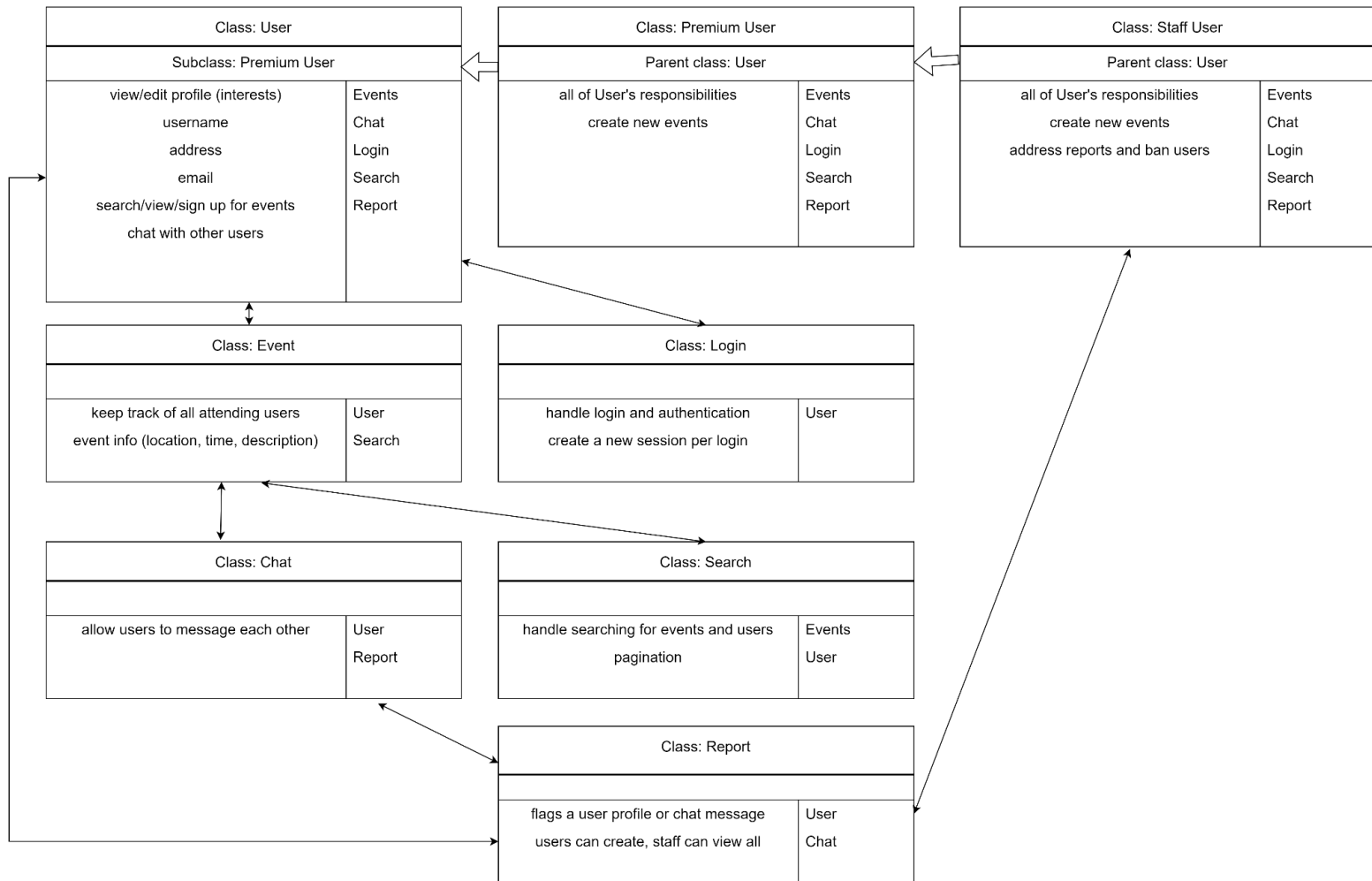


CRC cards from drawio for classes (high level description)



Description of system interaction (dependencies):

The app is built on the MEAN stack and requires having a MongoDB database set up and Express, Angular, and Node.js being installed in the operating environment that it will be running on.

Architecture of the system description (abstract) - components and connections

- Users will sign up and log in with their credentials which will be authenticated
- Users can interact and connect with events by being able to view and sign up for them
 - Premium users can create events for others to join
- Users can connect to other users that are interested in the same event by creating a chat
- Users can search and filter for events, or other users that are attending events
- Users can create reports tied to user profiles or chat messages, and staff users can view these reports and decide to act on them or not.
-

System decomposition

System architecture: high level components/technology used

- The pattern this application is based around is the three tier architecture, which is a common choice for web applications. The client (Presentation Tier) is represented by an Angular frontend, the Server (Logic tier) is powered with a combination of Node.js + Express, and the Database (Data tier) stores the data in a MongoDB database.

Class/object interactions relating the architecture description to the code interactions:

- Users log in through a login form which internally sends a POST request with the input credentials which are internally checked and compared to their hashes (with bcrypt) and if valid, allows the user to log in and creates a new temporary session for them so they will stay logged in until it expires. It will also be destroyed if the user logs out.
- The way users view events is through an event board that will load info with a GET request for each event, visually separated into cards which will show a preview of each event and allow users to click into the respective event page for further details.
- (Premium) Users can create events through an event creation form which they can provide the different parameters like name, description, and location for. The event will be created using a POST request.

Roles of components in the higher level view:

- Frontend: displaying the information on the site in a user-friendly way
 - CSS: aesthetics for styling the main content, navigation, login forms
 - The user can fill out the info in a form for register/signup/event creation that will be translated into the raw POST request internally
 - The event board is used to preview event information and event pages will be used to view more details for specific events..
 - The user profile shows the user information for the specific user.
- Backend: storing the actual info in the MongoDB database for user/events
 - The GET and POST requests for signup/login/viewing events/creating events will access the database and obtain/update the relevant data.
 - Users are represented internally in the database through a Mongoose schema that contains username, first name, last name, type (basic or premium), password, and a list of interests.
- Authentication: using bcrypt for login authentication (hashing passwords) in the user router

Anticipated errors and exceptions:

- Invalid input (viewing invalid user/event IDs): 404 not found error
- Network failure: the site will have an error message/won't load
- Incorrect login credentials: 401 Unauthorized
 - Also occurs if a basic user tries to create an event
- If the user's session has expired, they will be redirected to log in again.