

Group: 202-1

Team Name: Work in Progress

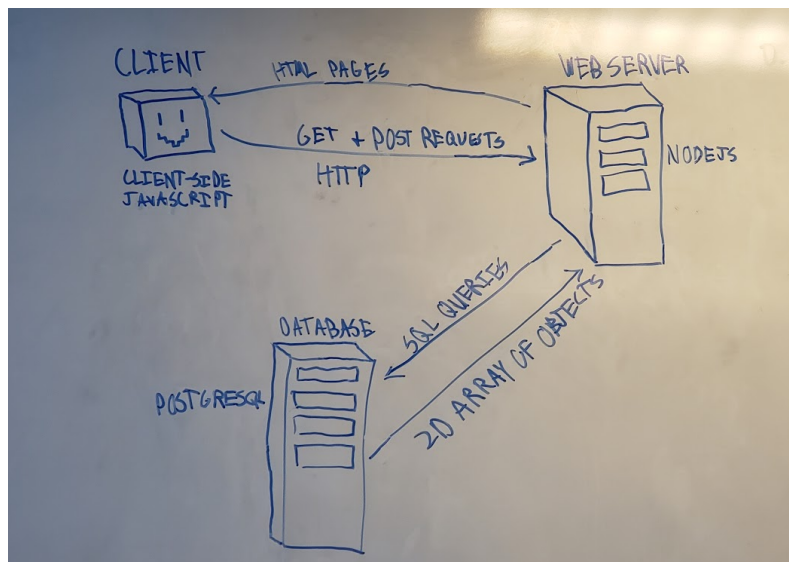
Team members: Evan Kuehr, Kaylee Engelhardt, Nick Price, Leon Shen, Gunther Wallach

Application name: CU Swap

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

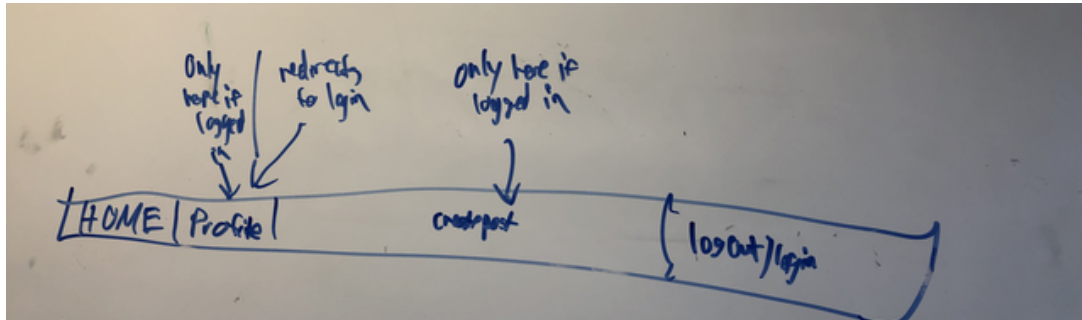
- Creating Posts (4)
 - Being able to add a new post to the database with tags, etc.
- Filter through posts (3)
 - View only certain posts based on selection of tags
- Login/giving user a session cookie (1)
 - Logging into the site and being assigned a cookie for your current session
- Register (7)
 - 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:

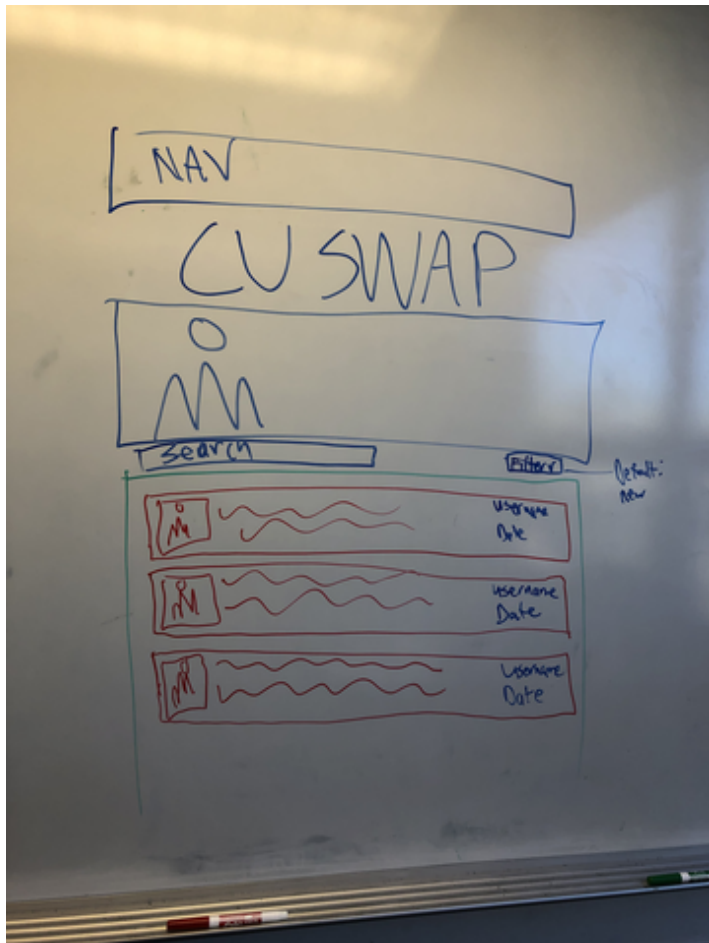


Front end design:

Nav bar:



Home page:



Login page:

A hand-drawn sketch of a login page layout. At the top is a horizontal rectangle labeled "NAV". Below it is a green-outlined box containing the text "Login". Under "Login" are two labels: "username:" and "password:". Each label is followed by a horizontal input field. The word "username" is written in green inside the first input field, and "password" is written in green inside the second. Below the input fields is a rectangular button labeled "SUBMIT". To the right of the "SUBMIT" button are two links: "Remember Me" and "Register?", each preceded by a small square checkbox.

