Design Document

for

What's Next

Version 1.1

Prepared by

Group 3: **Group Name:** ***Codecrafters***

Aditya Kumar	**210060**	**adityakum21@iitk.ac.in**
Akkinepally Kruthi	**210088**	**akruthi21@iitk.ac.in**
Aman Arya	**210106**	**aarya21@iitk.ac.in**
Apoorva	**210179**	**apoorvag21@iitk.ac.in**
Chitwan Goel	**210295**	**chitwang21@iitk.ac.in**
Geetika	**210392**	**geetika21@iitk.ac.in**
Krish Sharma	**210530**	**krish21@iitk.ac.in**
Paras Sikarwar	**210699**	**sparas21@iitk.ac.in**
Siddharth Kalra	**211032**	**siddharthk21@iitk.ac.in**
Talin Gupta	**211095**	**taling21@iitk.ac.in**
Varun Tokas	**211152**	

Course: **CS253**

Mentor TA: ***Archi Gupta***

Date: **10/02/2023**

Contents II

Revisions II

1 Context Design 1

1.1 Context model 1

1.2 human Interface Design 1

2 Architecture Design 2

3 Object-Oriented Design 3

3.1 Use case diagram 3

3.2 Class diagram 3

3.3 sequence diagram 3

3.4 State diagram 3

4 Project Plan 4

5 Other Details 5

Appendix A - Group Log 6

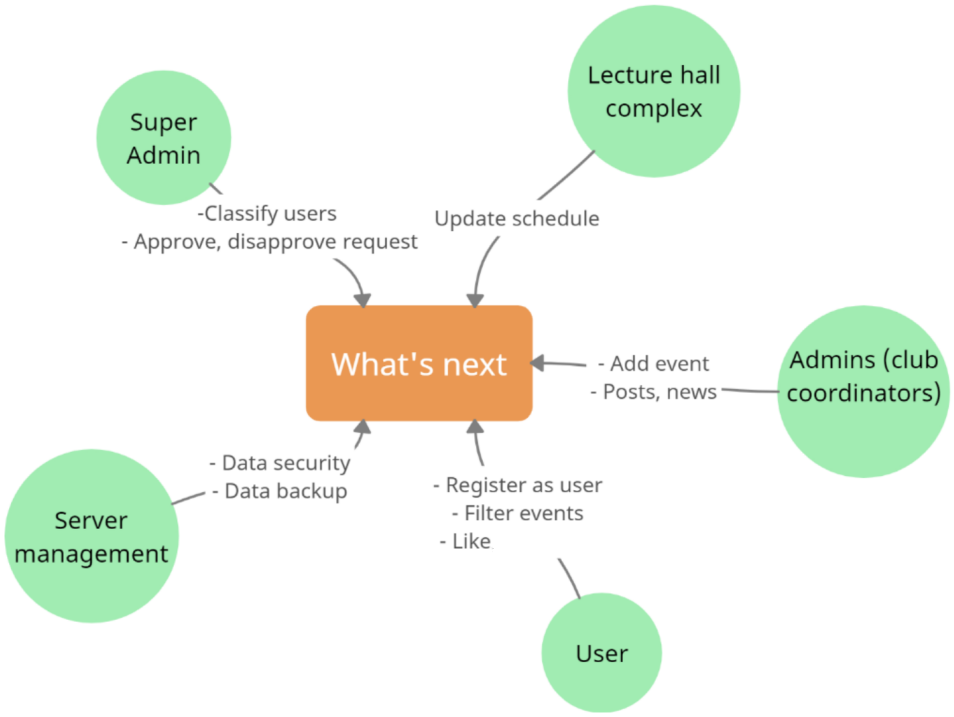
Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	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.	00/00/00

1 Context Design

1.1 Context Model

A context model defines how context data is structured and managed. It plays a significant role in supporting efficient context management.

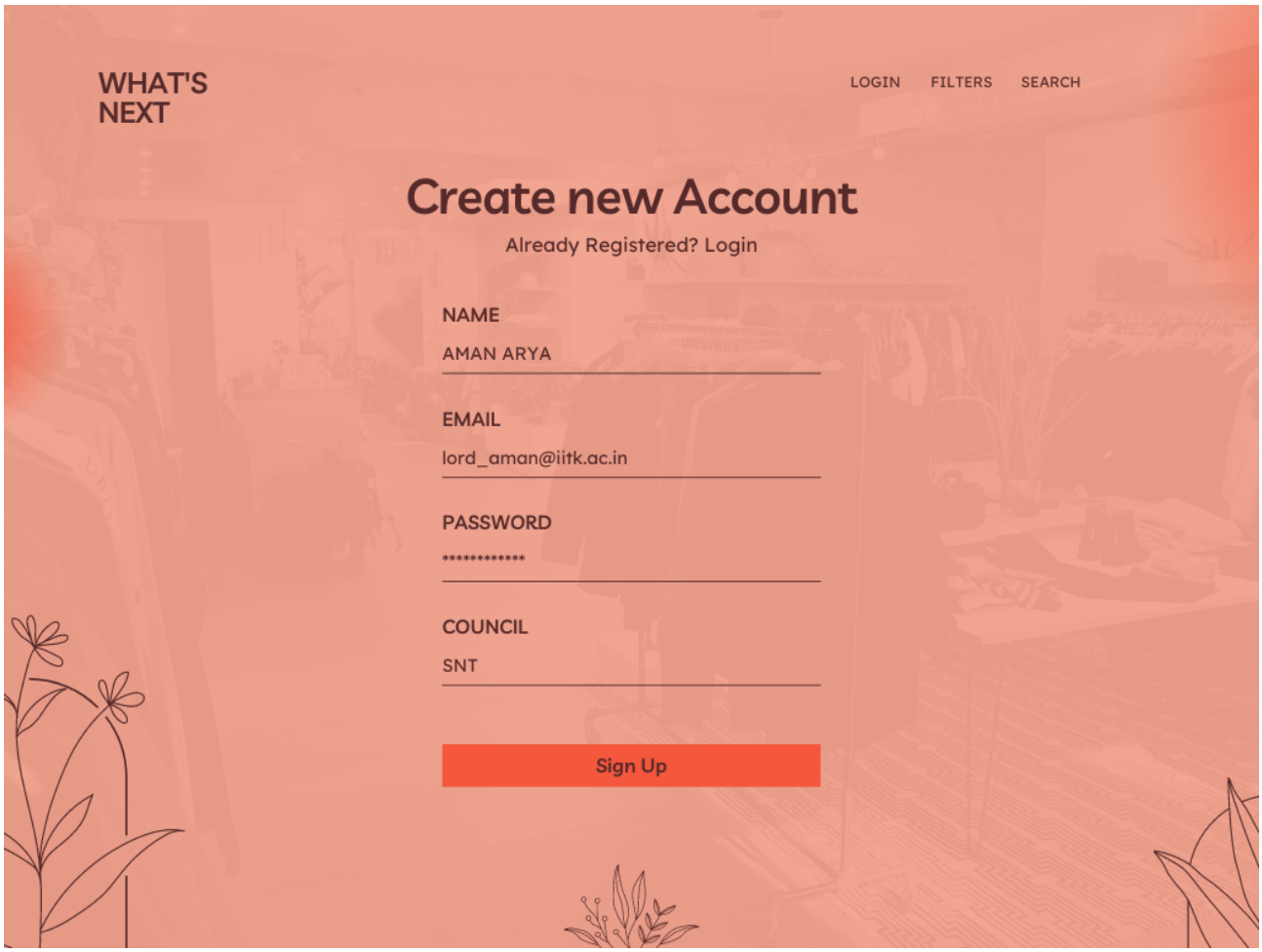


A brief explanation of What's Next context model is as follows:

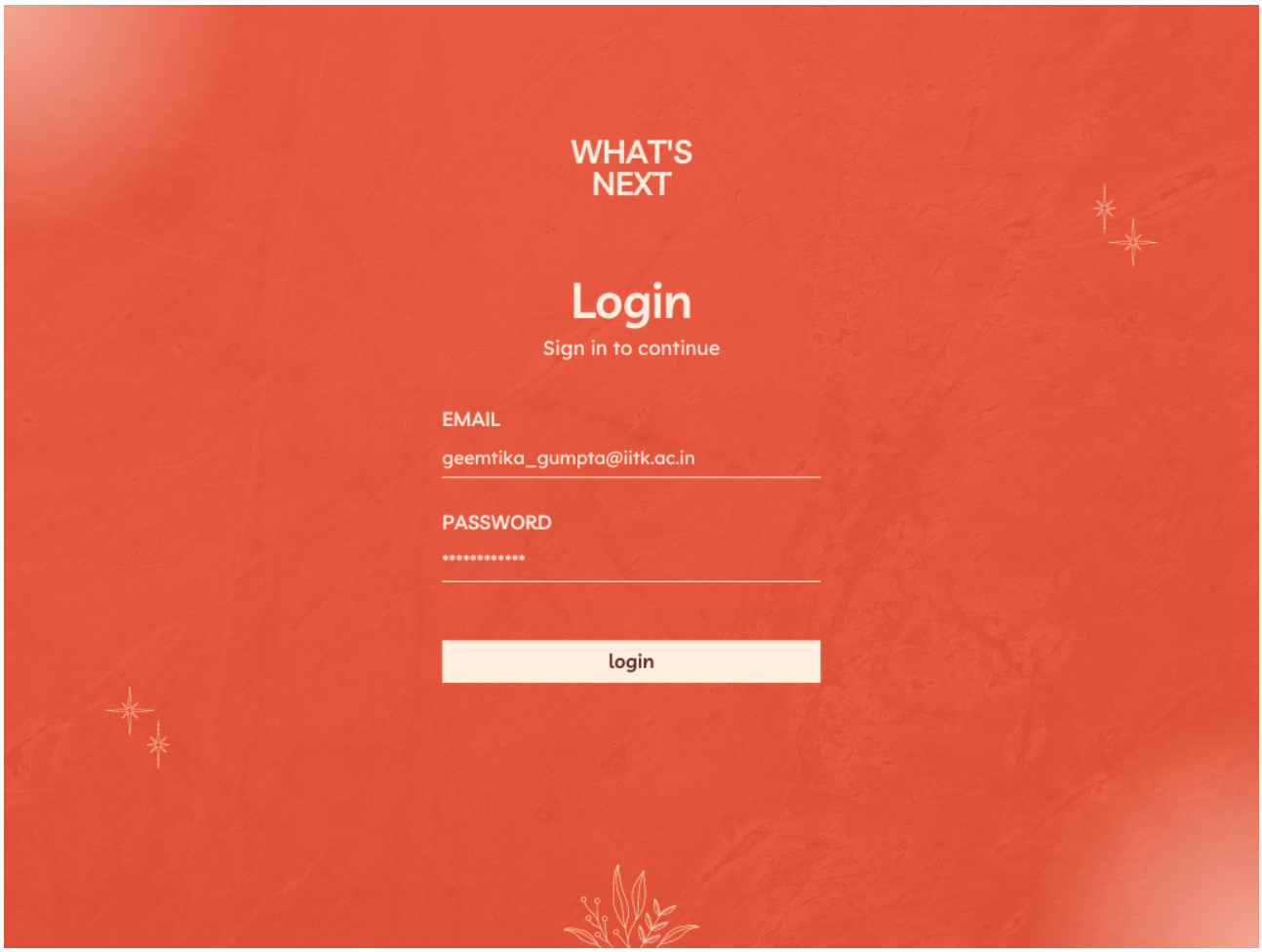
- **User Login:** A user will log into What's Next software while the software will receive login requests from multiple users.
- **Admin Login:** Admins will log into the software and see their dashboard.
- **Admin create event**: An admin will be able to schedule an event after logging in while the software will receive such requests from multiple admins.
- **Lecture Hall Complex**: Lecture hall admin will update the lecture schedule in the system database which will be reflected on the client's end.
- **Super Admin**: The super admin will classify the users as admins and general users, approve or disapprove special events requests and also manage the overall system in case there is a fault or issue.
- **Server:** The server will look into the database including backup and security of the software database.

1.2 Human Interface Design

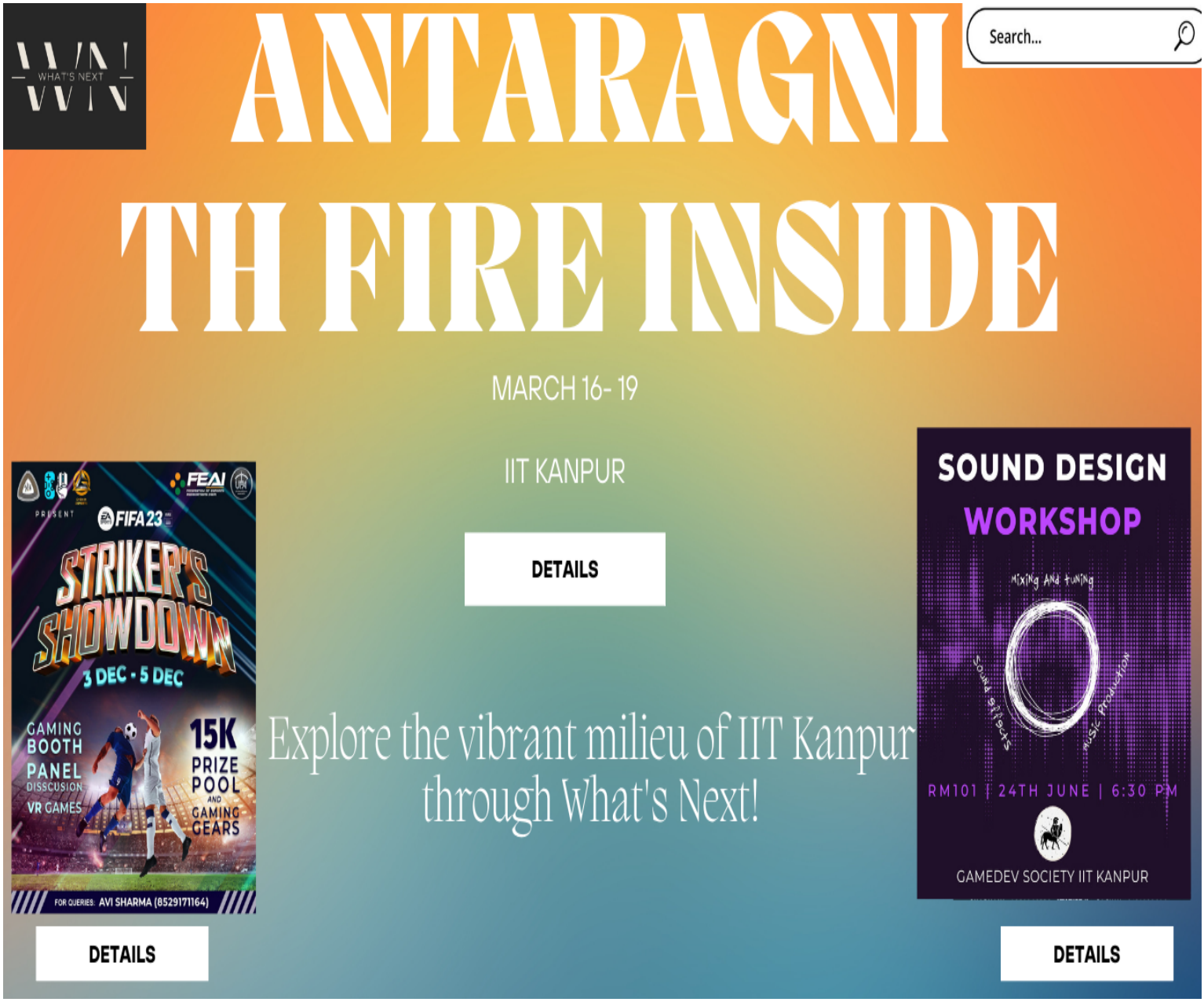
1. This is the Signup Page of our website



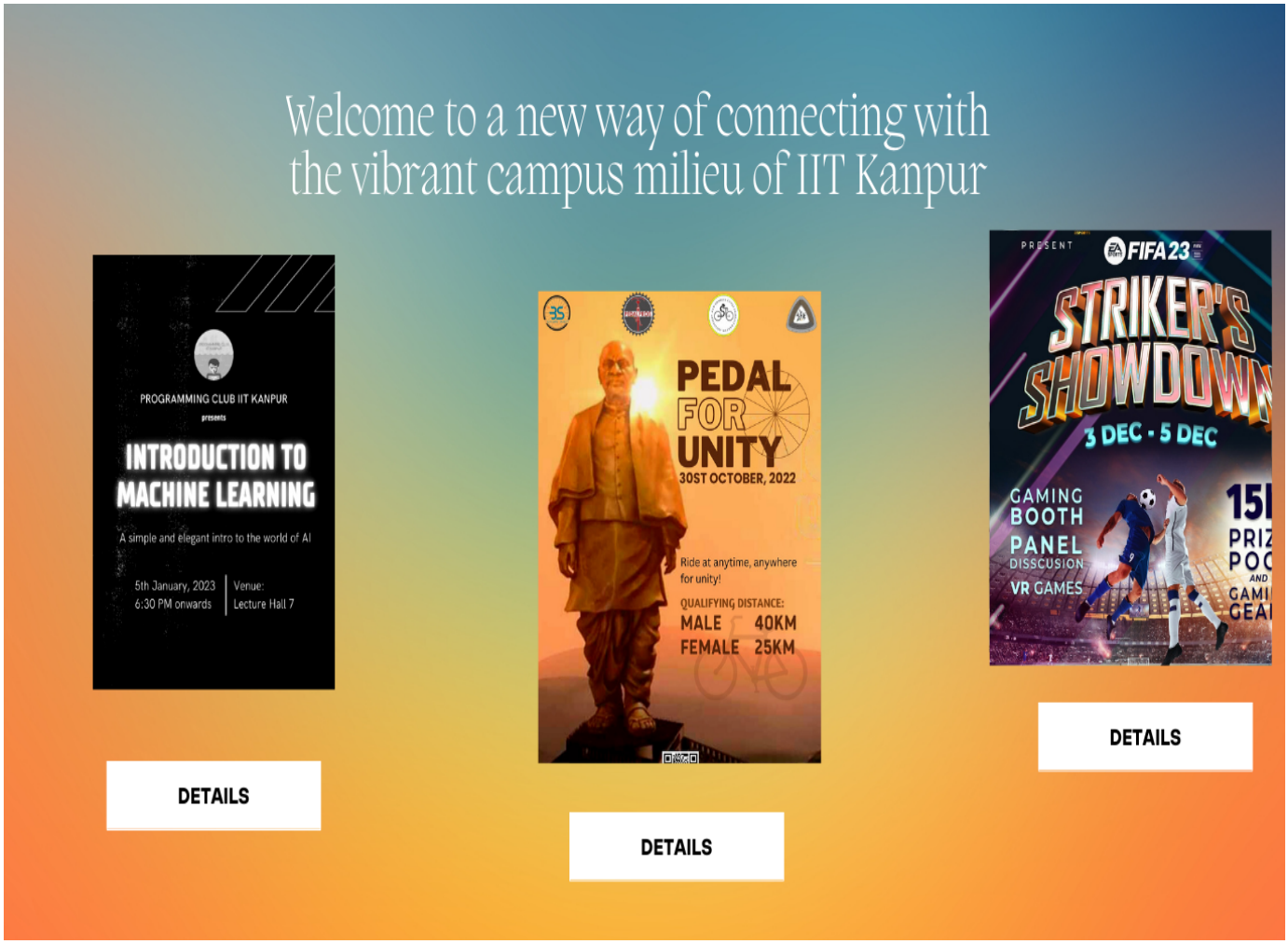
2. This is the LOGIN page that will be visible to any kind of user



