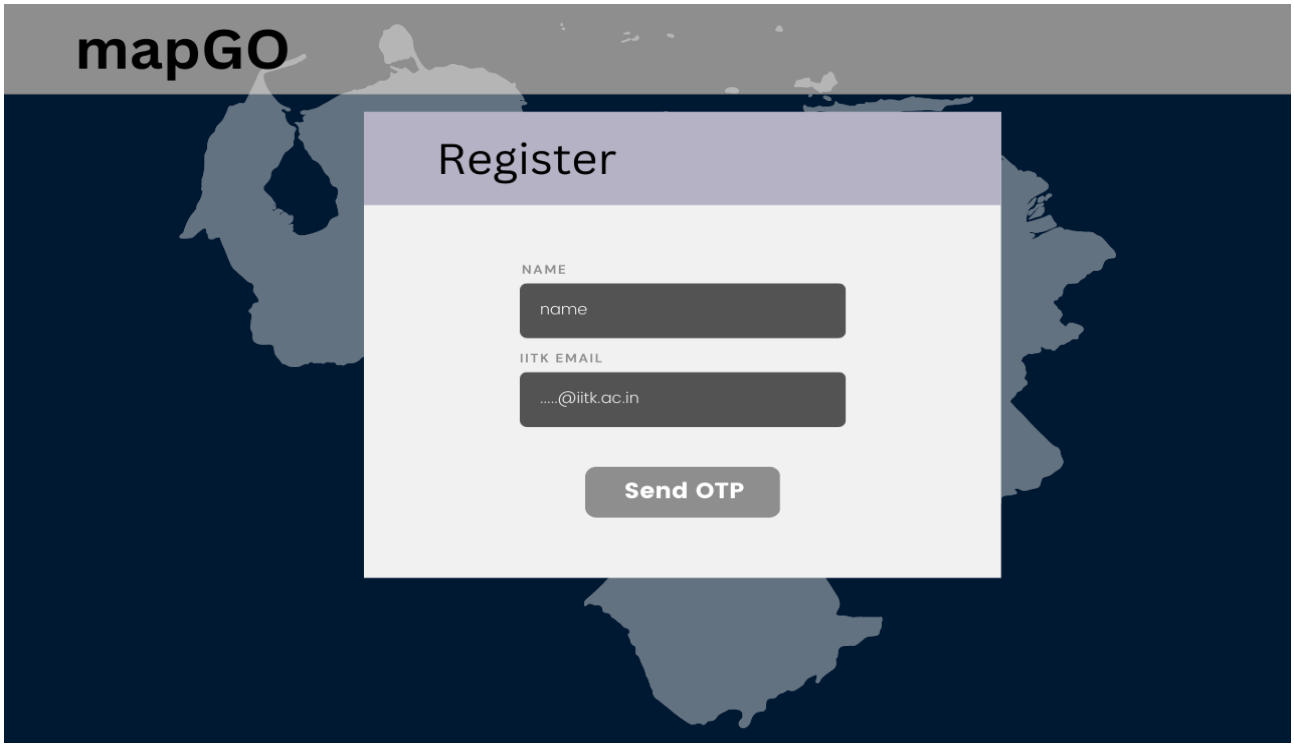
1.3 First page that pops up when mapGo is opened



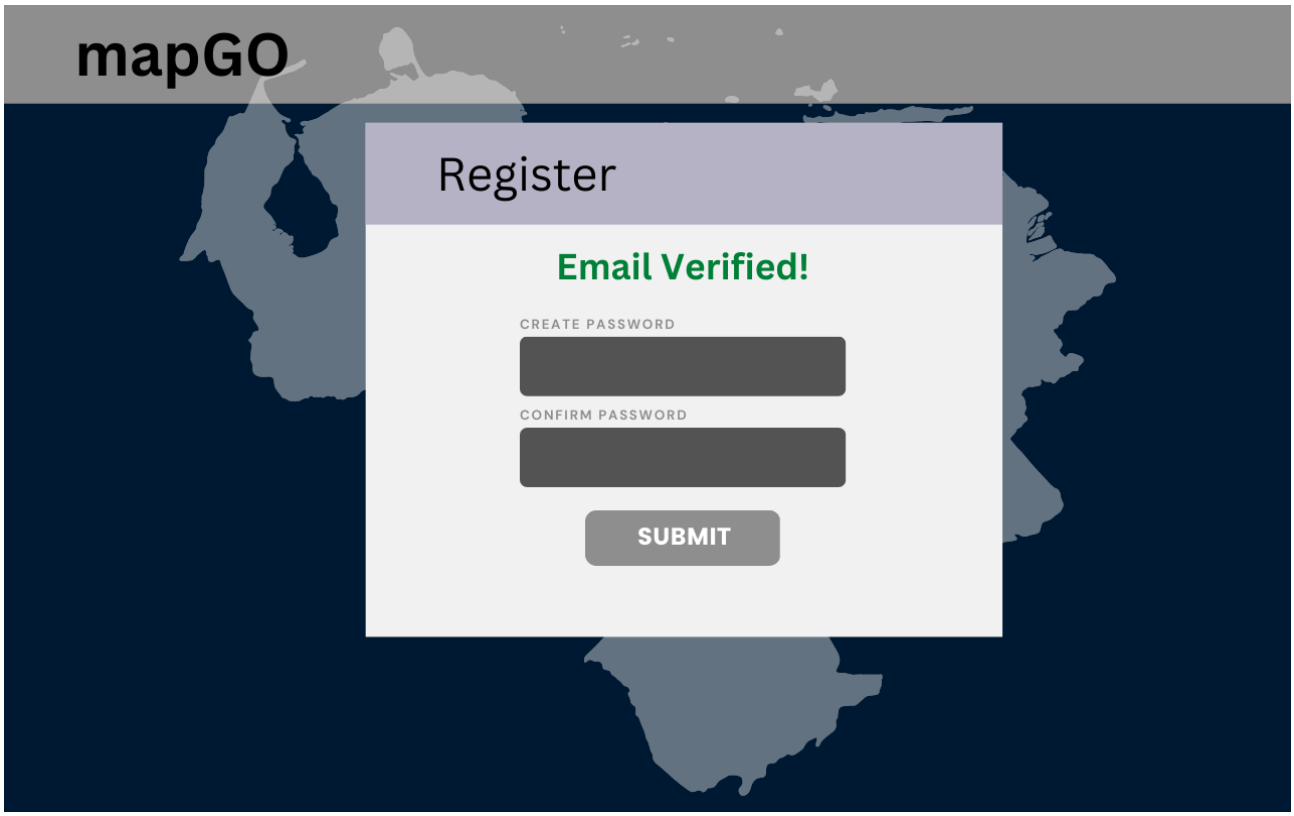
1.4 User Login Page



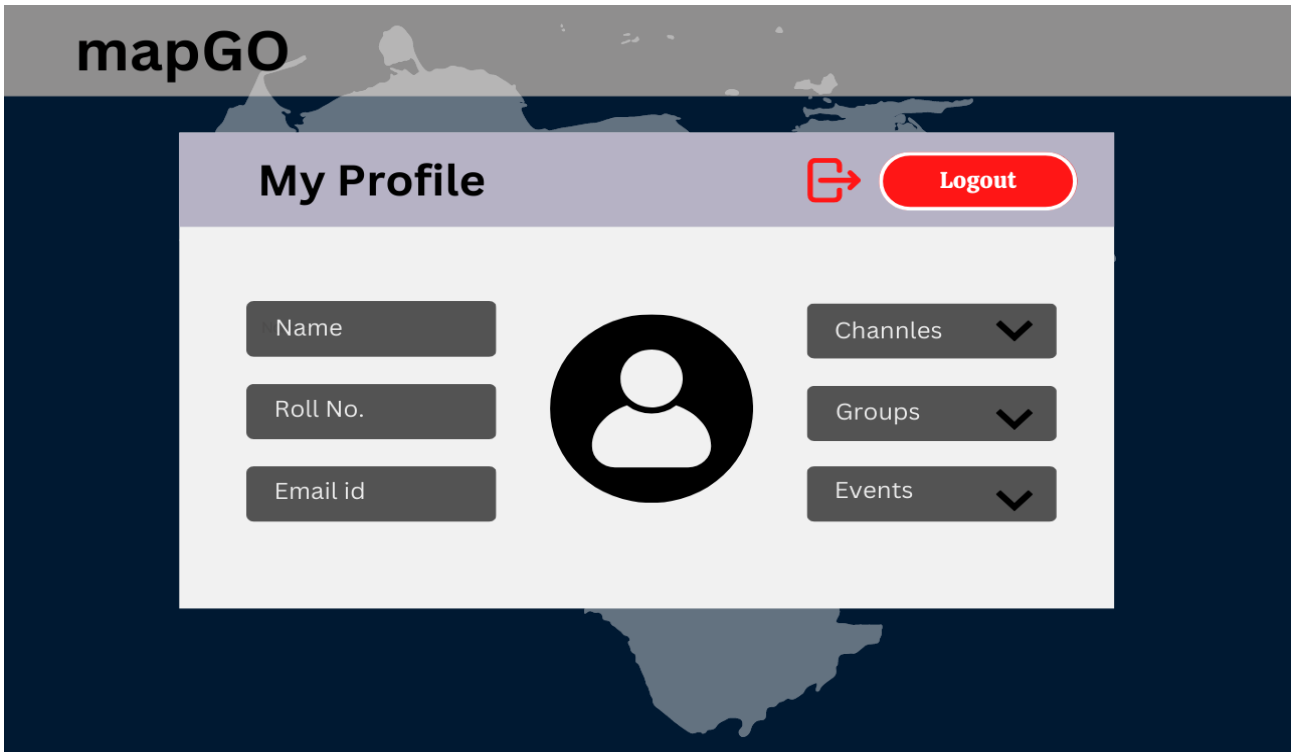
1.5 Registration Page



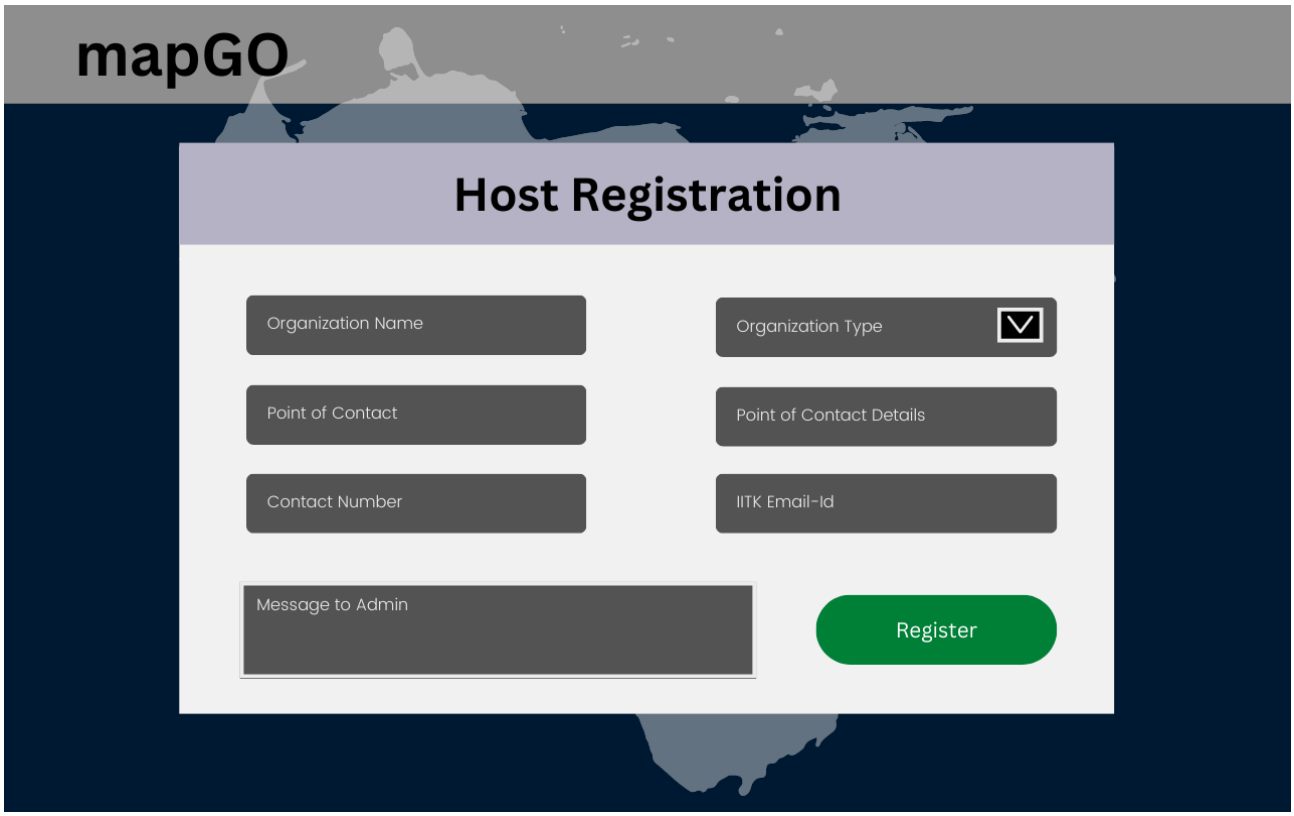
1.6 OTP Verification Page



1.7 Profile Page

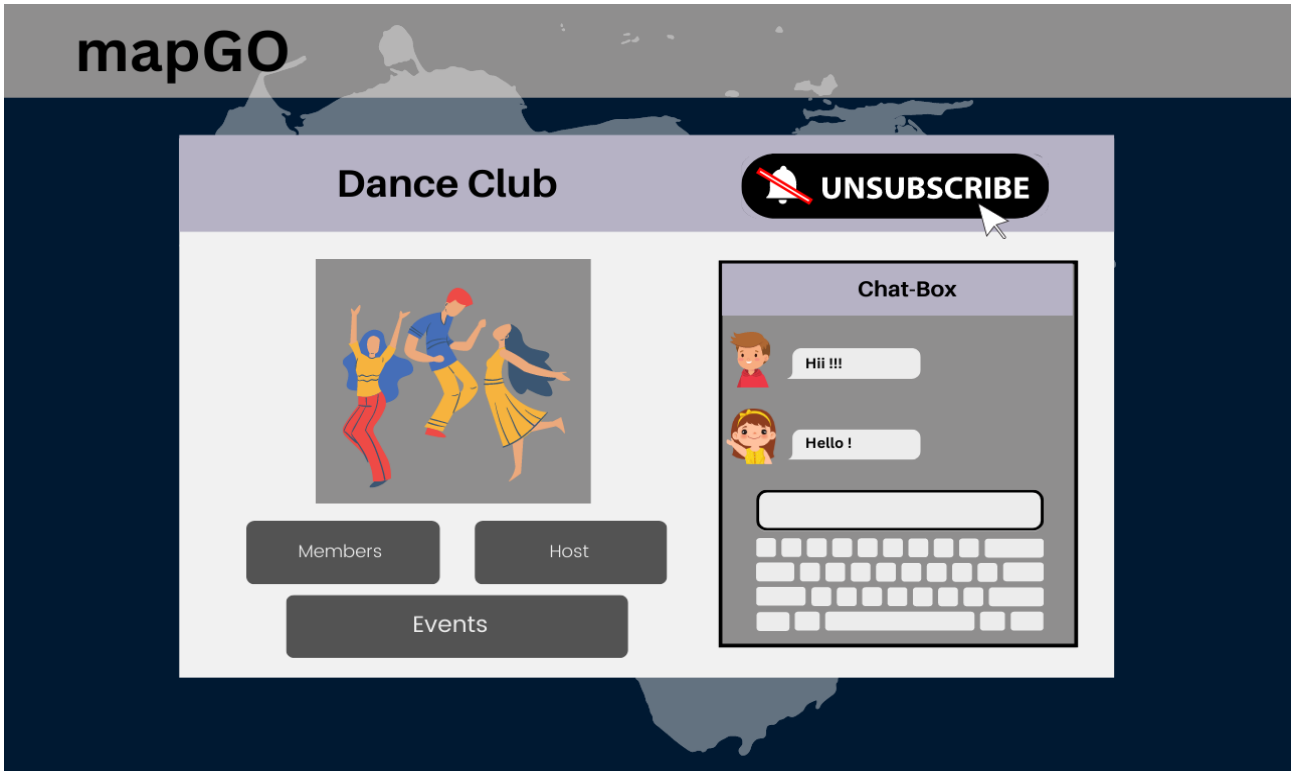


1.8 Host Registration Page

