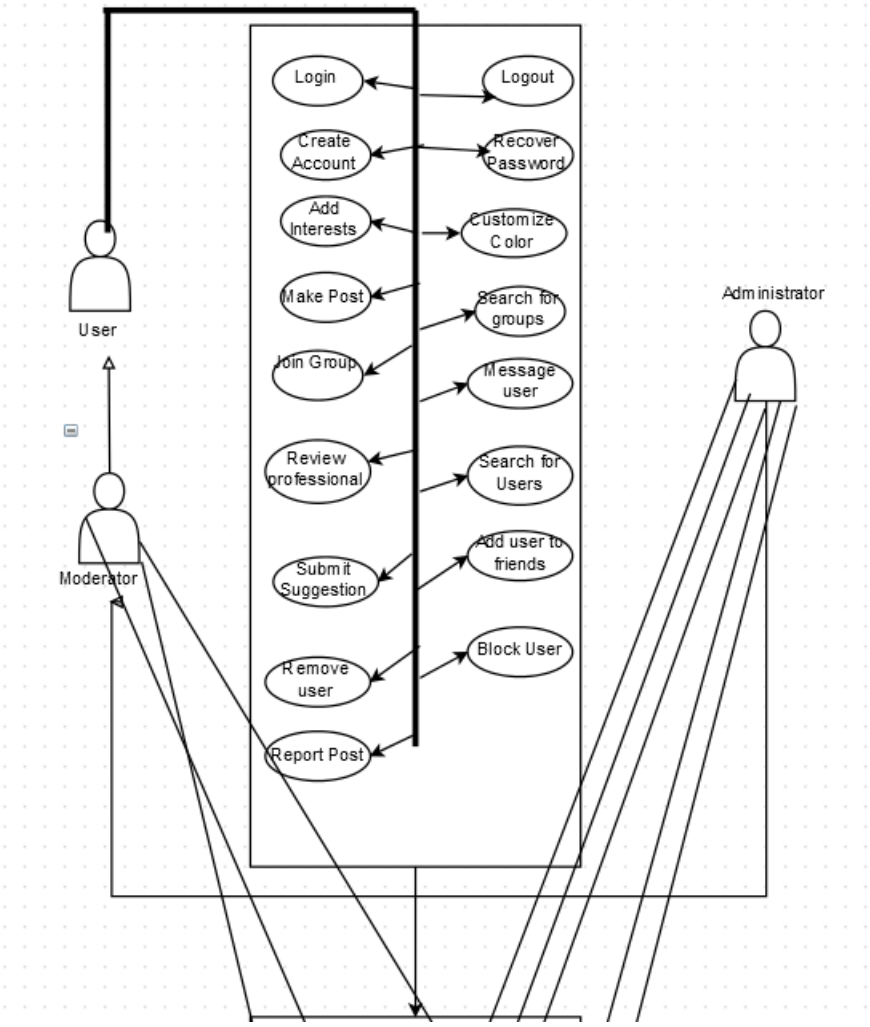
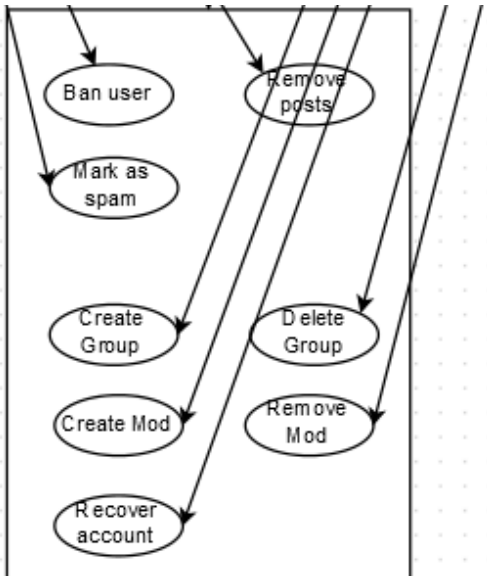

Login
Logout
Create Account
Recover Password
Add Interests
Customize Color
Make Post
Search for groups
Join Group
Message user
Review professional
Search for Users
Submit Suggestion
Add user to friends
Block User
Remove user
Report Post
User
Moderator
Administrator


Ban user
Remove posts
Mark as spam
Create Group
Delete Group
Create Mod
Remove Mod
Recover account

The Hue Manatees

Github Repository: https://github.com/beckerichm1/CSC440-GroupProject-Hue_Manatees.git