3. This is the landing page of the website, this is the very first page that the user will be seeing.

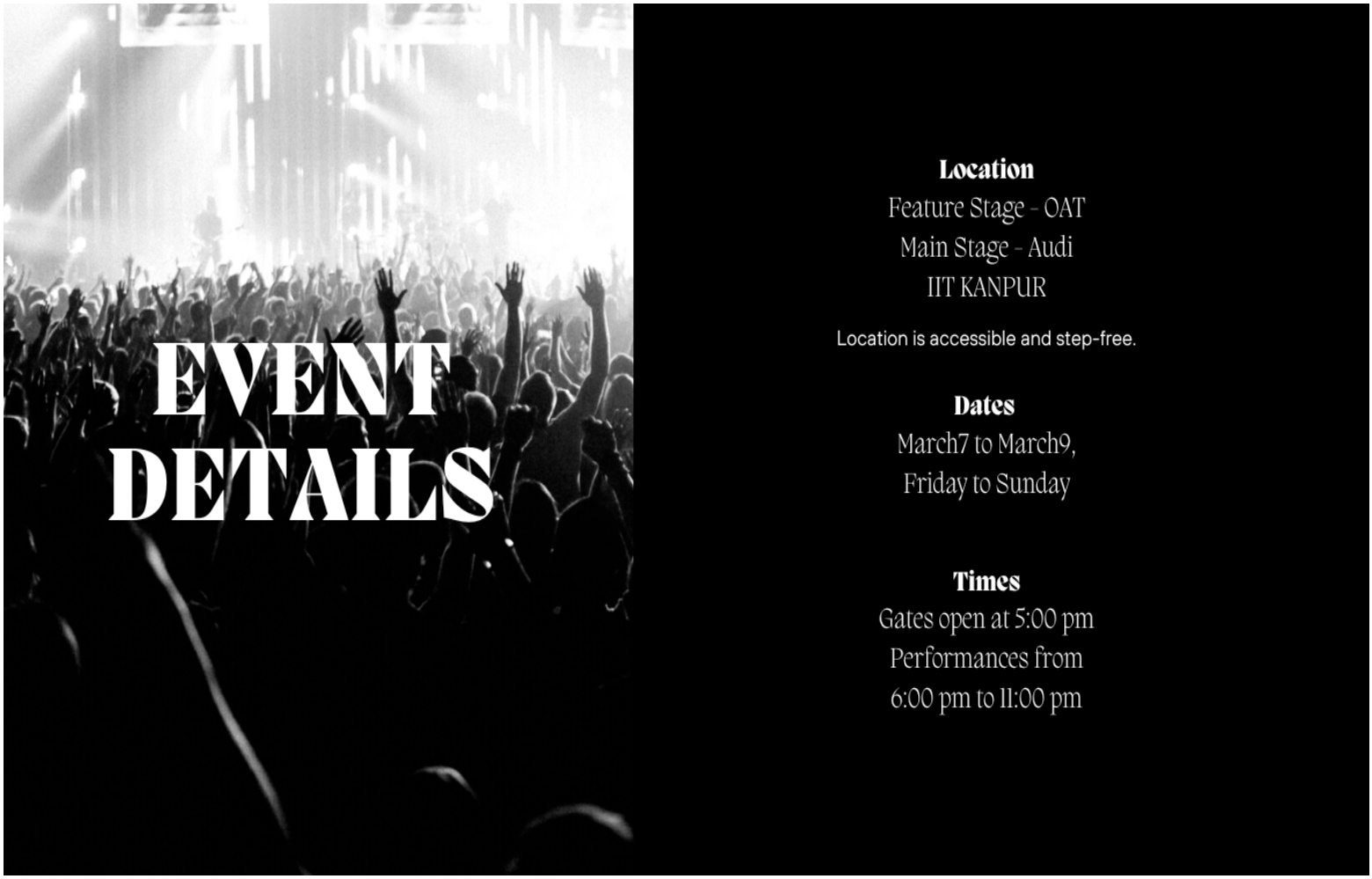


4. This is the continued landing page of the website

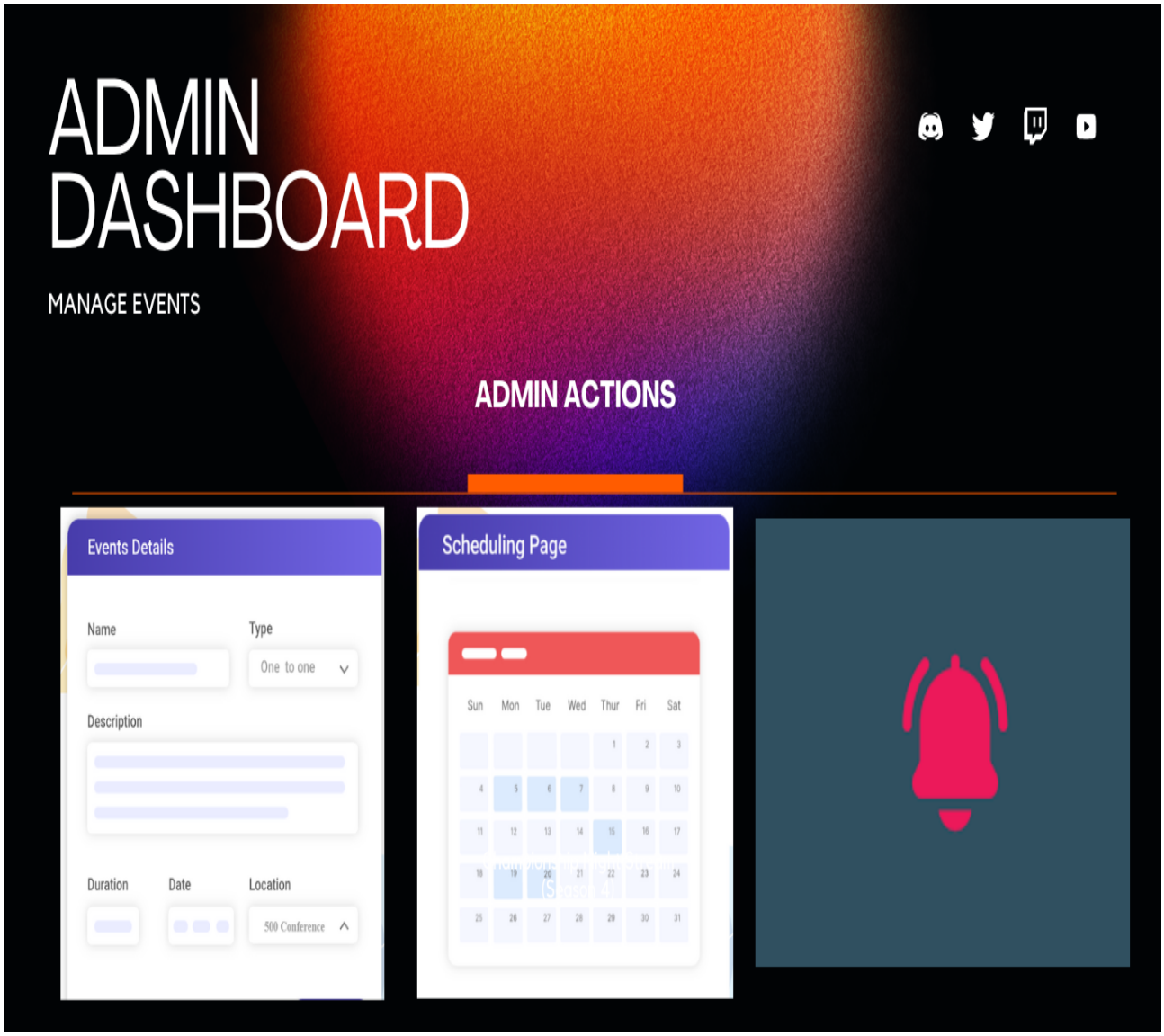


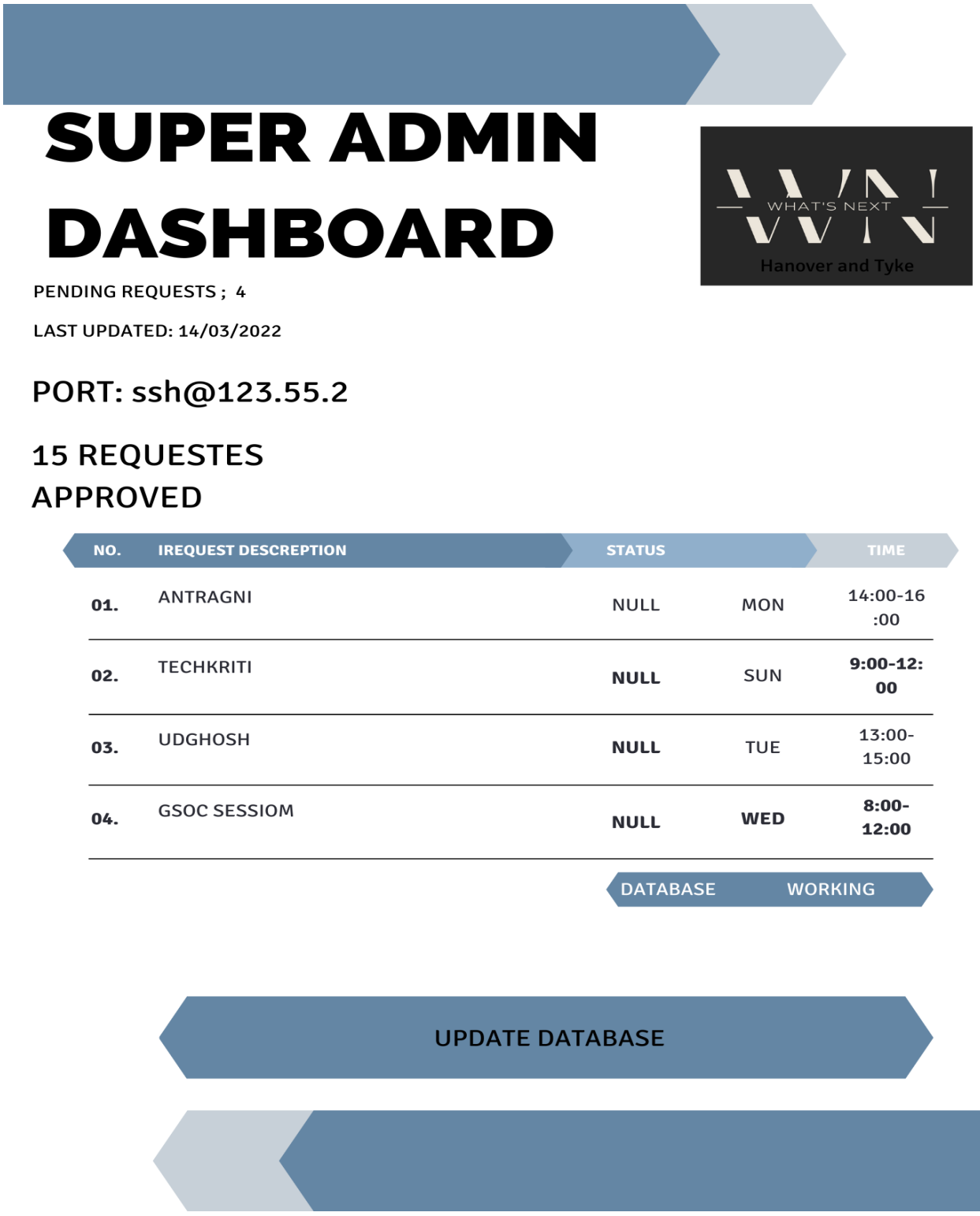
5. This is the view of FILTER functionality of our website

6. This is the EVENT DETAILS page of our website



7. This is the admin dashboard of our software





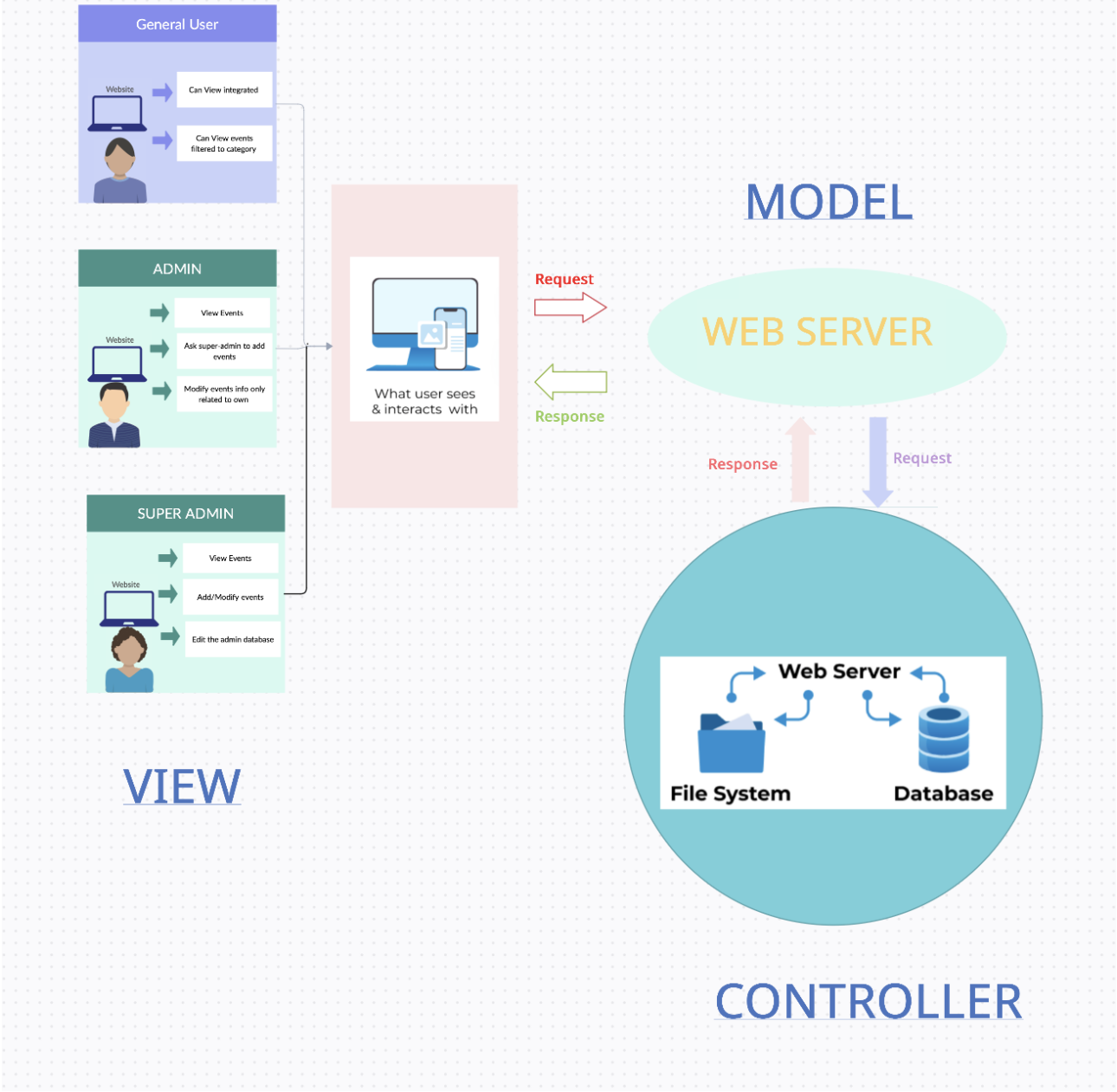
This is the SUPER ADMIN DASHBOARD

2 Architecture Design

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

The model manages the database. MongoDB is the database management system used. It receives information about query resolution and users' profiles from the controller and gives updates about queries, notifications and users' profiles to the controller. It also gets information of refreshed requests from the view. The view defines the display, user interface. HTML, CSS, Javascript are used for the view. It updates information about posted queries, announcements, profile management and status updations to the controller and gets the query notifications, status updations, and user and profile management from the controller. The controller receives updates from the view and sends information to the model. NodeJS is used for the controller. The view and the controller are directly connected to the browser.

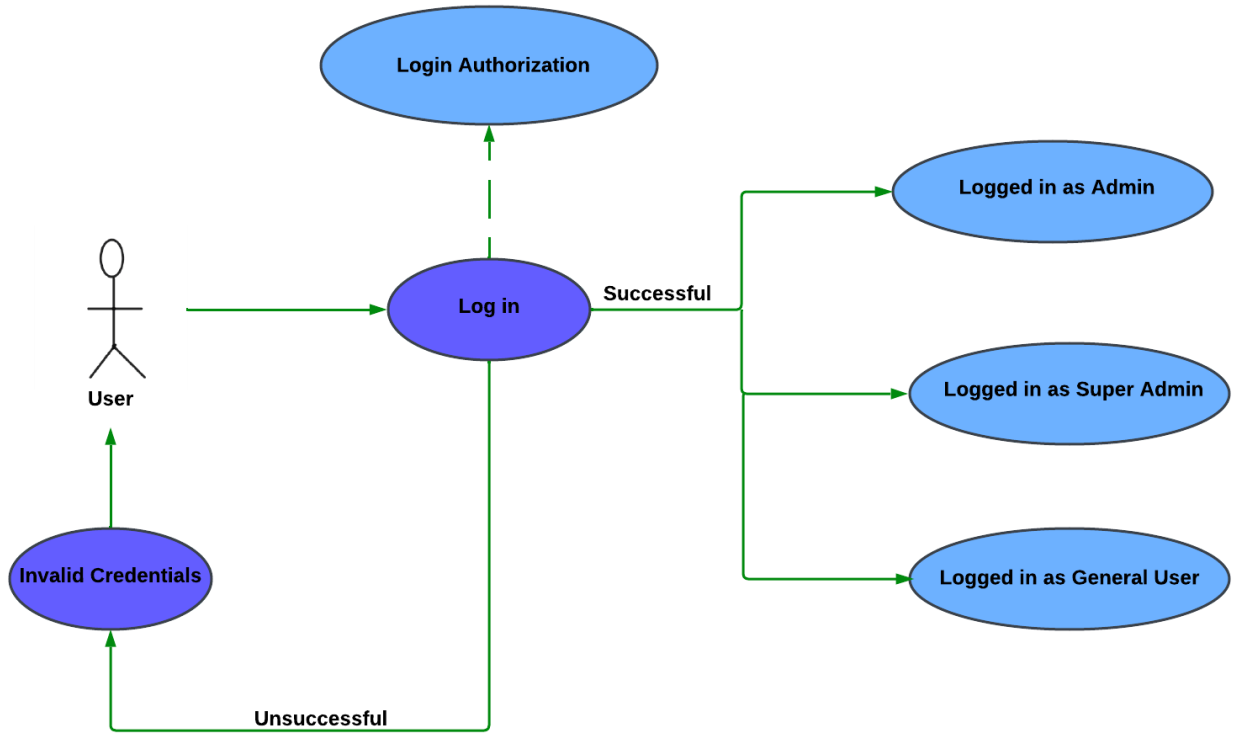
Thus, this architecture is the best fit for the project to connect frontend, backend and database from end to end for various functionalities as well it allows the data to change independently of its representation and vice versa. The only disadvantage is that it may involve additional code or code complexity where the data interaction is simple and thus may take more time in debugging.


General User
Website
Can View integrated
Can View events filtered to category
ADMIN
Website
View Events
Ask super-admin to add events
Modify events info only related to own
SUPER ADMIN
Website
View Events
Add/Modify events
Edit the admin database
VIEW
What user sees & interacts with
Request
Response
MODEL
WEB SERVER
Response
Request
Web Server
File System
Database
CONTROLLER

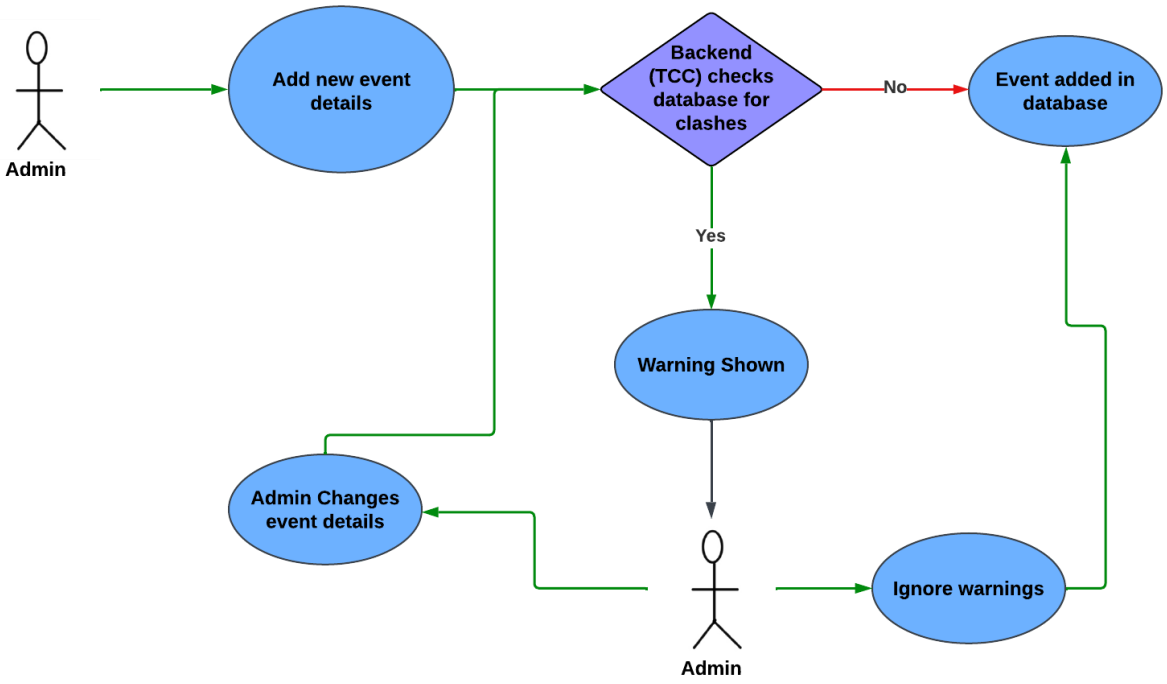
3 Object Oriented Design

3.1 Use Case Diagrams

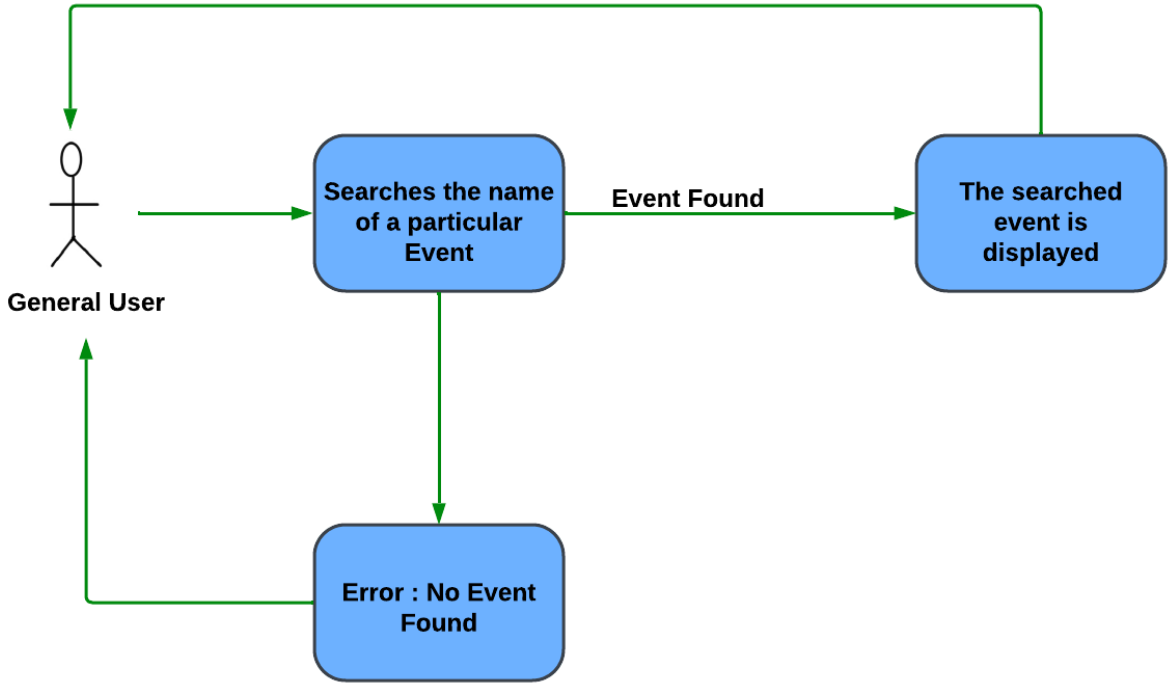
