

# **ClubHouse**

## **SYSTEM DESIGN DOCUMENT**

**Dhruv Patel**

**Priyank Dave**

**Amy Li**

**Arailym Mussilim**

**Noah Cristino**

**Tharuth Attanayake**

**Faraz Kaleem Malik**

# Contents

## Front-End

Club Admin Edit Profile Component	3
Club Admin View Profile Component	3
Club Admin Profile Page	3
Event Form (Club View)	4
Student Event Card	4
Register Form	5
Login	5
Navbar	5
AllClubs (Student View)	6
ClubsApplyButton	6
New Announcement	7
Club Admin Main Page	7
Positions	7
PositionsCard	8
PositionsCreate	8
PositionsForm	8
Club Register Request Form	9
MyClubs	9
MyClubsCard	10
MyAccounts	10
MyEvents	10
AuthCheck	11

## Back-End

User	12
Student	12
Club	12
Event	13
Tags	13
userDAO	13
adminProfileDAO	14
emailWrapper	14
membershipDAO	14
cookieDAO	15
clubMainDAO	15

<b>Software Architecture</b>	<b>16</b>
------------------------------	-----------



# Front-end

<b>React Component(s):</b> Club Admin Edit Profile Component	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Allow club admin to edit the profile page including, contact info, description of club, and profile picture.</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>• AuthCheck</li></ul>

<b>React Component(s):</b> Club Admin View Profile Component	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Displays the current profile page of the club associated with the club admin</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>• AuthCheck</li></ul>

<b>React Component(s):</b> Club Admin Profile Page	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Allow the Club Admin to go into edit mode or remain in view profile mode.</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>• Club Admin View Profile Component</li><li>• Club Admin Edit Profile Component</li><li>• AuthCheck</li></ul>

<b>React Component(s):</b> Event Form (Club View)	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Knows the logged in club's name</li> <li>• Knows the event name input</li> <li>• Knows the event image input</li> <li>• Knows the event start time input</li> <li>• Knows the event end time input</li> <li>• Knows the event location input</li> <li>• Knows the event description input</li> <li>• Knows the event's attendees</li> <li>• Knows the event tags input</li> <li>• When submit button is clicked, a post request gets submitted using api endpoint in events router, and then redirect to club profile</li> <li>• When delete button is clicked, redirect to previous page</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> StudentEventCard	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Displays eventName, organizer's name, start and end times, location, tags, more info in collapse card component</li> <li>• Reformats date, start and end times to display in "mmmm dS, yyyy" and "HH:mm" (shortTime), respectively</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> RegisterForm	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Lets user input username and password</li> <li>• Shows verification input field when required</li> <li>• Allows user to input verification code</li> <li>• Once verified, redirects user to the login screen</li> <li>• Perform basic verification on registration details</li> <li>• If any error, show them on corresponding fields at tiny subtext</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• Login</li> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> Login	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Allows the user to input an email and password and login</li> <li>• Logs user in and sets cookie if credentials match database</li> <li>• Knows the current state of the register form</li> <li>• Redirects user to home page if already logged in</li> <li>• Displays error message if password is incorrect or if the user doesn't exist</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> NavBar	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Sets navbar components</li> <li>• Changes color when clicked on different components</li> </ul>	<b>Collaborators:</b>

<b>React Component(s):</b> AllClubs (Student View)	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Provides megaview for club</li> <li>• Check database for clubs</li> <li>• Display club information from above</li> <li>• Consists of: <ul style="list-style-type: none"> <li>○ Club Banner</li> <li>○ Club Name</li> <li>○ Club Phone Number</li> <li>○ Club Email</li> <li>○ And a View Club button</li> </ul> </li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• ClubsApplyButton</li> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> ClubsApplyButton	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Access the applyMember endpoint onclick</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AllClubs</li> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> NewAnnouncement	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Knows the announcement subject</li> <li>• Knows the announcement message</li> <li>• Knows the recipients</li> <li>• When send button is pressed, a post request is made using api endpoint in announcements route, and then user is redirected to home page</li> <li>• When cancel button is pressed, clear form and redirect to home page</li> </ul>	<b>Collaborators:</b>

