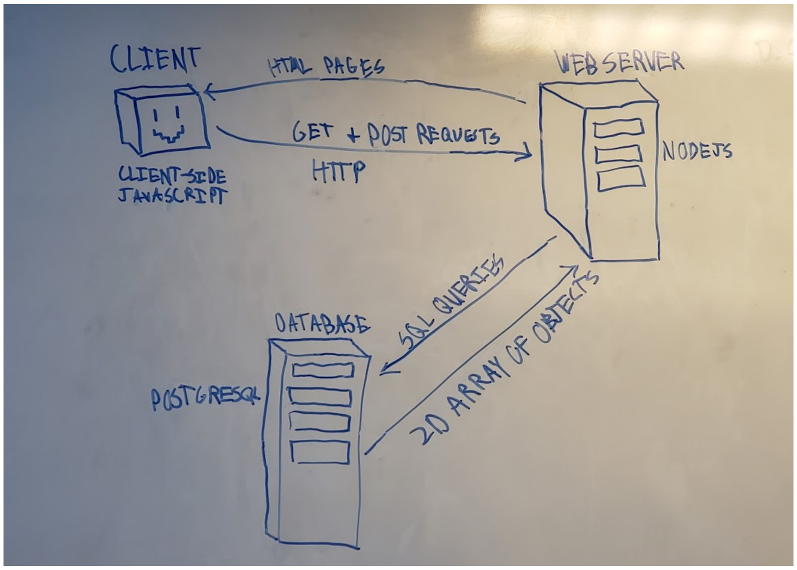
Group: 202-1
Team Name: Work in Progress
Team members: Evan Kuehr, Kaylee Engelhardt, Nick Price, Leon Shen, Gunther Wallach

Application name: CU Swap

List of features (Priority ranked from 1 = most important):

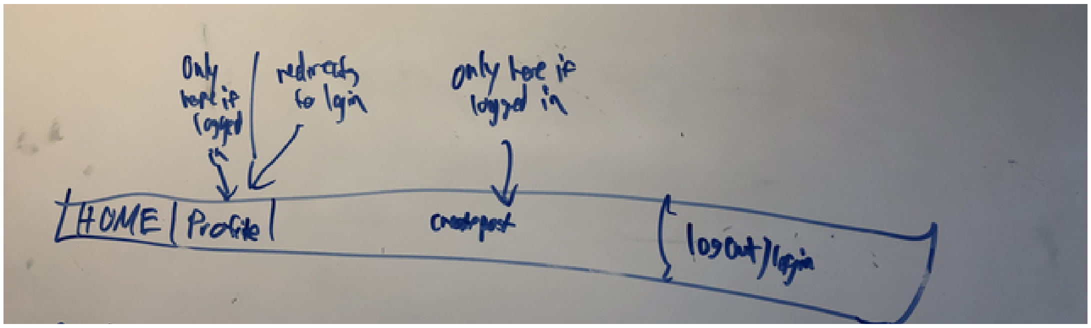
- Creating Posts (4)
  - Being able to add a new post to the database with tags, etc.
- Filter through posts (3)
  - View only certain posts based on selection of tags
- Login/giving user a session cookie (1)
  - Logging into the site and being assigned a cookie for your current session
- Register (7)
  - Creating an account
- Password hashing (6)
  - Hashing function which will store passwords into proper location
- Messaging System (5)
  - Keeping track of messages that users send to each other and making sure that the messages are only read by the right parties
- Viewing User profile and post history (2)
  - Loading all the posts a user has made and the messages they have received

Architecture diagram:

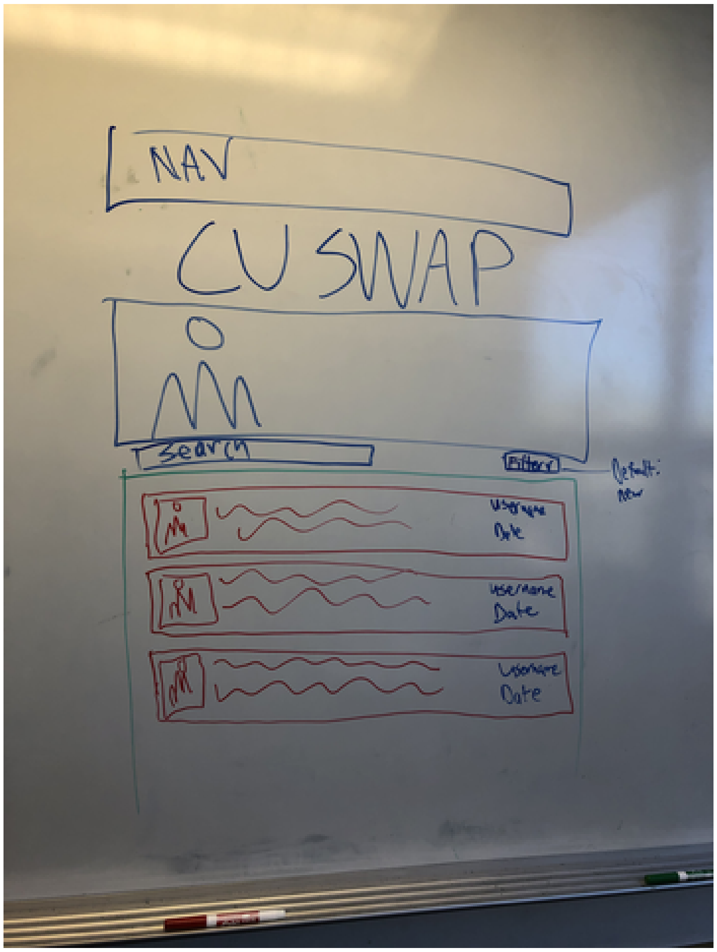


Front end design:

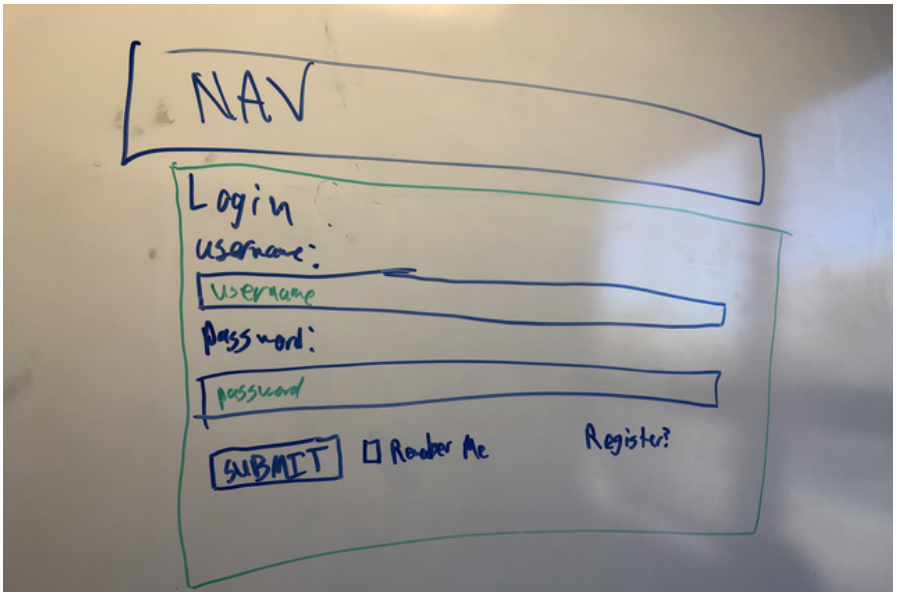
Nav bar:



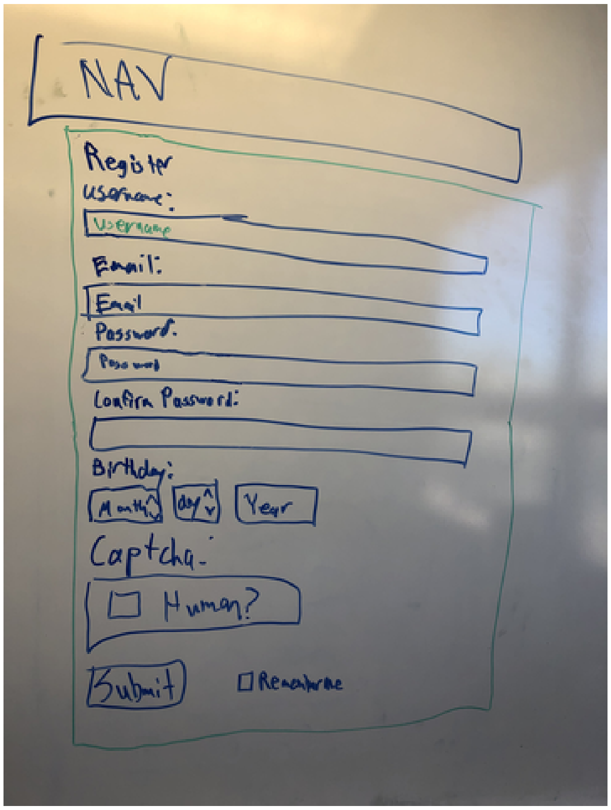
Home page:



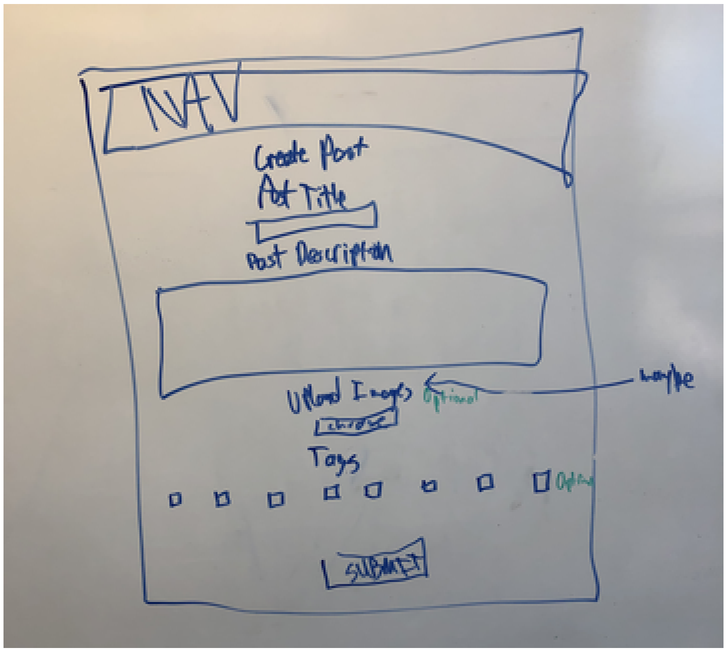
Login page:



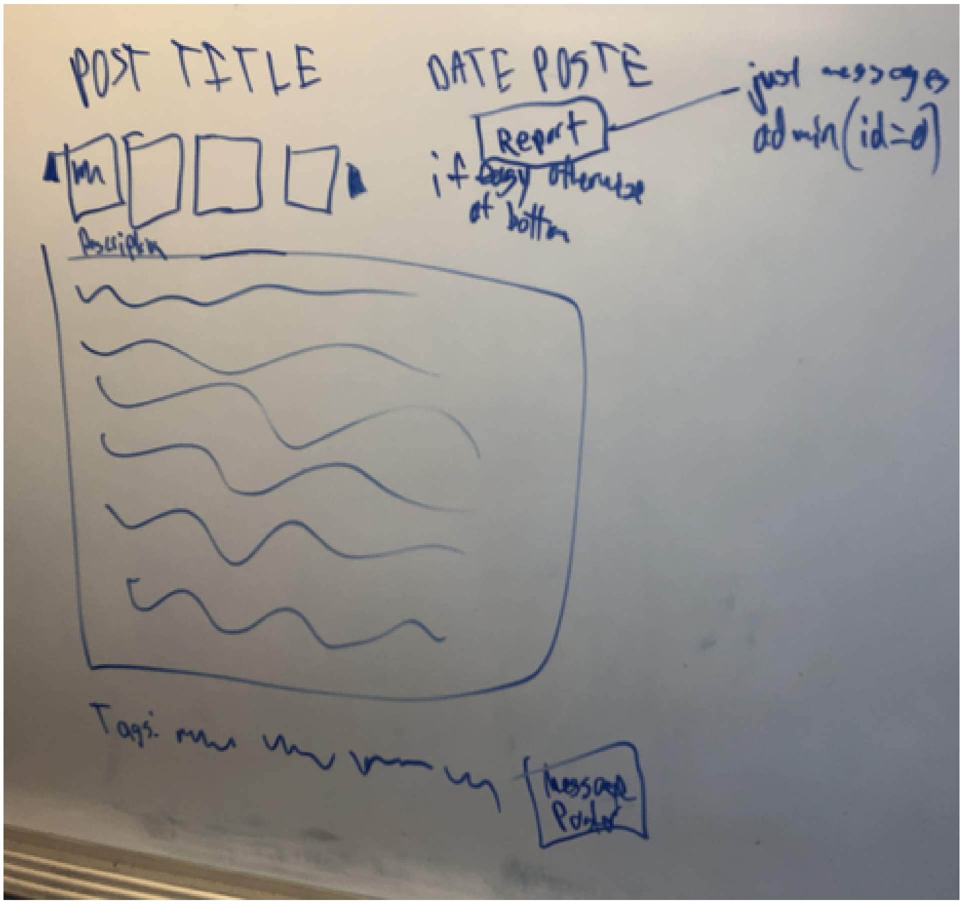
Register page:



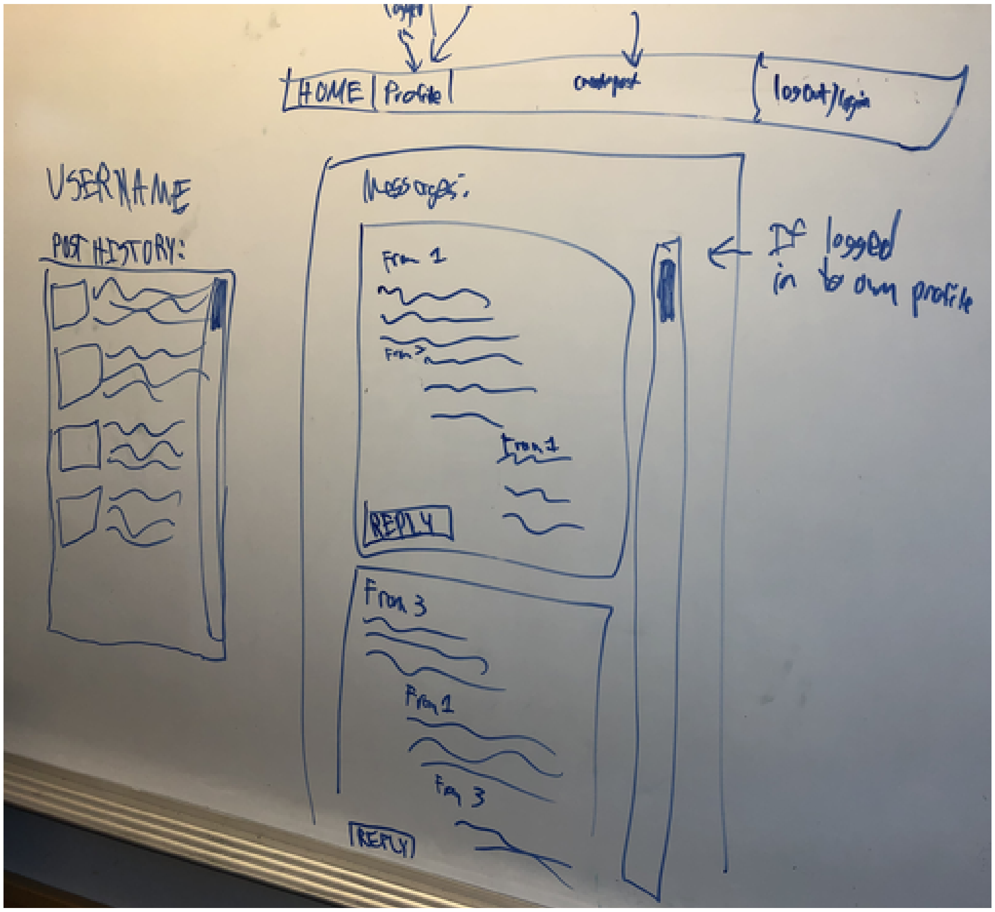
“Create post” page:



“View post” page:



User profile page:



Web service design:
Our application doesn't utilize APIs

Database design (utilizing Postgresql):
We will be using Postgres as our database. We have two tables, a user table that keeps track of user id, name, session cookies, messages, and passwords. The other table will keep track of all the posts which include post id's, titles, post body, filter tags, date of creation, the poster ID, and if the post is completed or not.

- Tables
  - Users
    - User Id (Serial Primary Key/Integer)
    - Name (50 char array)
    - Session Cookie (Int?)
    - Messages (Array of Strings)
    - Password (500 array char)

- Posts
  - Post ID (Integer)
  - Post Title (String)
  - Post Body (String)
  - Tags (Array of booleans)
  - Date Created (Date)
  - Poster ID (Integer)
  - Completion of the post (Boolean)