1. Login



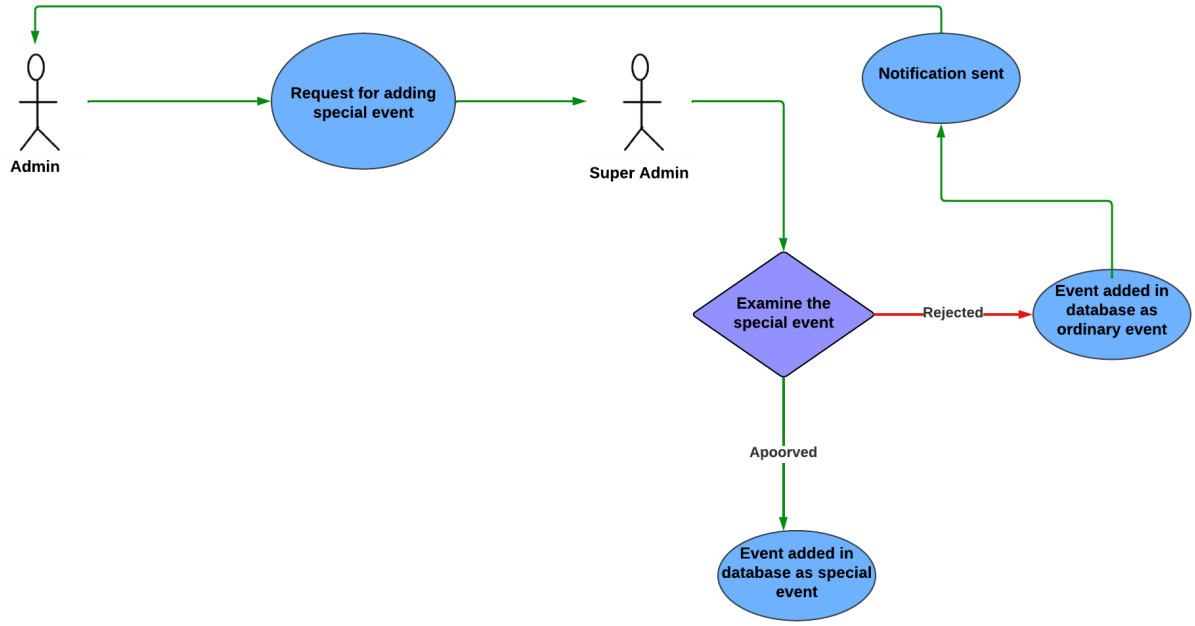
2.Add New Event



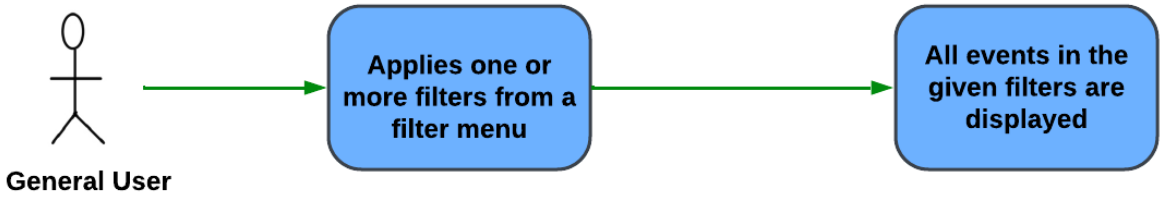
3.Searching for Event



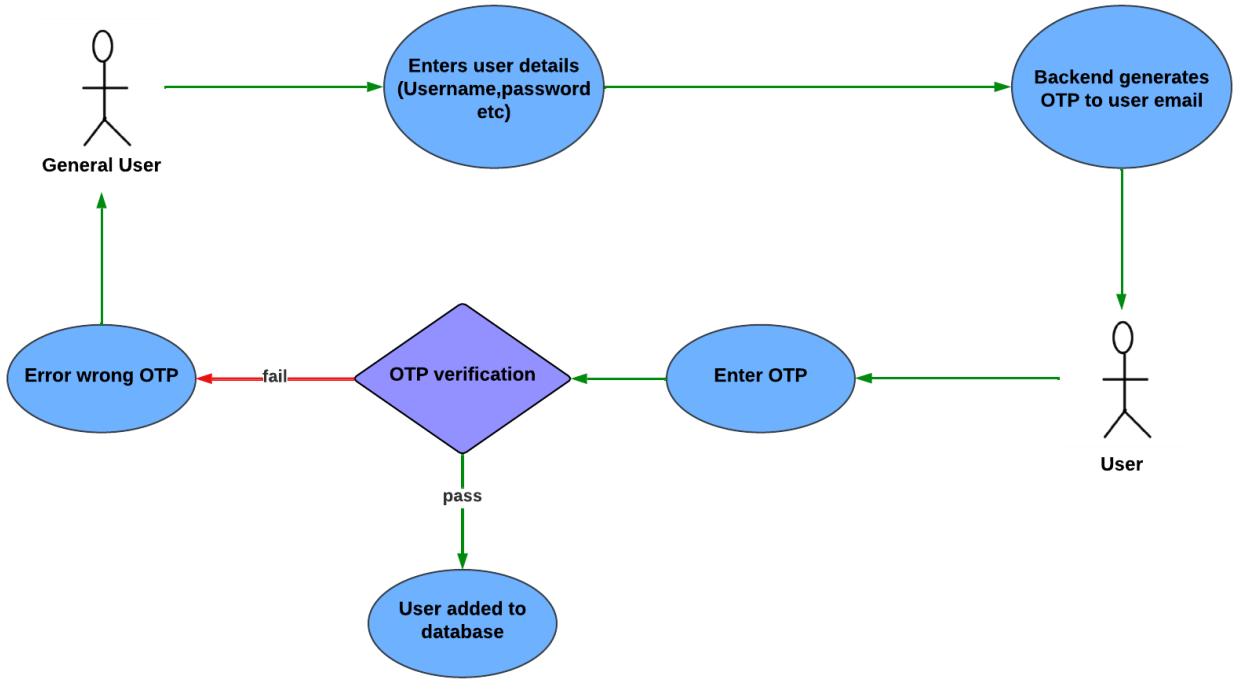
4.Adding a Special Event



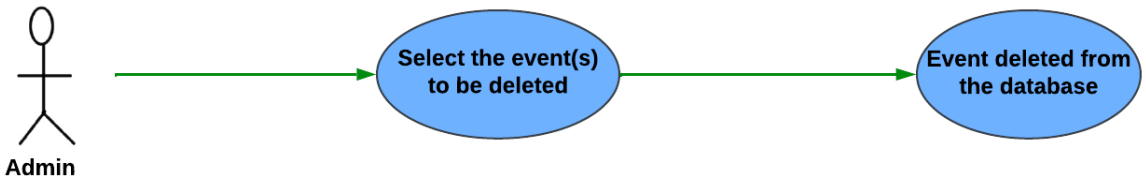
5.Applying Filters during Search



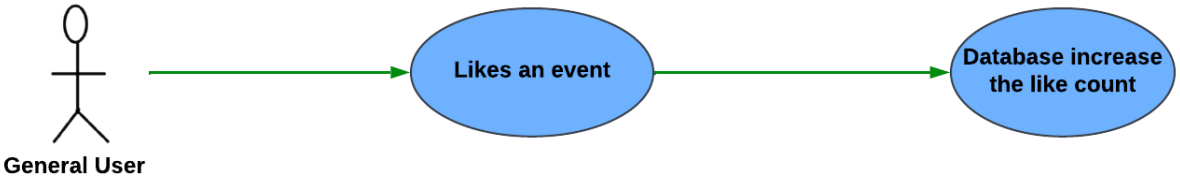
6.Sign Up



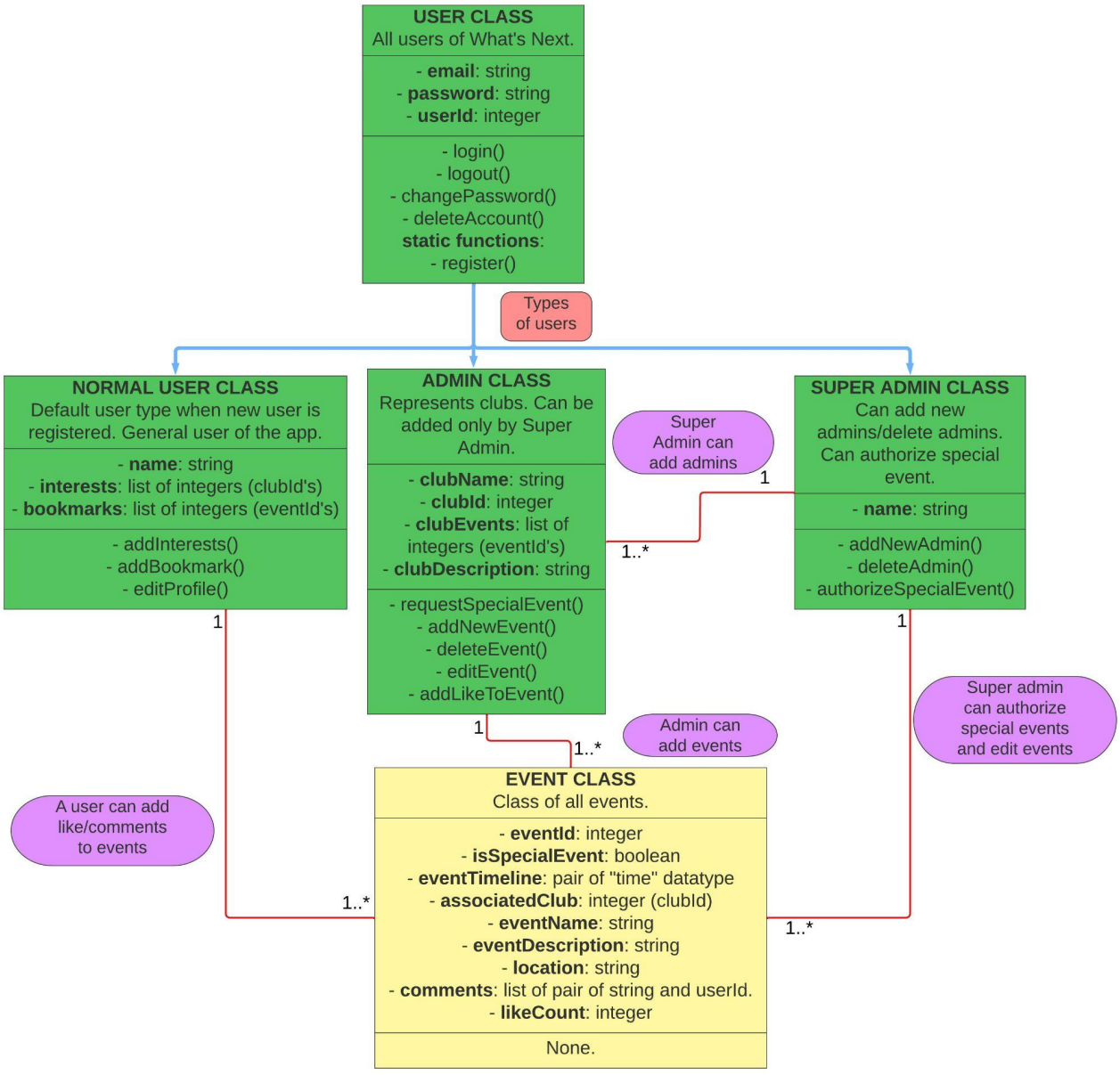
7.Deleting an Event



8.Liking an event

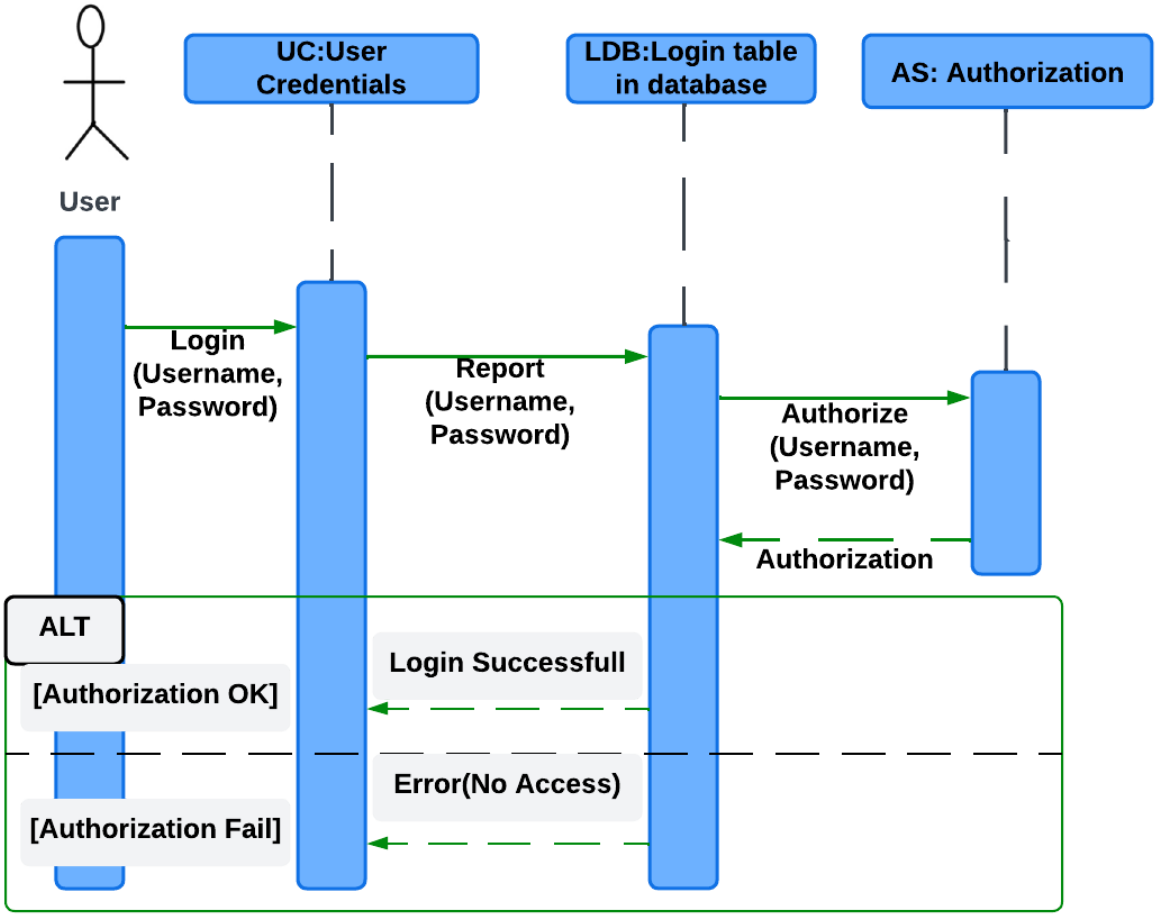


3.2 Class Diagrams

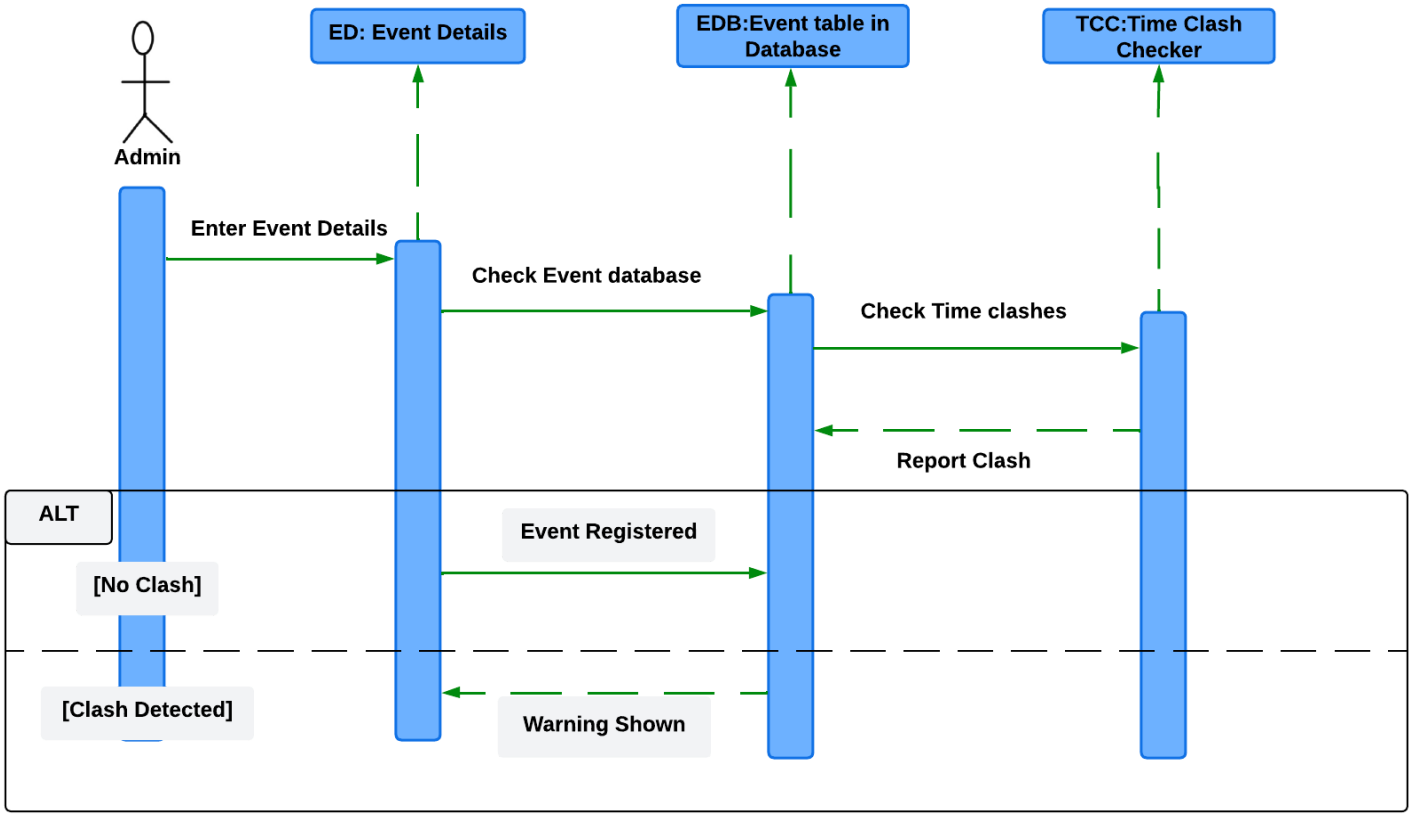


3.3 Sequence Diagrams

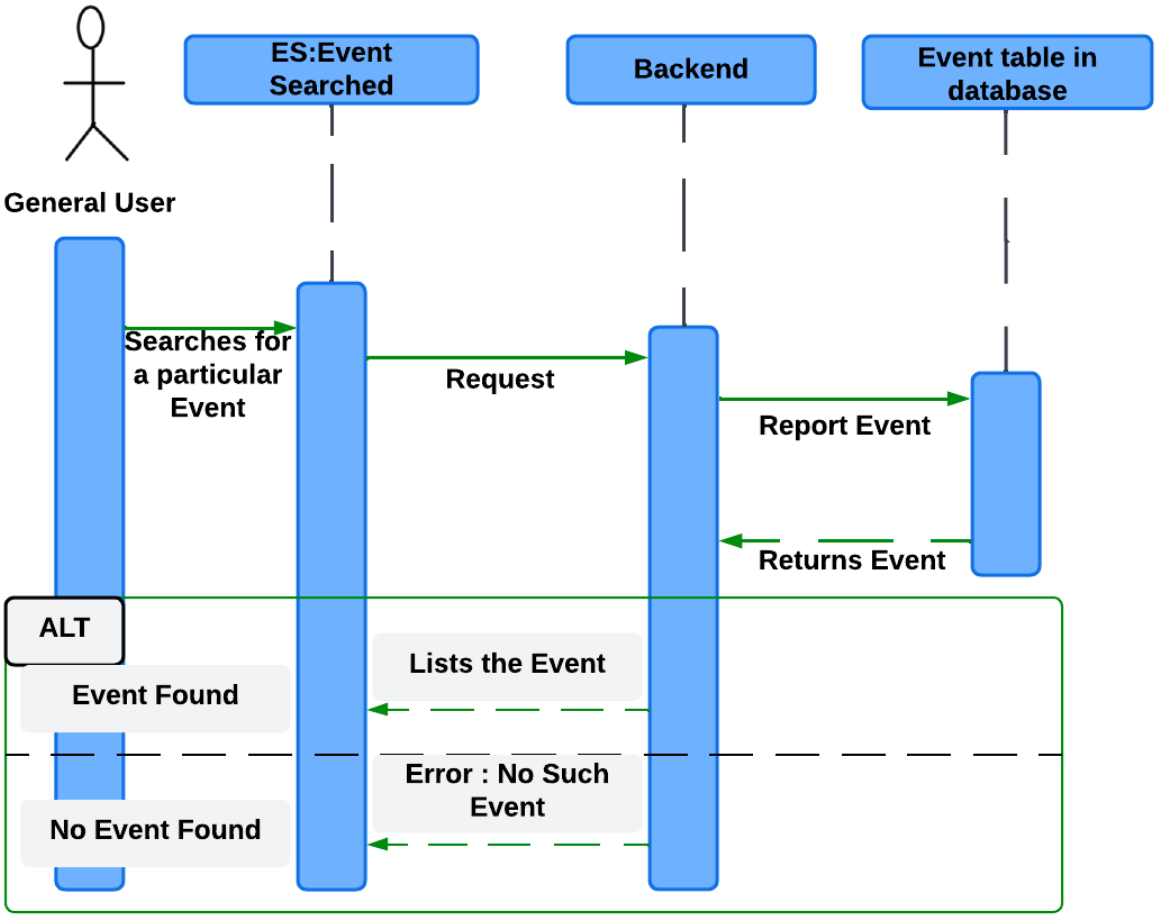
1.Login for User



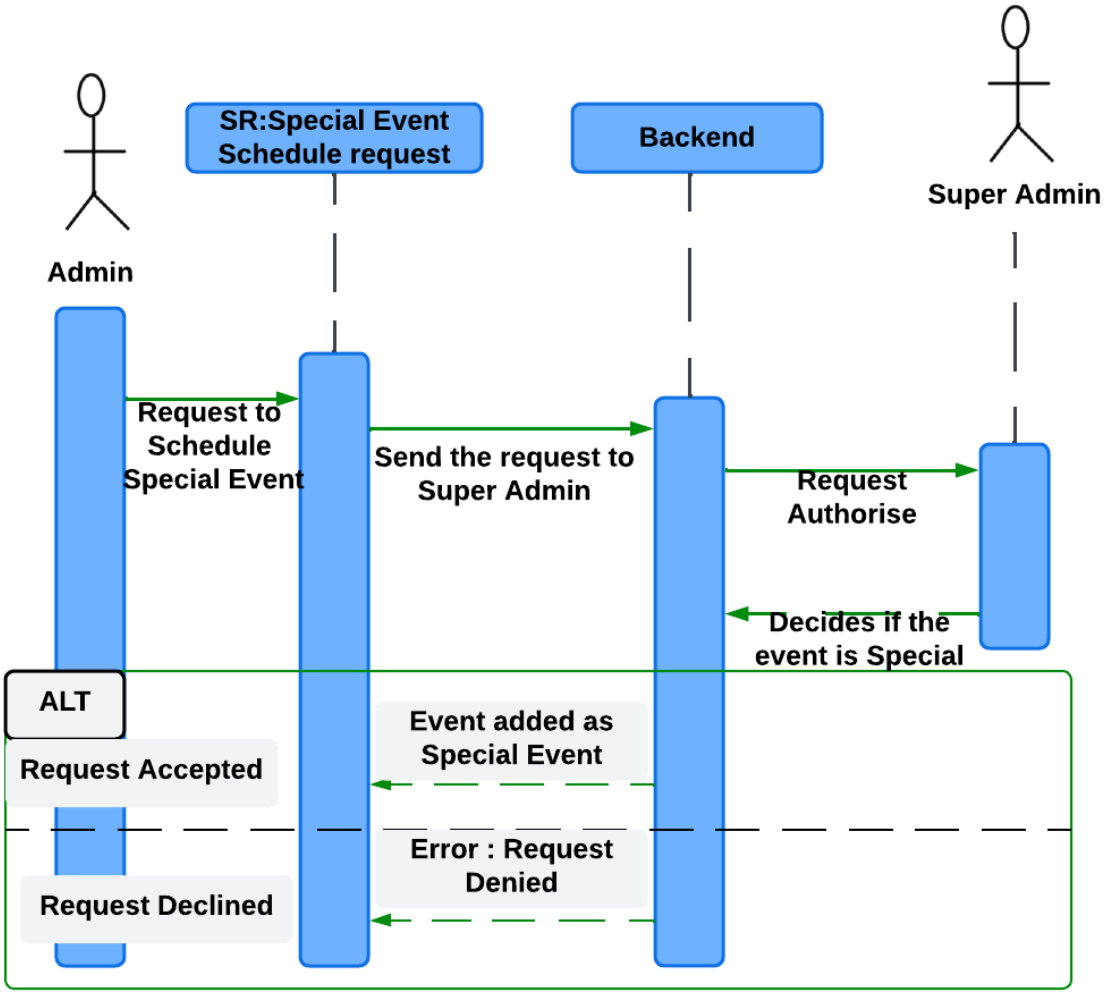
2.Register event



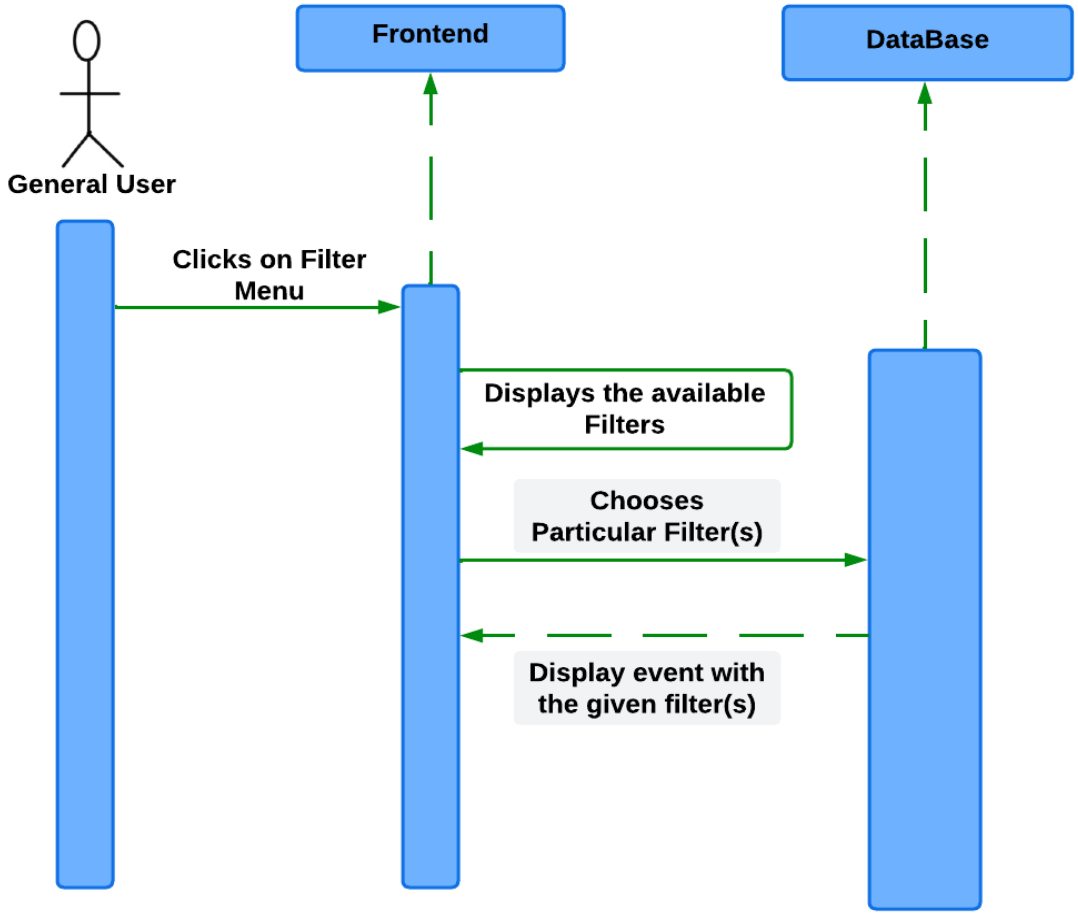
3.Search for an Event



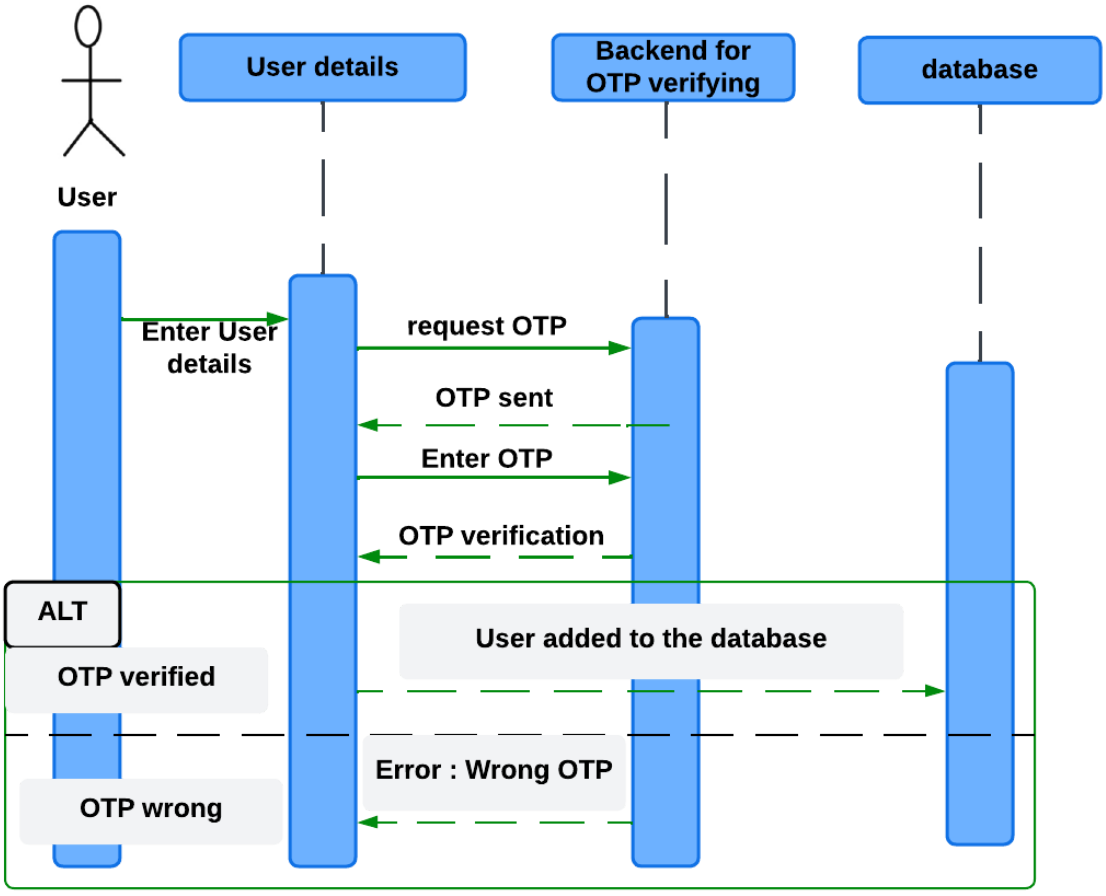
4.Adding Special Event



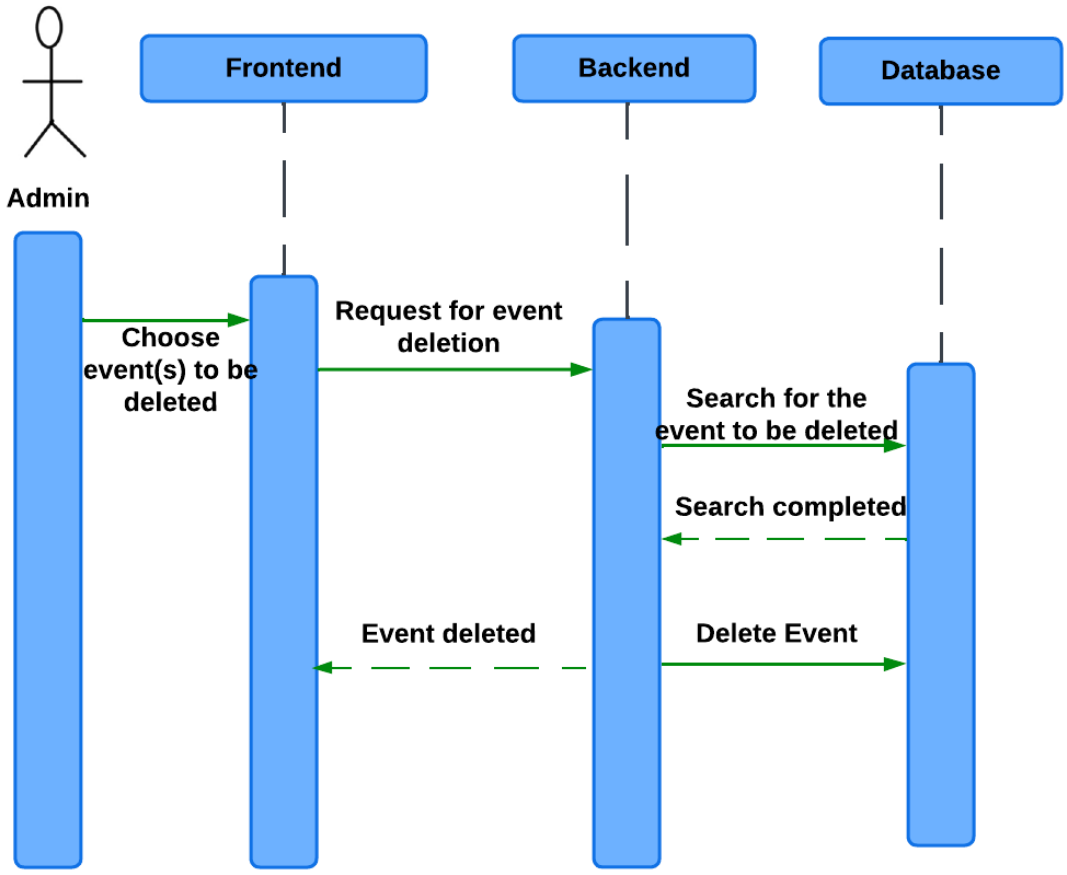
5. Applying Filters during Search


Frontend
DataBase
General User
Clicks on Filter Menu
Displays the available Filters
Chooses Particular Filter(s)
Display event with the given filter(s)

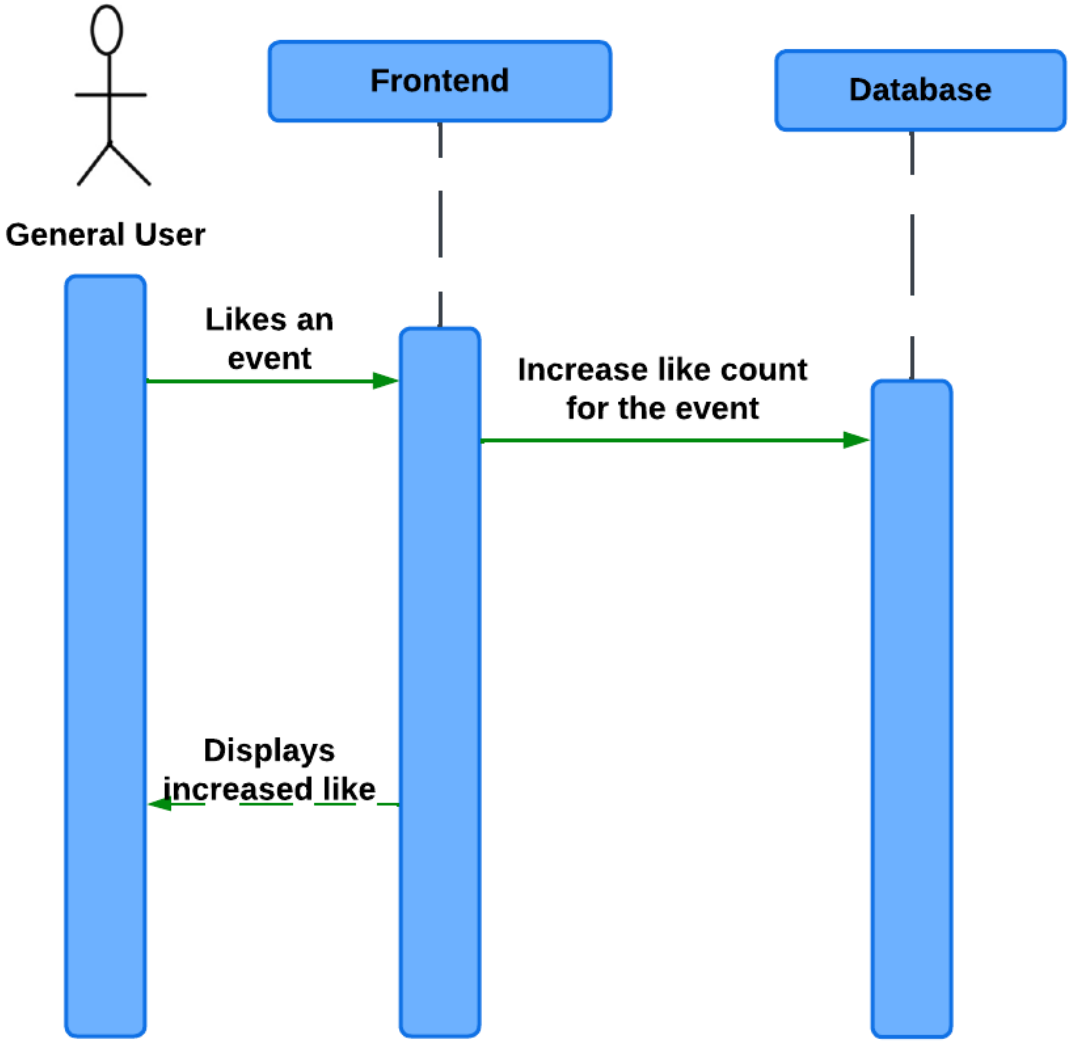
6.Sign Up


User
User details
Backend for OTP verifying
database
Enter User details
request OTP
OTP sent
Enter OTP
OTP verification
ALT
OTP verified
User added to the database
Error : Wrong OTP
OTP wrong

7.Deleting an event

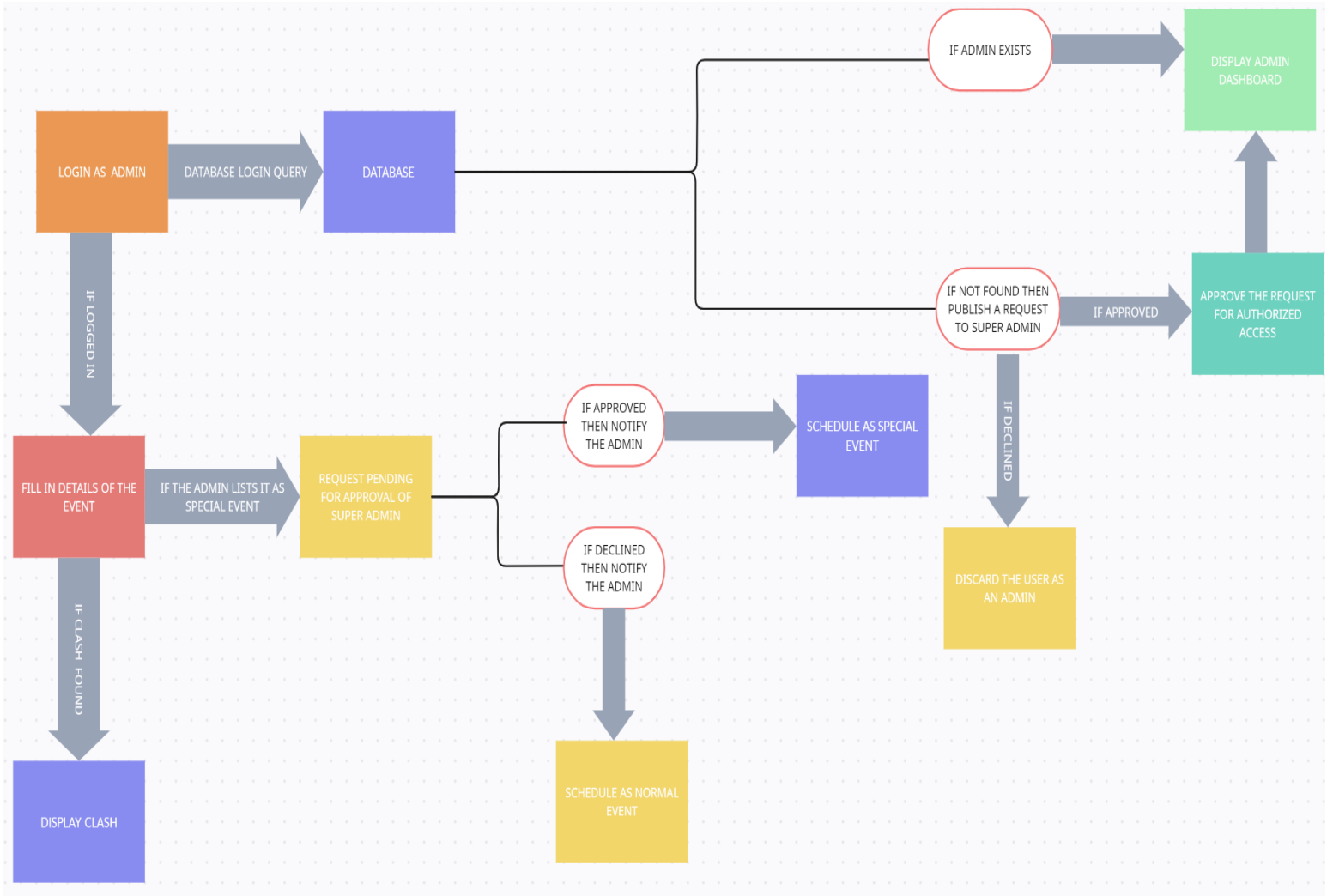


8.Liking an event

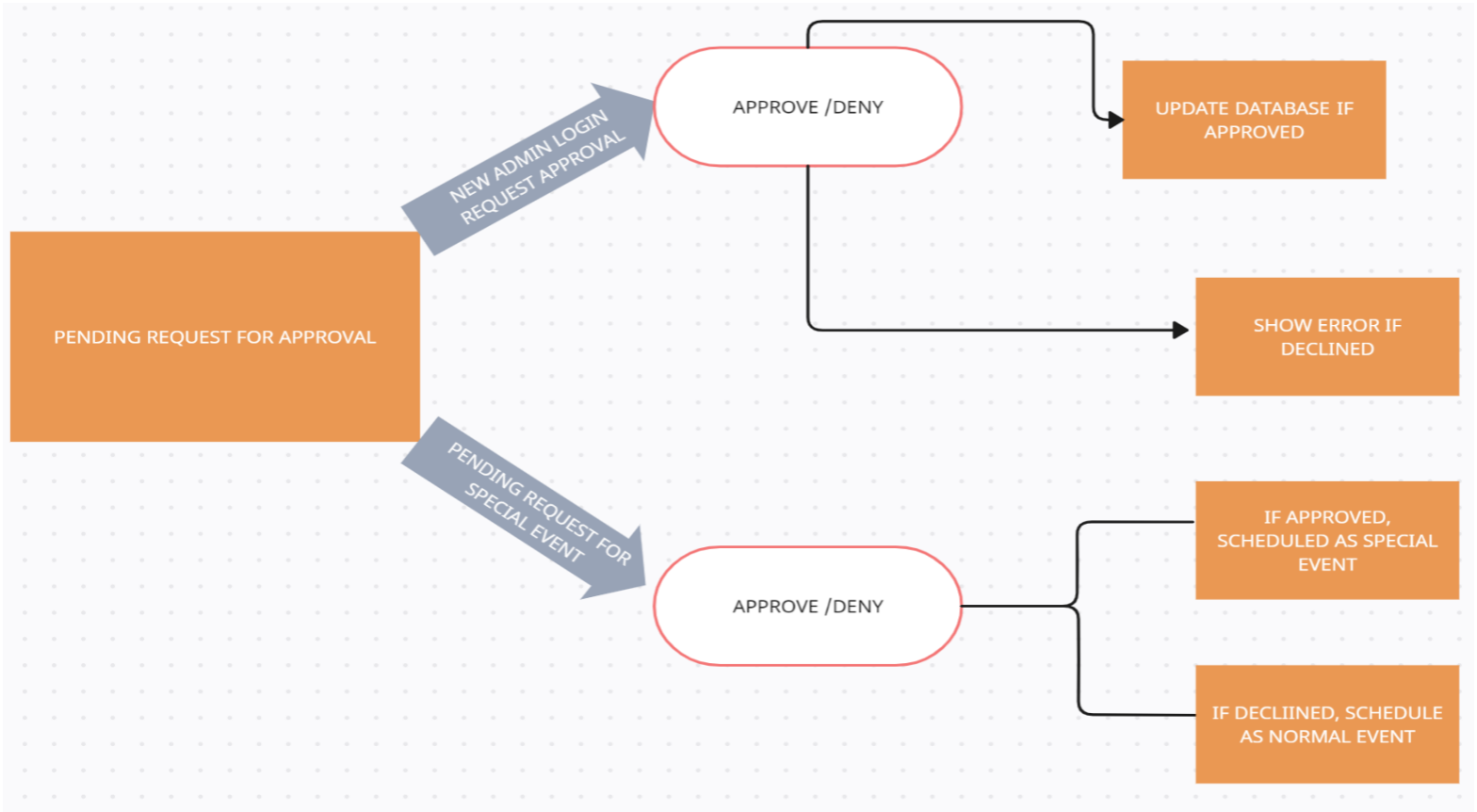


3.4 State Diagrams

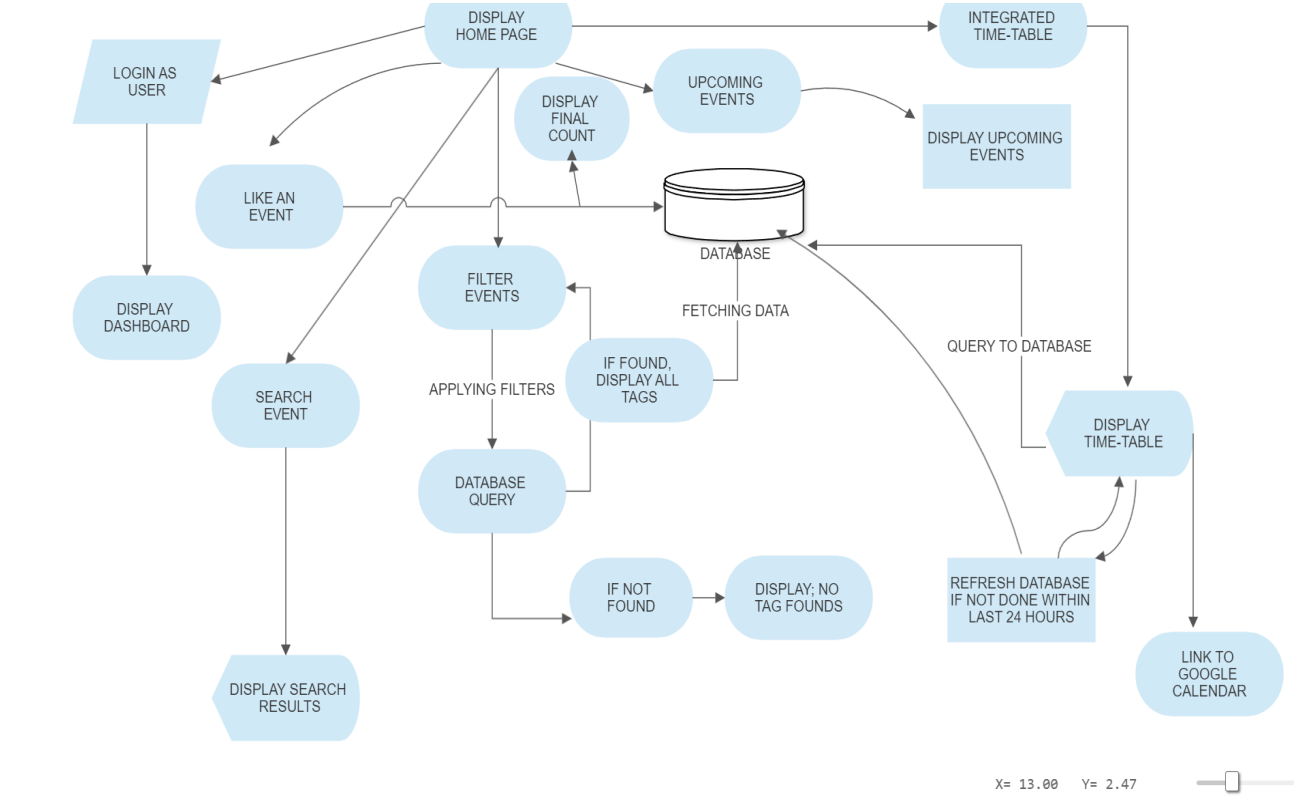
1. State Diagram for an ADMIN



2. State Diagram for SUPER ADMIN

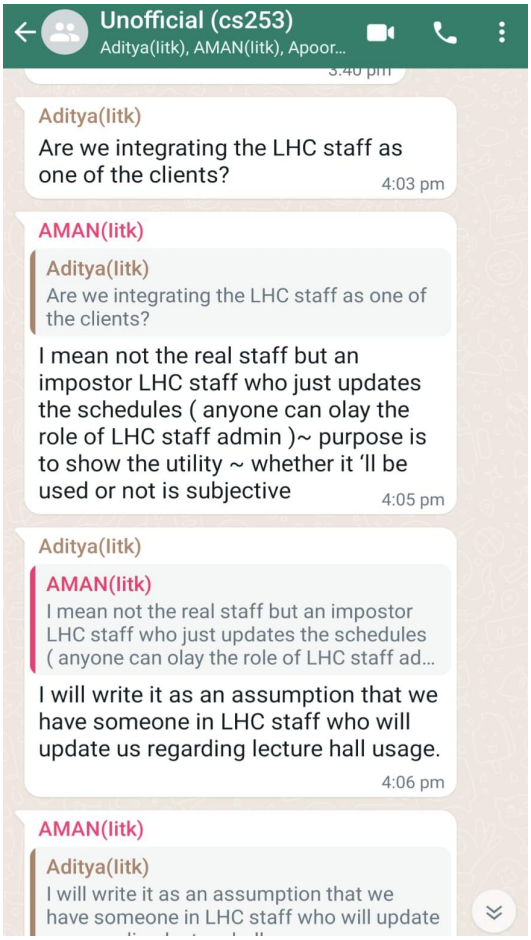


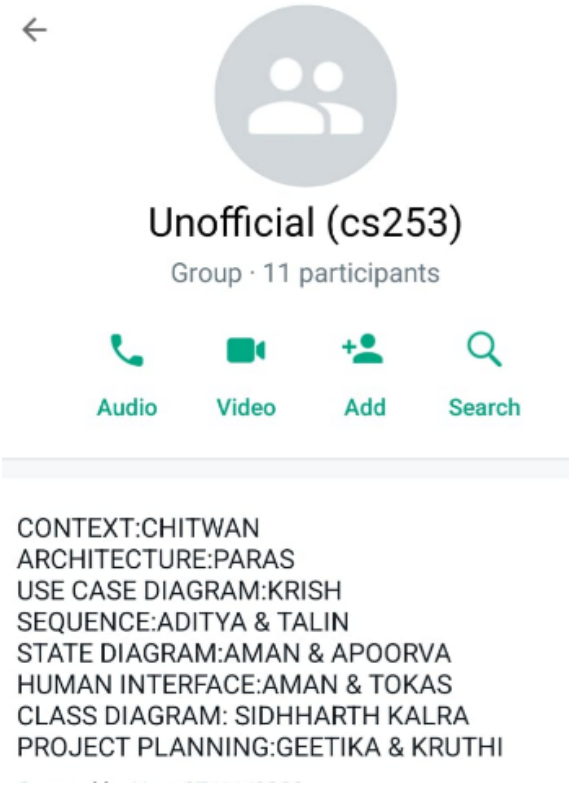
3. STATE DIAGRAM FOR GENERAL USER



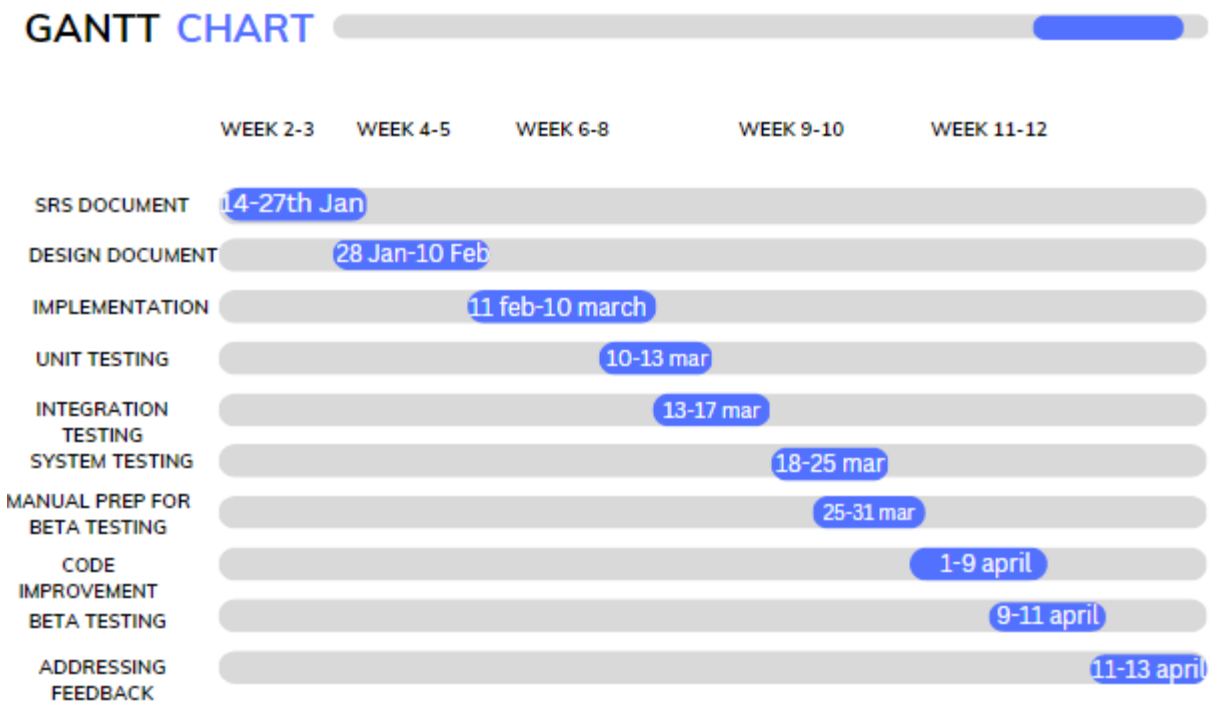
4.1 Communication:

- All the internal communication between the team members is via a whatsapp group.
- At Least one meet is held every week with all the team members to distribute the work in smaller groups(at the start), check the progress of each group and discuss every part to make sure every team member is on the same page.
- After every document's work distribution the description of the whatsapp group is changed accordingly with all the new distribution of work.


Unofficial (cs253)
Aditya(litk), AMAN(litk), Apoor...
Aditya(litk)
Are we integrating the LHC staff as one of the clients?
4:03 pm
AMAN(litk)
Aditya(litk)
Are we integrating the LHC staff as one of the clients?
I mean not the real staff but an impostor LHC staff who just updates the schedules (anyone can olay the role of LHC staff admin)~ purpose is to show the utility ~ whether it 'll be used or not is subjective
4:05 pm
Aditya(litk)
AMAN(litk)
I mean not the real staff but an impostor LHC staff who just updates the schedules (anyone can olay the role of LHC staff ad...
I will write it as an assumption that we have someone in LHC staff who will update us regarding lecture hall usage.
4:06 pm
AMAN(litk)
Aditya(litk)
I will write it as an assumption that we have someone in LHC staff who will update


Unofficial (cs253)
Group · 11 participants
Audio
Video
Add
Search
CONTEXT:CHITWAN
ARCHITECTURE:PARAS
USE CASE DIAGRAM:KRISH
SEQUENCE:ADITYA & TALIN
STATE DIAGRAM:AMAN & APOORVA
HUMAN INTERFACE:AMAN & TOKAS
CLASS DIAGRAM: SIDHHARTH KALRA
PROJECT PLANNING:GEETIKA & KRUTHI

4.2 Timeline:



4.3 Project Planning:

Person	**Work Assigned**
Chitwan Goel	Backend - Login page, Sign up, Events
Siddharth Kalra	Frontend - Dashboard(user); Frontend/backend integration testing
Varun Tokas	Backend - OTP generation and Email verification
Paras Sikarwar	Frontend - Login/Sign up page; Setting up the Database and Generating Queries
Geetika	Backend - Login page, Sign up, Events
Aditya Kumar	Frontend - Dashboard(admin)
Talin Gupta	Backend - OTP generation and Email verification
Apoorva Gupta	Backend - Filters; OTP generation and Distribution testing
Kruthi Akkinepally	Frontend - Login/Sign up page; Frontend/Backend integration testing
Aman Arya	Backend - Filters, Search bar; Authentication testing
Krish Sharma	Frontend - Landing page

The priorities for the proper functioning of a website include:

- Handling of data in the database with simultaneous access and quick searching while maintaining security
- Proper access control based on individual roles
- Avoiding memory leaks and overflows while protecting sensitive data
- Protecting user credentials and handling user data appropriately
- Ensuring compatibility with different devices, operating systems, and browsers
- Sustaining peak load times and handling high traffic
- Ensuring availability with minimum downtime for maintenance and updates
- Providing accessible design for users with disabilities and preferences
- Minimizing application crashes and ensuring quick recovery.

Functional testing: It is a type of software testing that verifies if the application meets the business requirements. It includes several testing methods such as:

- Unit testing: This tests each component of the code at the unit level to ensure that they work as expected.
- Integration testing: This tests the integrated units or components to ensure they work together as expected.
- System testing: This tests the complete and integrated system to ensure it meets the specified requirements.
- Acceptance testing: This tests the final product both internally and externally to determine if it is ready for delivery. This includes beta testing to gather feedback from end-users.

Non-functional testing: It focuses on the operational aspects of a software application. It includes the following testing methods:

- Performance testing: This tests the speed, scalability, responsiveness, and stability of the application under various conditions.
- Security testing: This tests the security of the data being used and stored in the software, and ensures that information is protected from unauthorized access or information loss.
- Usability testing: This tests the ease of use of the software from an end-user's perspective.
- Compatibility testing: This tests the compatibility of the software with different operating systems, platforms, browsers, and resolution configurations to ensure consistent functionality.

Appendix A - Group Log

Date	Timings	Duration	Minutes
31/1/23	6-7 pm	1 hr	● Work distribution
4/2/23	5-8 pm	3 hr	● Reviewing preliminary work done by each member. ● Discussion on improvements.
6/2/23	10-11:30 pm	1:30 hr	● Review of SDS by mentor. ● Clearing of our doubts regarding the document. ● Additional suggestions by mentor for further improvements.
7/2/23	6-8 pm	2 hr	● Discussion on project planning. ● Discussion on implementation of app. ● Work distribution of project implementation.
10/2/23	7:30-11:30 pm	4 hr	● Overall proof reading of the document. ● Work on UI design of website. ● Minor corrections were made.