<b>React Component(s):</b> Club Admin Main Page	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Allow the Club Admin to view and scroll existing members of the club</li> <li>• Allow the Club Admin to accept and deny users that have applied to join the club</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> Positions	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Provides megaview for all job postings</li> <li>• Check database for positions</li> <li>• Display posting information from above</li> <li>• Consists of: <ul style="list-style-type: none"> <li>○ Job Position</li> <li>○ Club Name</li> <li>○ Club Email</li> <li>○ And a View Club button</li> </ul> </li> </ul>	<b>Collaborators:</b> AuthCheck



<b>React Component(s):</b> PositionsCard	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Format for the Position map in “Positions” page</li> <li>• Includes minimal pseudo-styling</li> </ul>	<b>Collaborators:</b>

<b>React Component(s):</b> PositionCreate	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Provides tab to create a new job listing</li> <li>• Inserts new position into database</li> <li>• Consists of: <ul style="list-style-type: none"> <li>○ Job Position</li> <li>○ Club Name</li> <li>○ Club Email</li> <li>○ Job Description</li> <li>○ Job Requirements</li> <li>○ Club Image uploader</li> </ul> </li> </ul>	<b>Collaborators:</b> AuthCheck

<b>React Component(s):</b> PositionForm	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Format for the Position form in “PositionCreate” page</li> <li>• Includes styling</li> </ul>	<b>Collaborators:</b> AuthCheck

<b>React Component(s):</b> Club Register Request Form	
<b>Parent Class (if any):</b> N/A <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Knows the club name</li> <li>• Knows the club email</li> <li>• Knows the club phone number (if entered)</li> <li>• Knows the club tags</li> <li>• Knows the club description</li> <li>• If the submit button is clicked, the entered data is validated, written to the database, then redirected to a confirmation page</li> <li>• If input is invalid, an error message is displayed</li> <li>• If the cancel button is clicked, the user is redirected to the home page</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> MyClubs	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Provides minimal view of all clubs that user is part of</li> <li>• Check database for registered club members</li> <li>• Display posting information from above consisting of the club name and a hyperlink to the club</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• AuthCheck</li> </ul>

<b>React Component(s):</b> MyClubsCard	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b> N/A	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Format for the Clubmap in “MyClubs” page</li> <li>• Includes minimal styling</li> </ul>	<b>Collaborators:</b>

<b>Class Name:</b> My Accounts	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• showing user events</li> <li>• showing user clubs</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• MyEvents</li> <li>• MyClubs</li> </ul>

<b>Class Name:</b> My Events	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• showing user events</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• EventCard</li> <li>• AuthCheck</li> </ul>

<b>Class Name:</b> AuthCheck	
<b>Parent Class (if any):</b> React.Component <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Checks if user is allowed to access a page</li> <li>• Redirects to a given url if not allowed</li> </ul>	<b>Collaborators:</b>

# Back-end

<b>Class Name:</b> User	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Knows its email and password</li><li>• Know what type of account credentials are attached to</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>•</li></ul>

<b>Class Name:</b> Student	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Knows its real name</li><li>• Knows associated User Object</li><li>• Knows all ids of clubs she/he a general member of</li><li>• Knows all ids of clubs he/she follows</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>• User</li></ul>

<b>Class Name:</b> Club	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Knows its name</li><li>• Keeps track of its general members</li><li>• Knows its categories/tags</li><li>• Keeps track of</li></ul>	<b>Collaborators:</b> <ul style="list-style-type: none"><li>•</li></ul>

<b>Class Name:</b> Event	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Knows its organizer's name</li> <li>• Knows its name</li> <li>• Knows its image</li> <li>• Knows its start time</li> <li>• Knows its end time</li> <li>• Knows its location</li> <li>• Knows its description</li> <li>• Knows its attendees</li> <li>• Knows its tags/categories</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>•</li> </ul>

<b>Class Name:</b> Tags	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Keeps track of event and club tags</li> </ul>	<b>Collaborators:</b> <ul style="list-style-type: none"> <li>•</li> </ul>

<b>Class Name:</b> usersDAO	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Performing database operations on users and potentialUsers</li> <li>• inserting</li> <li>• finding</li> <li>• deleting</li> </ul>	<b>Collaborators:</b>

<b>Class Name:</b> adminProfileDAO	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Performing database operations on clubs and club specific events</li> <li>• Inserting image strings</li> <li>• Querying club profile information</li> <li>• Fetching event details specific to the club</li> </ul>	<b>Collaborators:</b>

<b>Class Name:</b> emailWrapper	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Sending a verification email</li> </ul>	<b>Collaborators:</b>

<b>Class Name:</b> membershipDAO	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Finding and adding club membership applications in the database</li> </ul>	<b>Collaborators:</b>

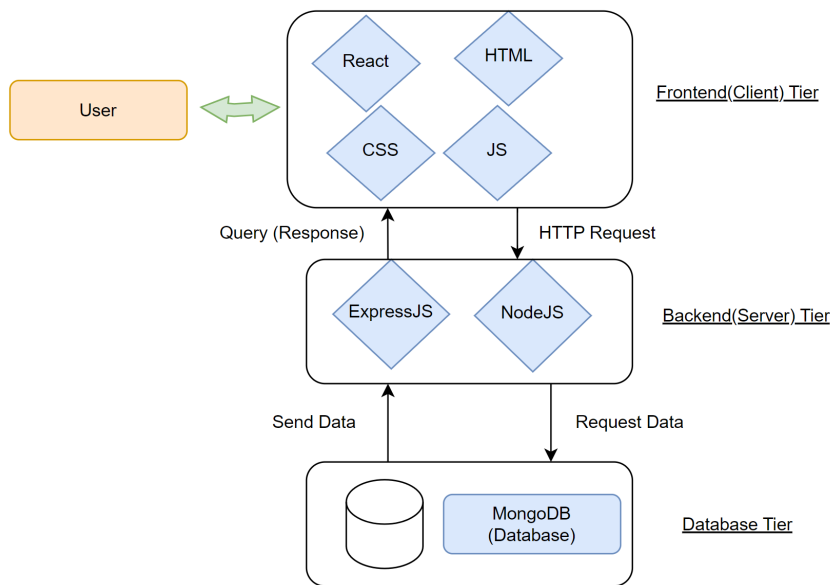
<b>Class Name:</b> cookieDAO	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Getting and setting cookies</li> </ul>	<b>Collaborators:</b>

<b>Class Name:</b> clubMainDAO	
<b>Parent Class (if any):</b> <b>Subclasses (if any):</b>	
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Performing database operations on club members and club applicants</li> <li>• Fetching club members and club applicants</li> <li>• Removing club applicants on deny</li> <li>• Inserting club members on accept</li> </ul>	<b>Collaborators:</b>



## Three Tier Architecture

This project utilized the Three Tier Architecture (MERN Stack). The user interacts with React (the client) frontend of the web application. Any user requests will enable React to send a HTTP request to the application server which is written using NodeJS and ExpressJS. The server then serves the client by accessing the MongoDB database.



## System Decomposition

The system architecture has three separate parts: the database, the frontend, and the backend. The frontend is what the users will be interacting with. The frontend is developed using React and its components. We also used Material UI for universal styling and creating smoother working components. We also call the API endpoints that the backend provides using the fetch call. This way we can allow for CRUD properties to be introduced in our application. The backend of the app was built using Node.js and Express.js. Express is a popular node framework and has a lot of benefits. One of the ones we used is for writing handlers for http requests at different URL paths (routes). This is how we set up the API endpoints that interact with our database (MongoDB). In our database, we have set up different collections for different objects such as clubs, events, and users. We have implemented input sanitation code to make sure we don't receive invalid inputs. And for invalid images, we have implemented pop-up alerts that show up on the user's screen.