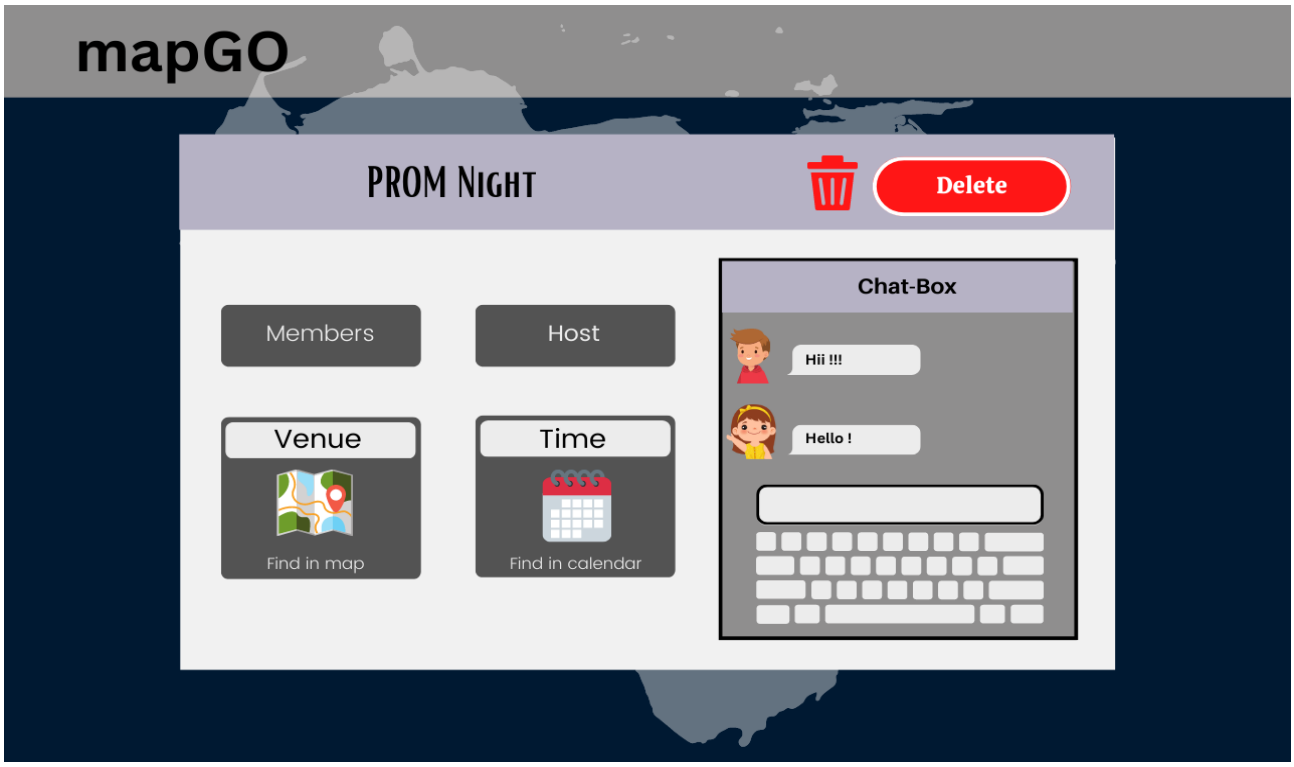


1.9 Shows all the channels we are subscribed to

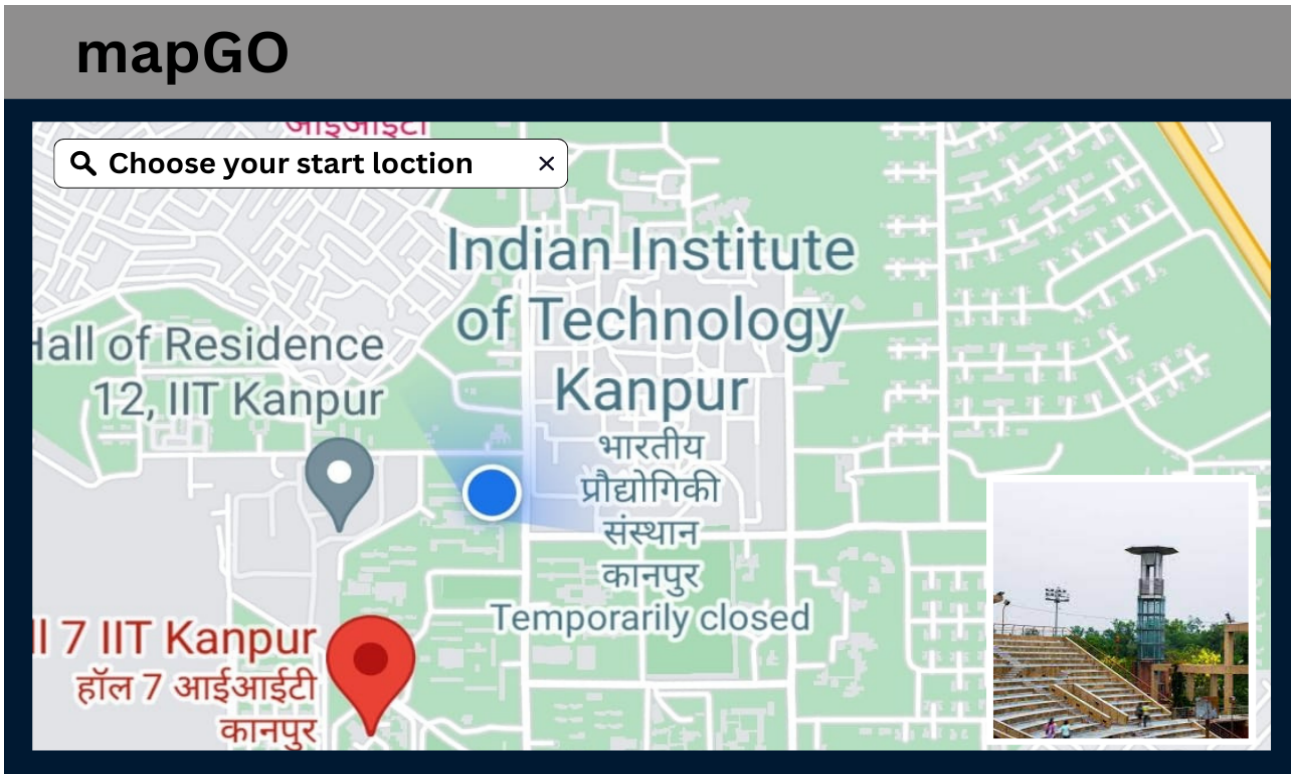
1.10 Page containing details of a specific page



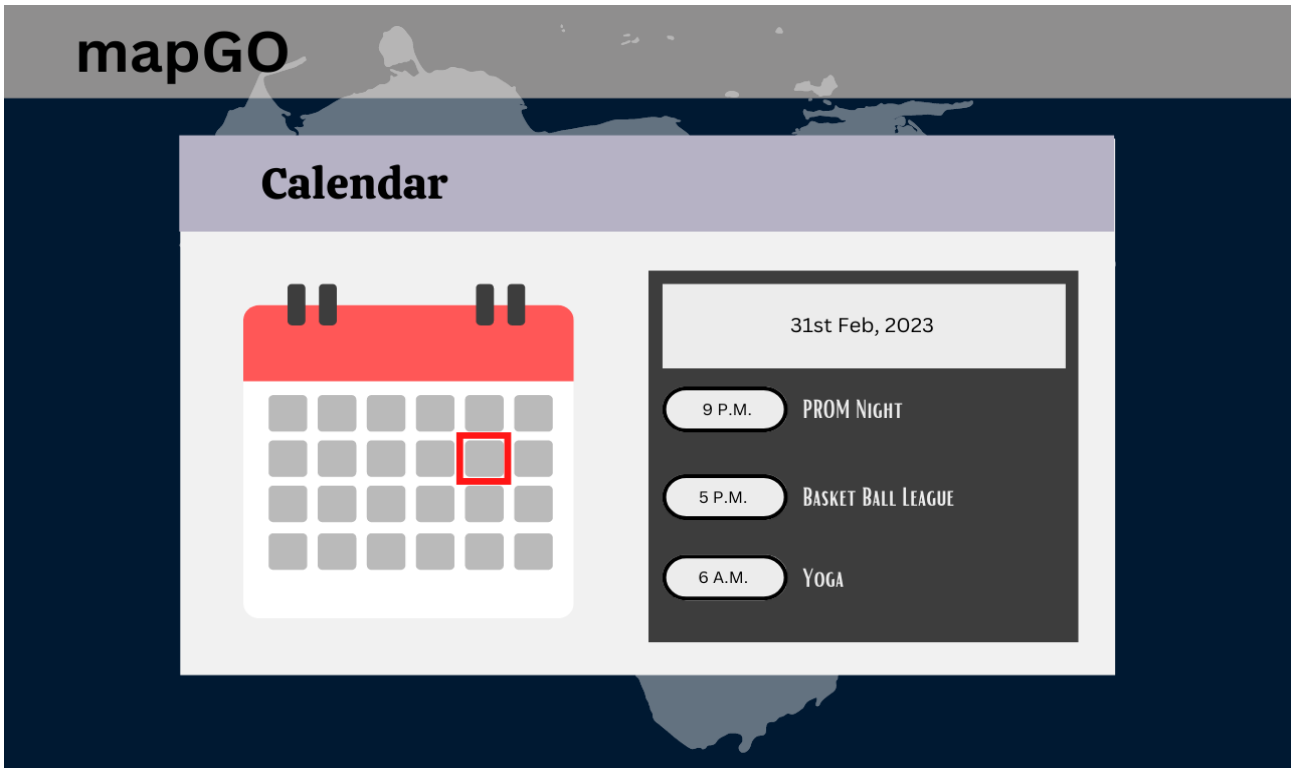
1.11 Page containing details of an event we are interested in



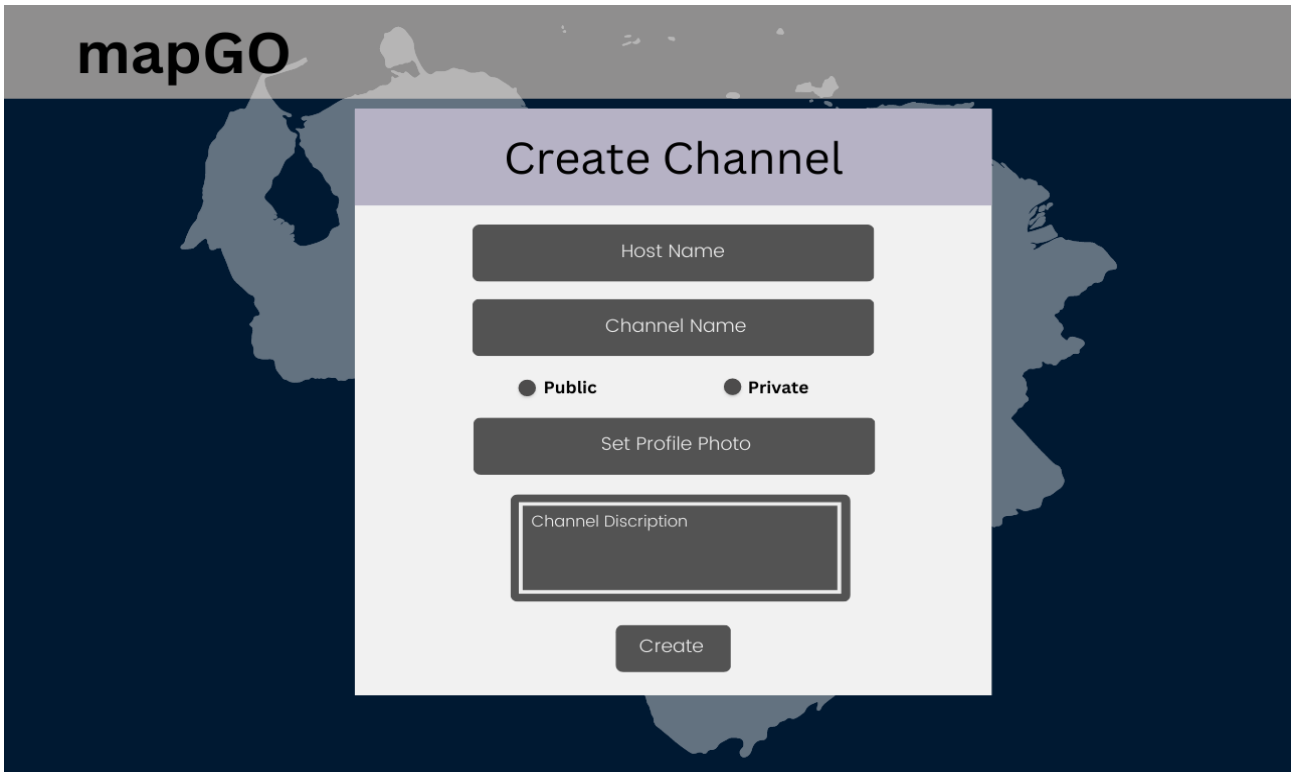
1.12 Navigation using the map



1.13 Calendar that shows our upcoming events



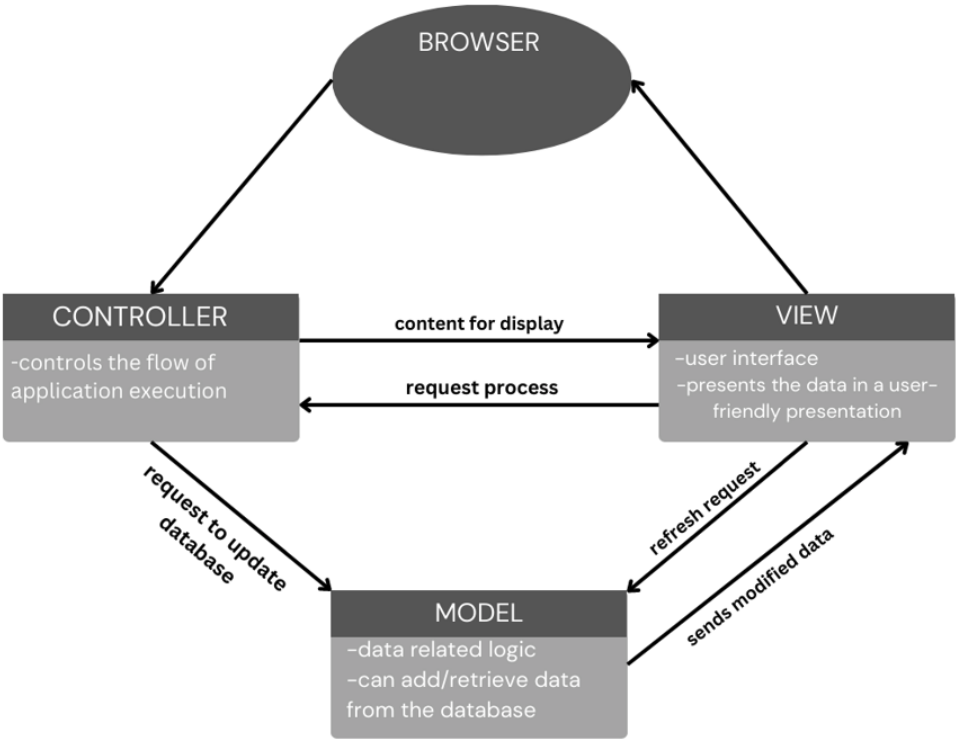
1.14 Creating a new channel



1.15 Creating a new group

1.16 Home Page that shows everything

● 2. Architecture Design



This application shall use the MVC Architecture: Model-View-Controller.
The reasons for the use of tis architecture:-

1. Easy development and maintenance: any future modifications can be amde easily as the programmer can find the component where a particular section of code is located based on its functionality.The components can be made simultaneously so it is the best for the web apps with a large team of web designers and developers. If we want to change something relating to one of the component we can do so easily without disturbing the other components.

2. Easier debugging process: The major advantage of using such code practices is that it helps to find specific portions of code quickly and allows the addition of new functionality with ease. .The main goal of this design pattern was to solve the problems of users controlling a large and complex data set by splitting a large application into specific sections that all have their own purpose. Segregation into 3 such components is helpful in the development process.

3. Extendability: future modifications and improvements in the application performance is also easy. In case of increasing the number of users, only the database size in the model block will be modified.

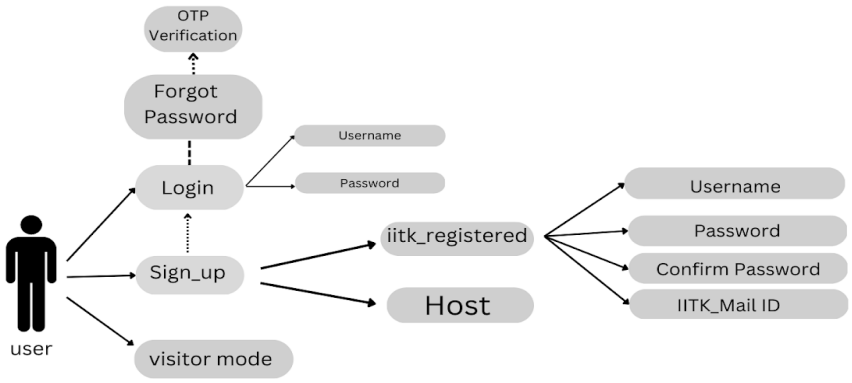
4. Response time: because of the separation into separate blocks asynchronous methods can be applied, hence execution takes lesser amount of time.

● 3. Object Oriented Design

● 3.1 Use Case Diagrams

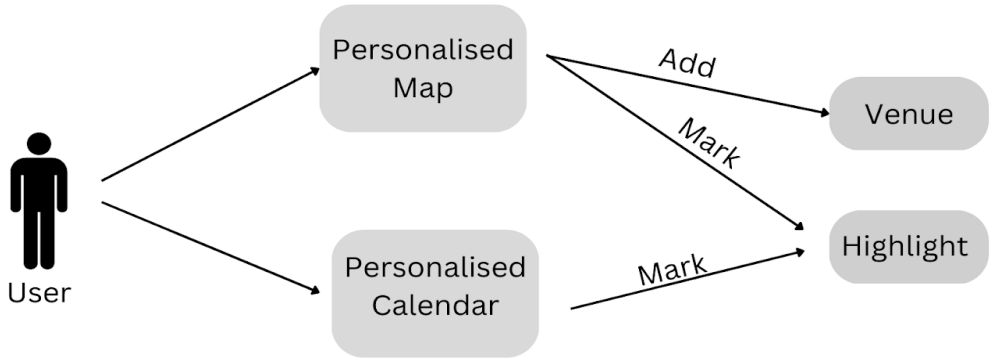
3.1.1 User registration, login and forgot credential use

Purpose: The user shall be able to register to use the application and login later with the credentials. in case the user forgets the password, they shall be able to use their account after verification



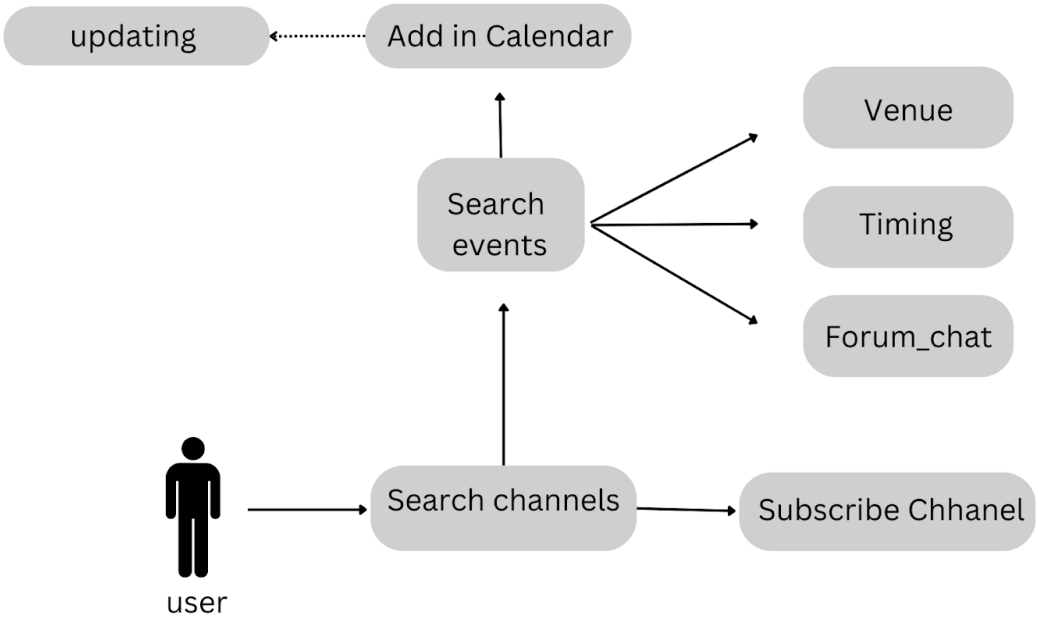
3.1.2 Personalised calendar and map

Purpose: a user shall edit and maintain a personalised schedule.

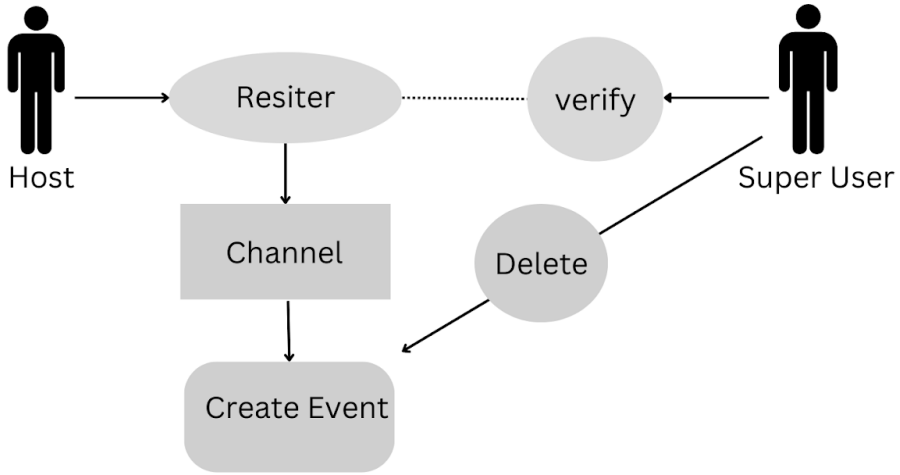


3.1.3 Interaction with channel and map

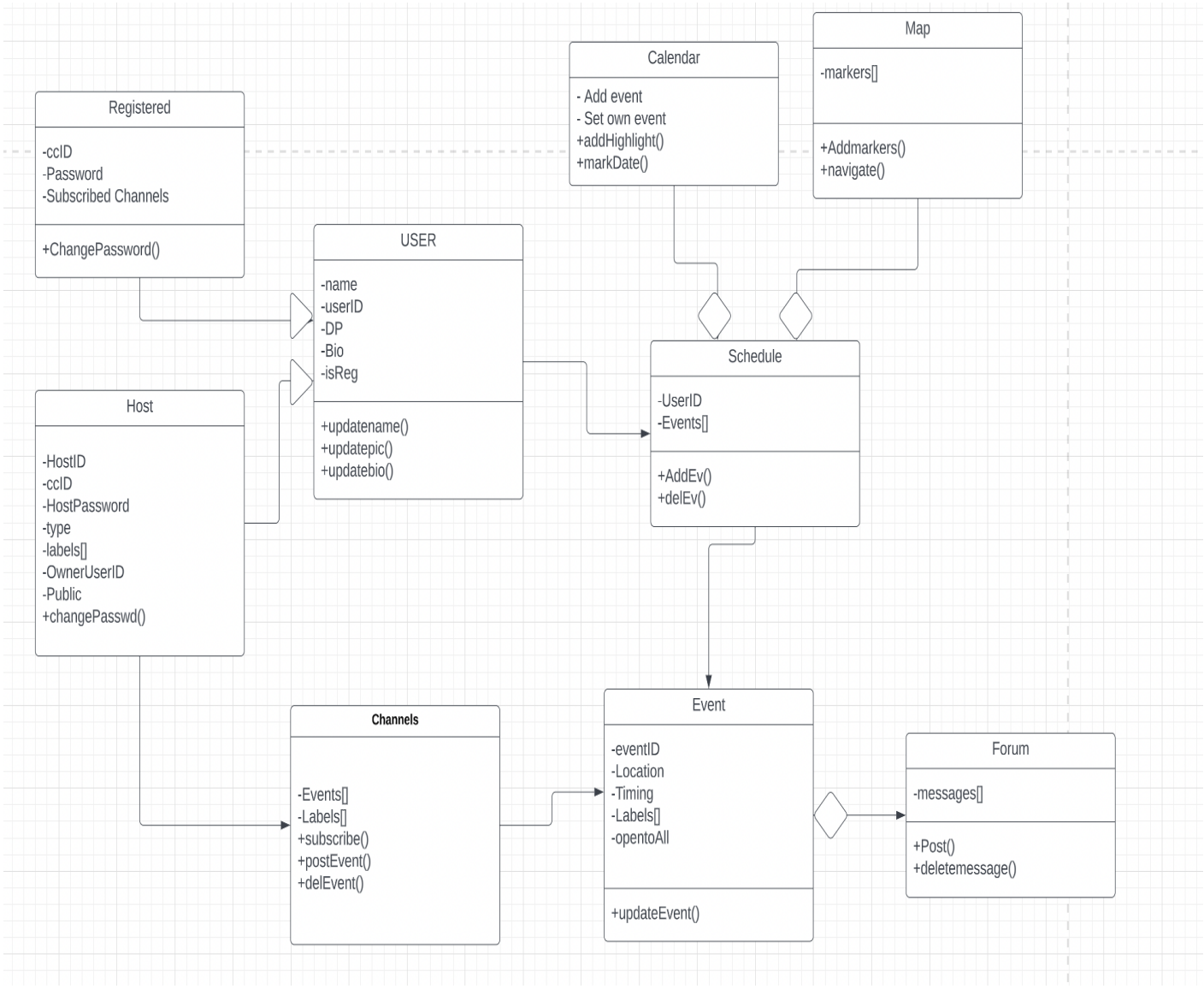
Purpose: A user shall be able to search for events/channels and add it to their schedule.



3.1.4 Host registration, verification and announcement of events

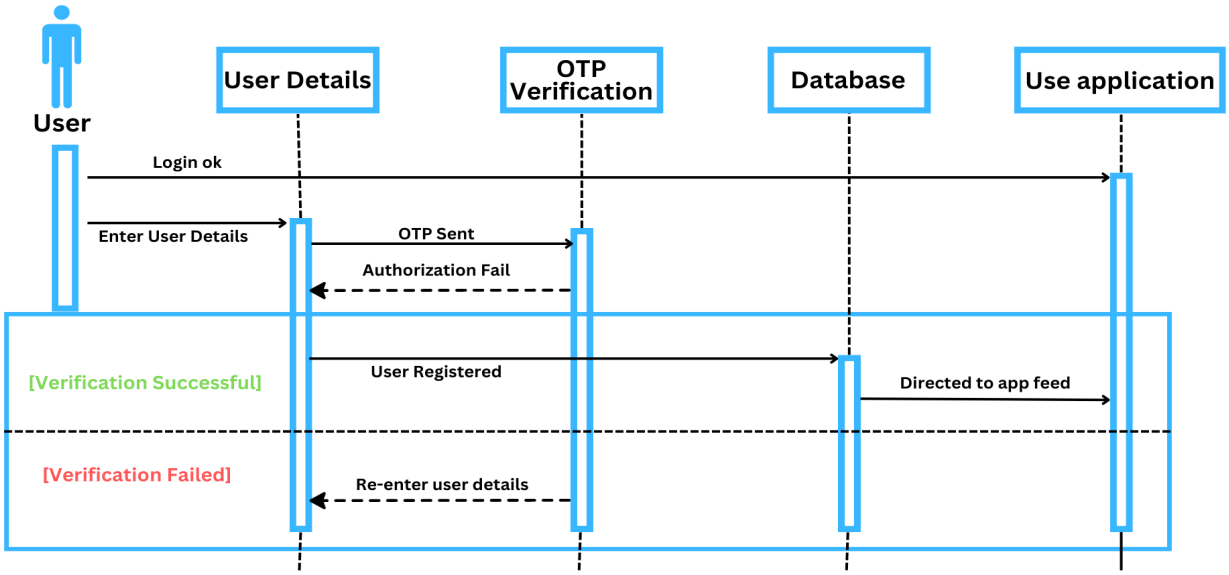


- ## 3.2 Class Diagrams

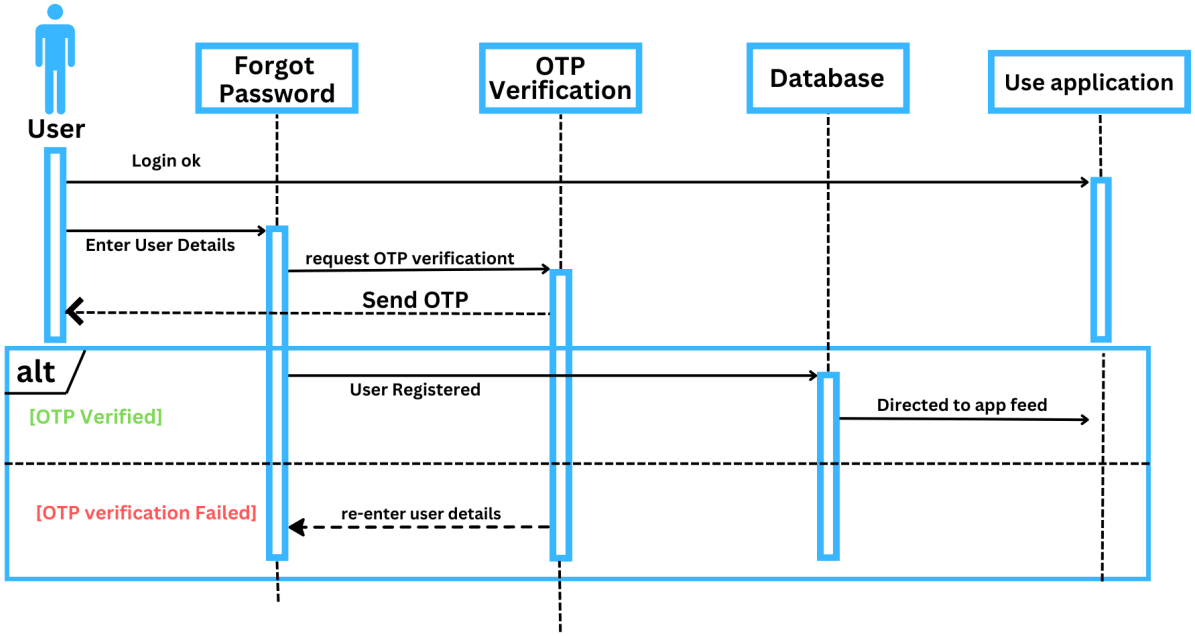


- ## 3.3 Sequence Diagrams

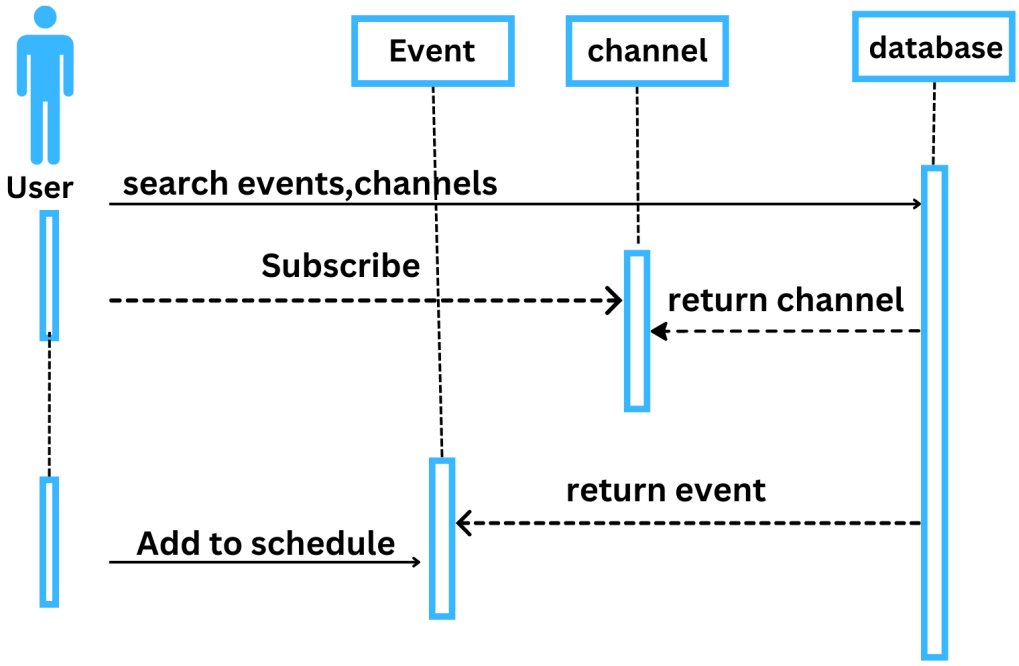
3.3.1 User registration and login process into the application



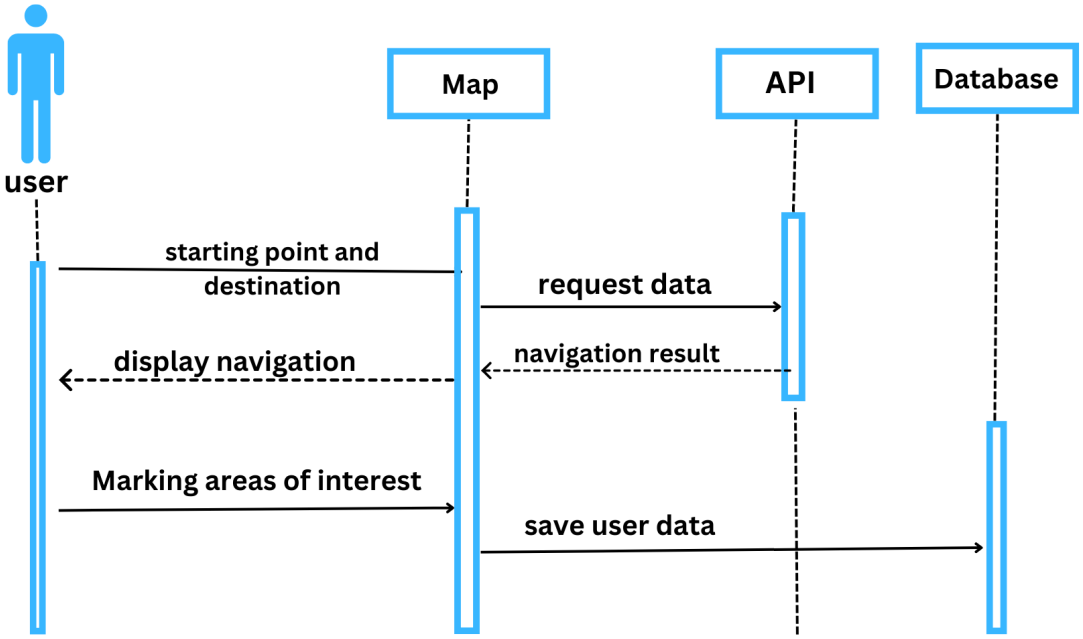
3.3.2 Forgot password feature of the application.



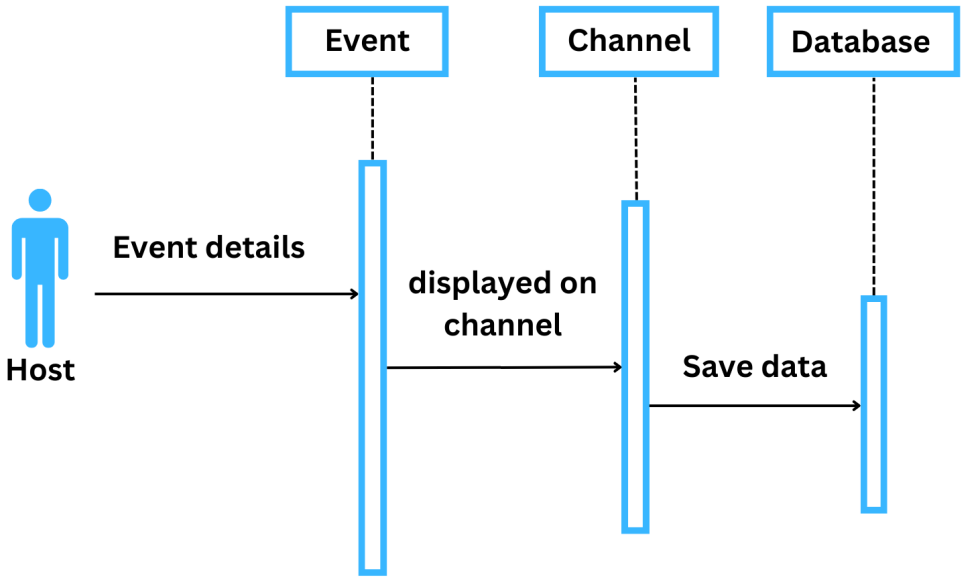
3.3.3 searching events, channels that a user may be interested in.



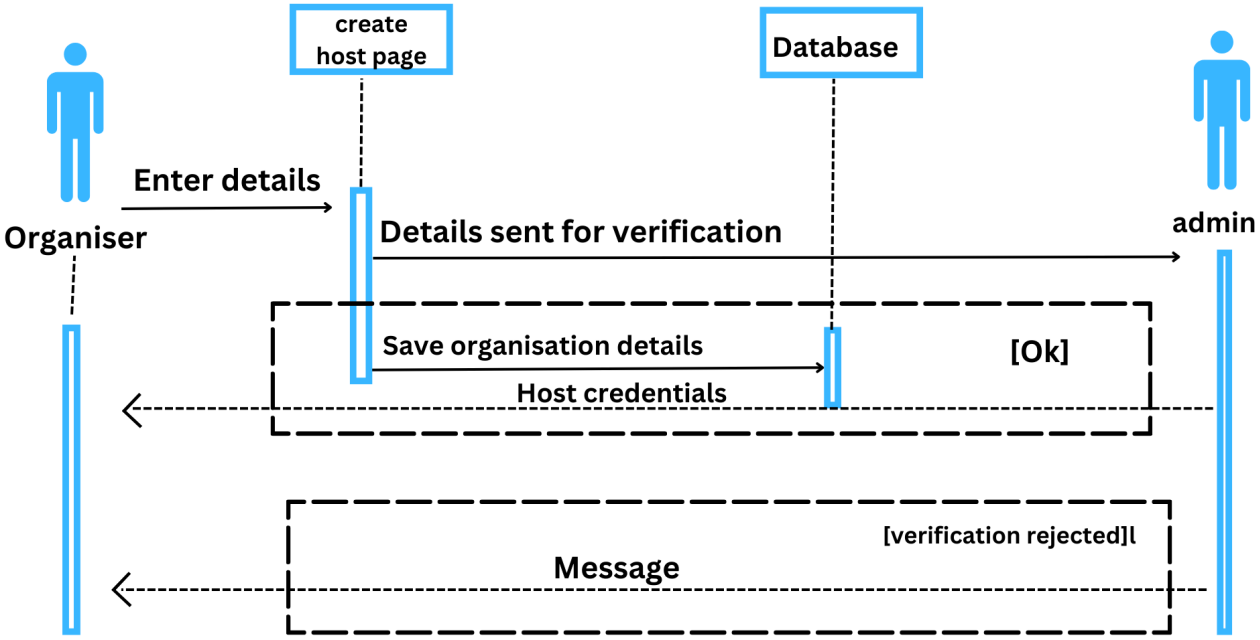
3.3.4 Using the navigation feature of the map, and customising the map



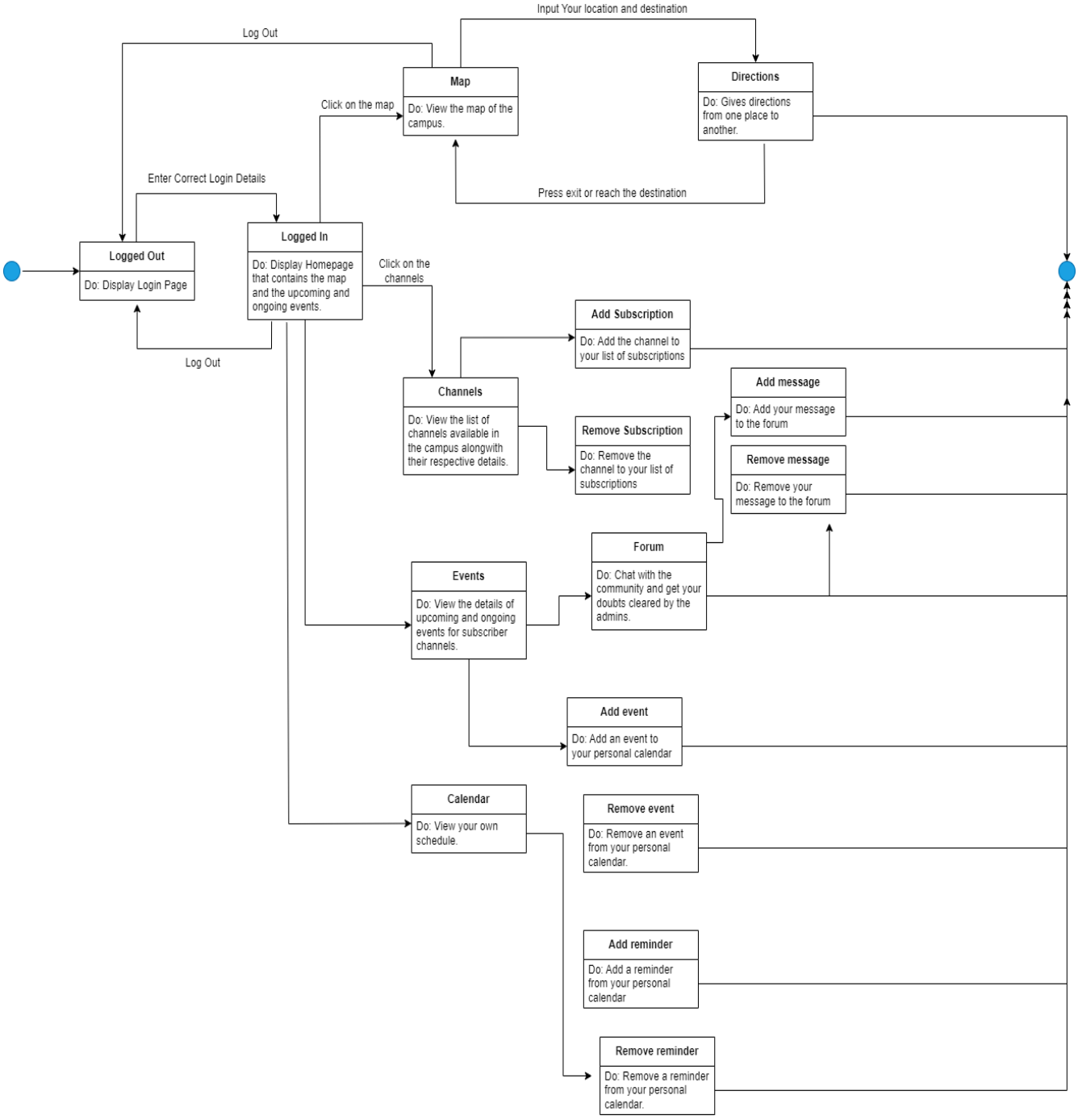
3.3.5 Host announcing an event.



3.3.6 organiser creating a host page, verification from the admin and providing a channel for the organiser to announce events.



- ## 3.4 State Diagrams



● Project Plan

The project execution timeline can be broadly divided into the following 7 major tasks:

1. **Implementation**

Time: February 11, 2022 - February 10, 2022

Discussion over the division of tasks involved in the programming of the application. Major tasks include writing the source code, studying materials on topics necessary, researching the implementation of similar systems. This task requires repeated learning plus implementation routines

Responsible member: Rohan Ravi

2. Unit testing:

Time: February 10, 2022 - February 13, 2022

Testing the functioning of each individual component, modules that are programmed to execute a particular task.

Responsible member:Abhinav Garg

3. Integration testing:

Time: February 13, 2022 - February 17, 2022

Testing the functioning of a group of modules that work together in order to produce the desired output. This stage intends to locate any faults in the communication between different units when working together.

Responsible member: Prachi Choudhary

4. System testing:

Time: March 17 2022 - February 21, 2022

Testing the functioning of the entire system. Thai period can also be used to gain feedback from a sample set of end-users, whose input can be used in the code improvement stage.

Responsible member: Harshini Dola

5. **Manual preparation for beta testing:**

Time: February 22, 2022 - March 31, 2023

Compiling a manual which explains the application details necessary for beta testing team.

Responsible member: Modem Shanwitha Yadav

6. Code improvement:

Time: April 1, 2022 - April 7,2022

Any feedback from the beta testing team, end-user etc. will be processed and used to better the application.

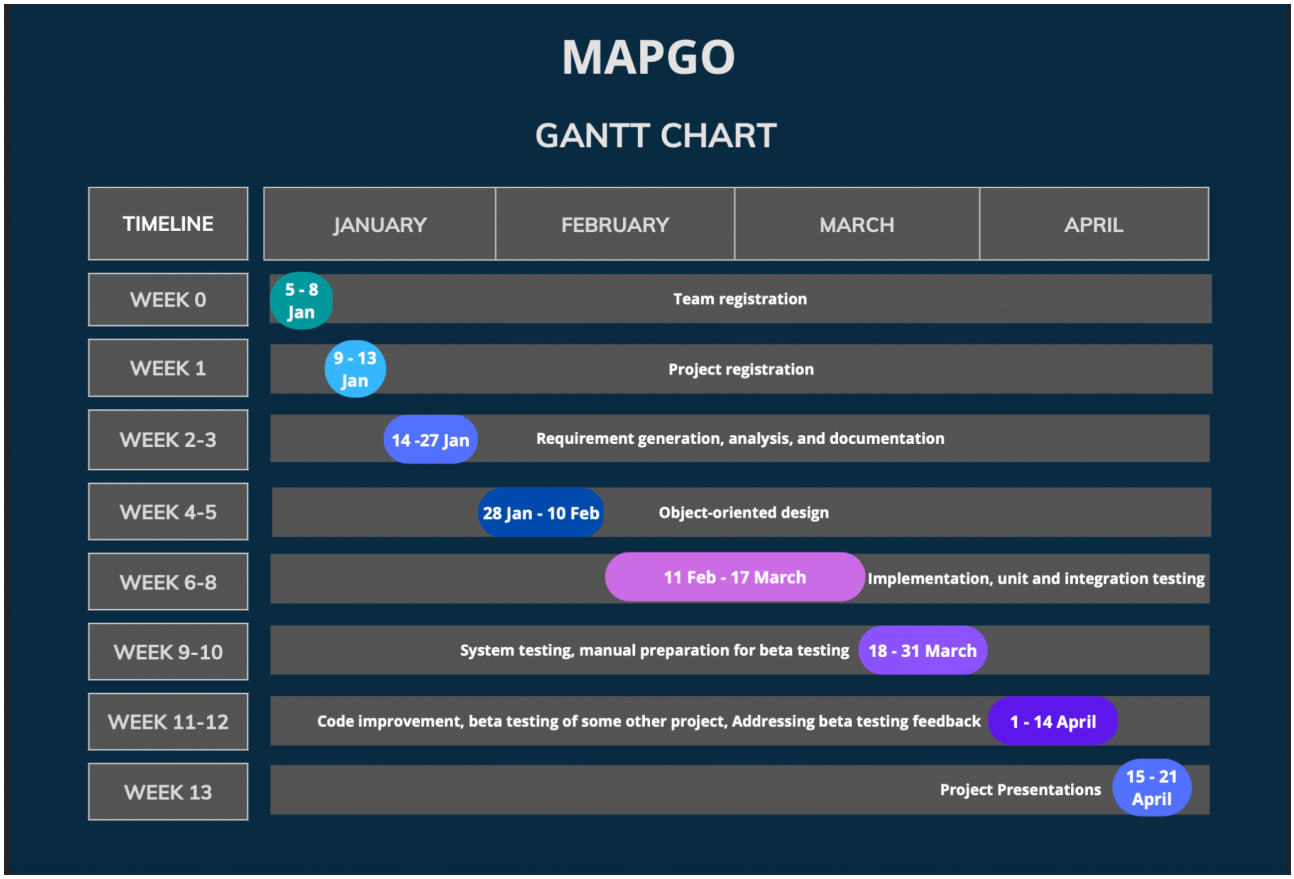
Responsible member: Ujjwal Gautam

7. Beta testing of some other project:

Time: April 8, 2022 - April 14,2022

Following the manual provided by the other team, beta testings will be carried out on the application produced by that team.

Responsible member:Yashas



● Other Details

Appendix A - Group Log

S no.	Date	Description
1	January 28, 2022	Studied the design document template provided by the course instructor.
2	January 30, 2022	Group meeting: discussion on what should be mentioned in each section of the document. Division of work among members.
3	February 2, 2022	Group meeting: discussion of the tech stack to be used in the project. Researched resources online to learn the required tech needed.
4	February 8, 2022	Group meeting with project mentor: asked doubts regarding the contents in the document. Clarified if the current material had any faults. Received validations and remarks from the mentor.
5	February 9, 2022	Redistribution of the remaining tasks
6	February 10, 2022	Final touches and submission of the document.