Register page:

A hand-drawn sketch of a register page layout. At the top is a horizontal rectangle labeled "NAV". Below it is a green-outlined box containing the text "Register". Under "Register" are several labels and input fields: "username:" followed by an input field with "username" written in green inside; "Email:" followed by an input field with "Email" written in green inside; "Password:" followed by an input field with "Password" written in green inside; "Confirm Password:" followed by an empty input field; "Birthday:" followed by three input fields labeled "Month", "day", and "Year", each with a small downward arrow icon; and "Captcha:" followed by a rectangular input field containing a checkbox and the text "Human?". At the bottom of the green box is a rectangular button labeled "Submit". To the right of the "Submit" button is a link "Remember Me" preceded by a small square checkbox.

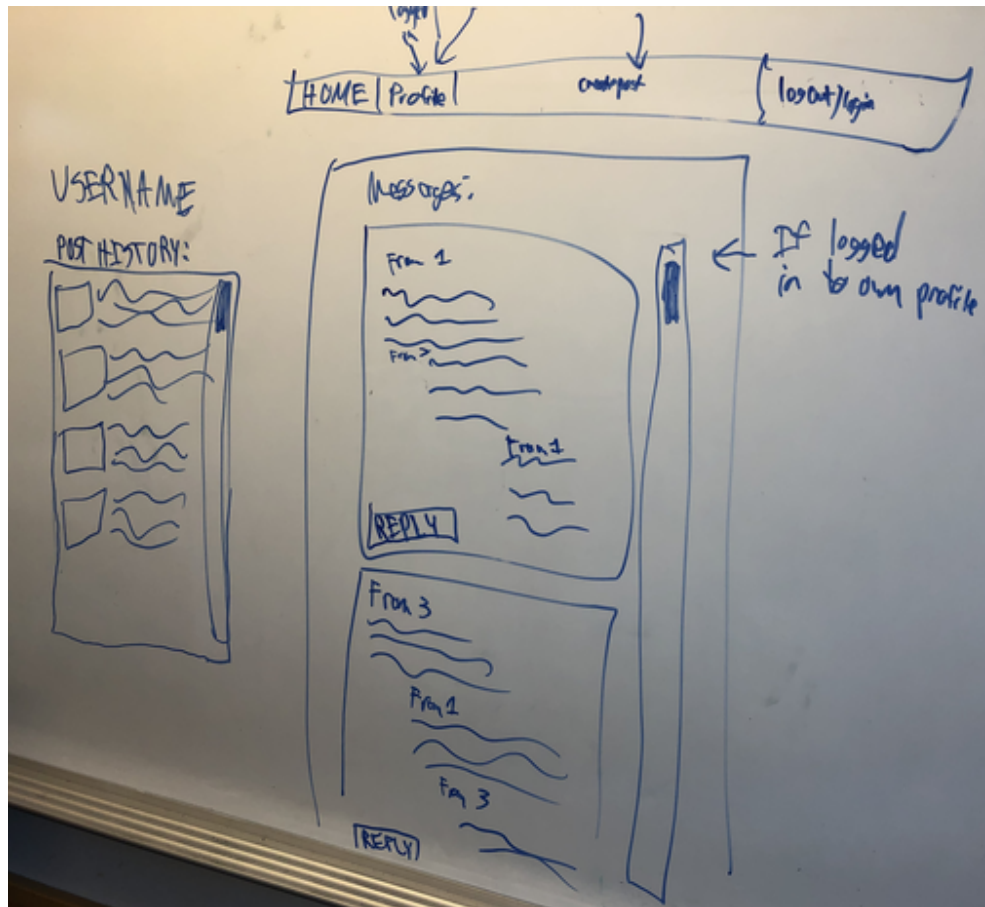
“Create post” page:

A hand-drawn sketch of a "Create post" page. At the top left is a box labeled "NAV". Below it is the text "Create Post". This is followed by a "Post Title" label and a text input field. Below that is a "Post Description" label and a larger text area. Under the description is a box labeled "Upload Images" with a "choose" button below it. To the right of the "Upload Images" box is a green label "Optional" with an arrow pointing to it. Below the image upload section is a "Tags" label followed by five small square input fields and a green label "Optional" with an arrow. At the bottom is a "SUBMIT" button. A handwritten note "maybe" with an arrow points to the "Optional" label next to the image upload section.

“View post” page:

A hand-drawn sketch of a "View post" page. At the top left is the text "POST TITLE" above a row of four square image placeholders. Below the images is a "Description" label and a large rectangular area filled with wavy lines representing text. To the right of the title and images is the text "DATE POSTED" above a "Report" button. A handwritten note "just messages admin(id=0)" with an arrow points to the "Report" button. Below the "Report" button is the text "if easy otherwise at bottom". At the bottom left is a "Tags:" label followed by four wavy lines representing tags. At the bottom right is a "Message Profile" button.

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)