Requirements:

- 1. Authentication: Users need to be able to create accounts and must login to access user specific data, such as preferences, data sets, or custom visualizations. Different classes of users may exist, with access to their own sets of use cases.
- 2. User interface: A well-designed user interface has a good layout for displaying the data, user authentication, user manipulation on the data.
- 3. Data processing: On request of an authenticated user, the application must be able to retrieve data from the remote server, remove or repair inconsistent or erroneous data, and parse the data.
- 4. Storage: Once imported, data should persist until deleted by a user. Multiple data sets may be stored at once. Select an appropriate storage system for the type of data used: text file, binary file, SQL database, or NoSQL database.
- **5. Communication (public[forum] and private [Inbox]):**
- **6. Customization: different profile color schemes/panel color schemes?**
- **7. Security: Secure user information (personal)**
- **8. Configuration: Configure server to proper settings. Set up DB, php, etc.**

Vision:

Use Cases:

Actors: User, Administrator, Moderator

User:

- **UC21: Log into application**
- **UC22: Log out of application**
- **UC23: Create new account (sign up)**
- Manage personal/professional profile
    - **UC1: Add interests**
    - **UC2: Customize profile color scheme**
    - **UC3: Recover/change password**
- Interaction
    - PANELS (group pages)
        - **UC4: Join group**
        - **UC5: Make panel post (forum)**
    - **UC6: Message other users (IM(maybe)/Inbox)**
- **UC7: Search for users**
- **UC8: Search for groups**

- **UC9: Review professionals**
- **UC10: Submit new panel suggestions**
- Buddies (friends)[relations to other users]
    - **UC11: Add user as friend**
    - **UC12: Remove user from friends**
    - **UC13: Block other users**
- **UC25: Report post**

Administrator: (built on top of moderator?)

- **UC14: Create group panels**
- **UC24: Delete group panels**
- **UC15: Create moderator status**
- **UC16: Remove moderator status**
- **UC17: Recover account information**

Moderator: (built on top of normal user profile)
Reference: https://www.reddit.com/wiki/moderation1

- **UC18: Ban users**
- **UC19: Remove posts**
- **UC20: Mark as spam**

Matthew Beckerich

Use Case Name

User Log in

Scope

Primary Actor

User

Stakeholders and interests

User: wants to be in the their account with access to any associated features that come with the account

Admin: wants system to log any new accounts or failures to log in.

Preconditions

The software exists to allow a person to login.

Postconditions

The user will be in an account of their creation. They have a live connection to the website.

Main Success Scenario

1. The user connects to the site.
2. The user enters username and password
3. The system authenticates the account (hash and salt the passwords)
4. The user is directed to their customized page

Alternate Flows

2A. The user has no user name and password

1. Execute Create Account use case

2B. The user mis-enters the username

1. System checks and username does not exist
1. Resend user login page indefinitely

1A. System checks and credentials are wrong

1. System starts a count to limit the number of reattempts and resends
2. User reattempts
    a. Login credentials authorized, log user in

A2. Mistake increment counter, limit to 3 mistakes and try once more

Use Case UC#1: Add Interests
Scope: ??????
Level: user-goal
Primary Actor: User
Stakeholders and Interests:

- User: Wants to increase odds of finding an appropriate Panel to join.

Preconditions:

- User must have created and logged into an account.
- Preferred interests must already exist.

Success Guarantee (Postconditions):

- User successfully adds an interest to his or her profile.

Main Success Scenario (Basic Flow):

1. User browses a list of interests.
2. User can expand categories of interests to see more specific interests.
3. User can use a search bar to find interests.
4. The system can suggest similar interests.
5. User selects an interest and adds/joins it.

Extensions (Alternate Flows):

1. User browses a list of interests.
2. User can expand categories of interests to see more specific interests.
    a. User notices an interest that is incorrectly categorized.
    b. User notifies an administrator.
3. User can use a search bar to find interests.

. Search bar returns no results.
a. User can re-word search.
b. User can't find preferred interest.
c. User contacts an administrator or moderator.

4. The system can suggest similar interests.

. There are no similar interests to suggest.

5. User selects an interest and adds/joins it.

. The user fails to join.
a. An error message shows ("Could not add ___ as an interest").
b. User contacts an administrator.

Special Requirements:
Technology and Data Variations List:
Frequency of Occurrence: Can occur countless times so long as there exist interests a user has yet to join.
Miscellaneous: