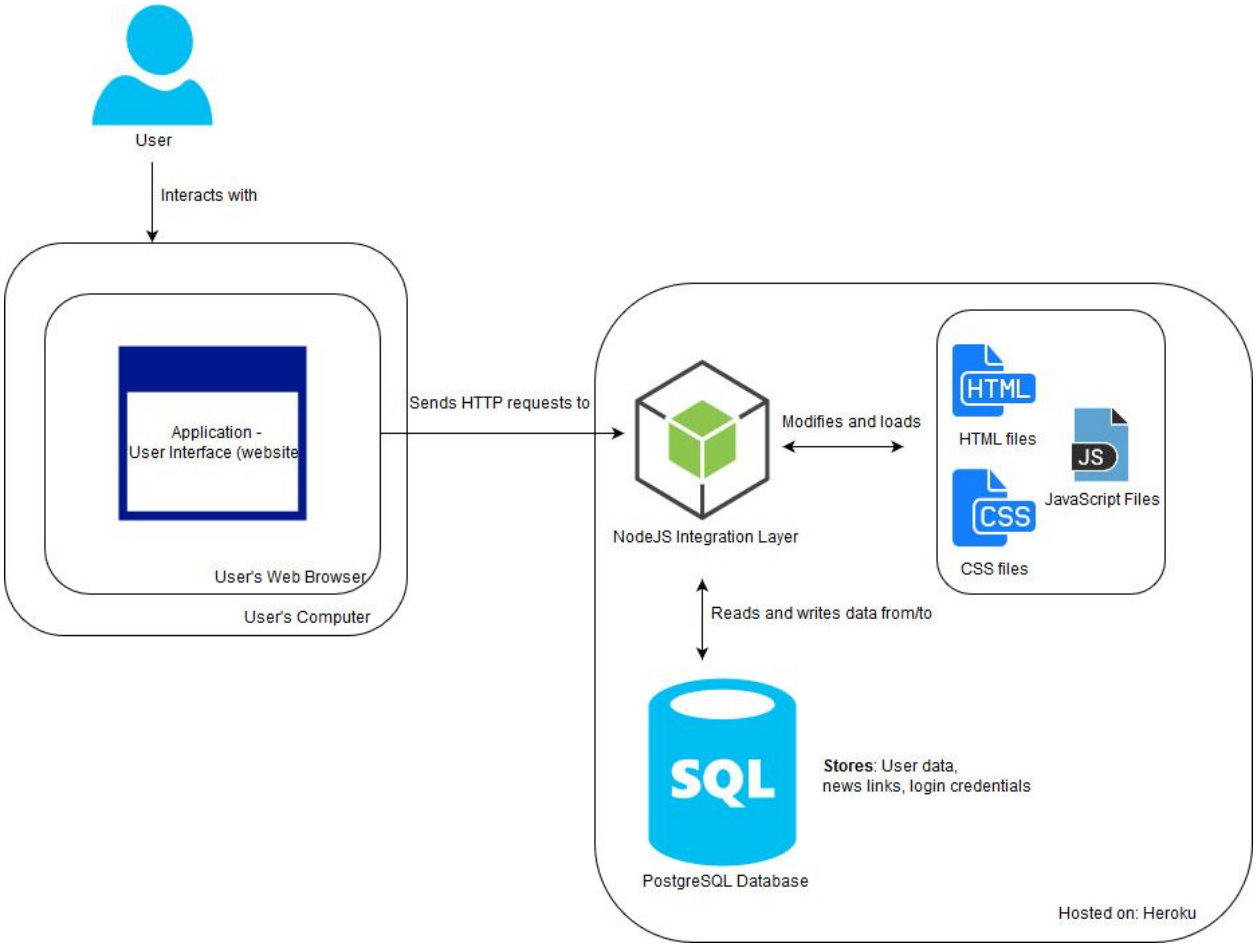
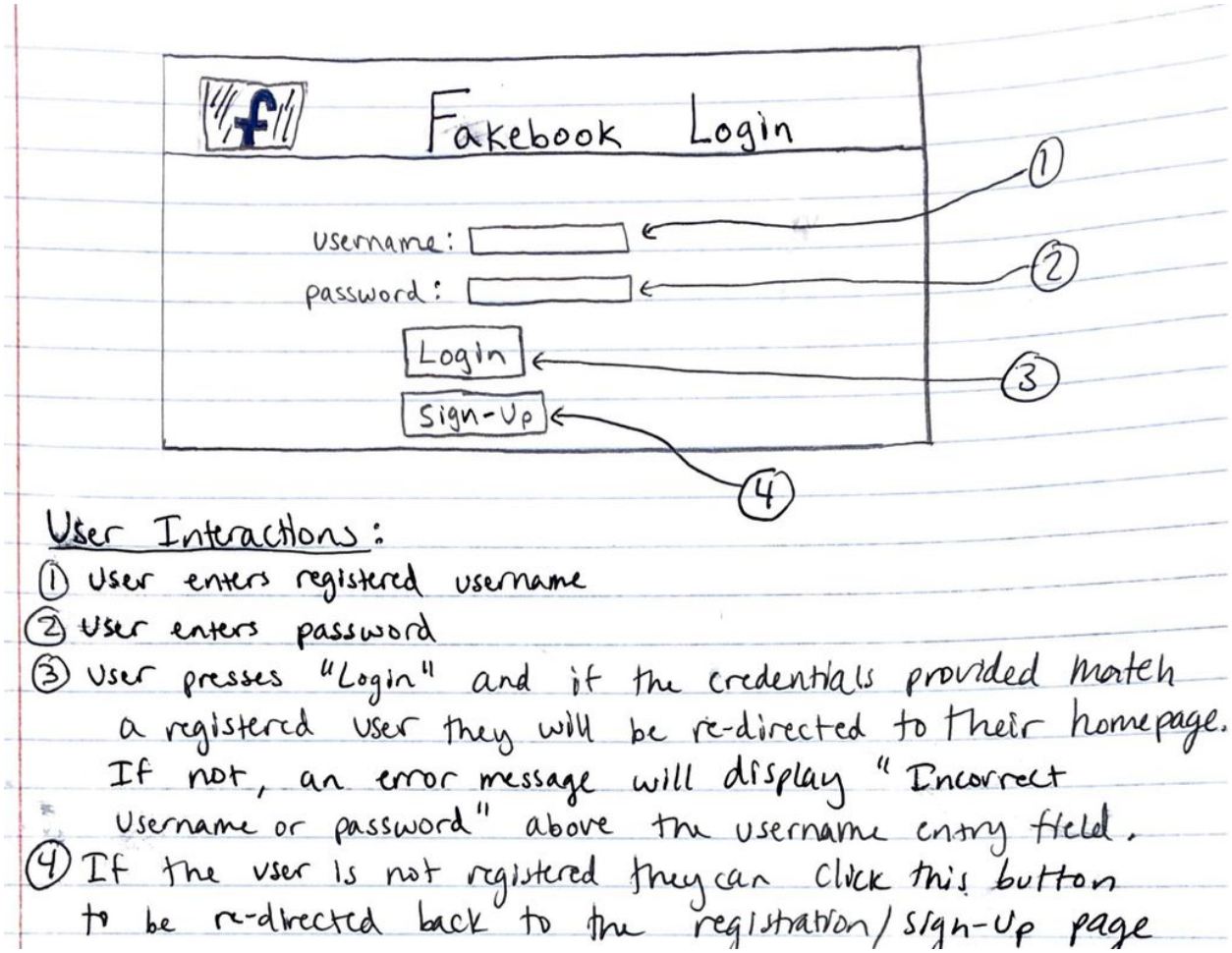
- Revised List of Features
    - Register
        - User inputs username, name, password, email
        - Javascript checks form to make sure that password is of certain length, contains lowercase letters, password and confirm password match, etc… (like in Homework 2)
        - Upon form submission, server.js file adds new user info to SQL table
    - Log in
        - User inputs login credentials and clicks submit
        - HTML form submits a GET request to server.js
        - Server.js logs user in to a session and returns the correct profile information
    - Manage sessions with cookies
        - Express-session in nodeJS server.js file
        - Ensure that when user is logged, in they cannot friend request themselves, they can edit *only their own* profile information, log out button is visible, etc.
        - Log in session to last (?? 1 hour ??) until automatically logged out
    - Log out
        - End session
        - Display log in page
    - View profile
        - User sees correct information of any user's profile
            - Profile picture, name, email, birthday, etc…
        - If user is looking at their own profile, they can edit the information, and do not have an option to send friend request.
        - If user is looking at another's profile, they cannot edit the information, and can send a friend request.
    - Update profile info
        - If user is on their own profile page, they can edit profile info
            - Profile picture, name, email, birthday, etc…
        - HTML form sends POST request to server.js to update SQL table
    - Post
        - User clicks "create post" on main newsfeed page
        - User is prompted with a text input popup form
        - Upon submitting HTML form, POST request sent to server.js that adds post to SQL table linked with user poster
        - The post can now be shown on newsfeed for user and friends of user
    - Display news feed
        - User sees 20 most recent posts from themselves and their friends
        - Dynamically rendered into ejs/html file using nodeJS
    - Search users
        - User searches name in search bar

        - Using regex(JavaScript), if name matches when “search” is clicked, all users that match that search string are displayed (name and profile picture).
        - When user clicks on a profile, he/she is guided to that other user’s profile page.
    - Add friend
        - When user is on another individual’s profile, “add friend” button is visible
        - When clicked, sql table of friend requests updates
    - Receive friend request notifications
        - GET friend requests from SQL table of friend requests
        - If any belong to user, they are displayed
        - User can click “accept”, and SQL table updates to add each user as each other’s friend in SQL table of users.
    - Comment
        - User sees “comment” form available with any post
        - User can enter comment and click “submit comment”
        - HTML form submits POST request to server.js that adds the comment text to a SQL table of the post
    - Display comments
        - Each post has an “accordion” that reads “show comments”
        - When clicked, each comment and along with the user that posted it is displayed in a list
        - HOW: server.js sends comments as variable in GET request to newsfeed, newsfeed ejs file builds comments in a
    - Like
        - Each post has a “like button”
        - When pressed, SQL table updates number of likes on a post
        - Posts show the number of likes

- Architecture Diagram

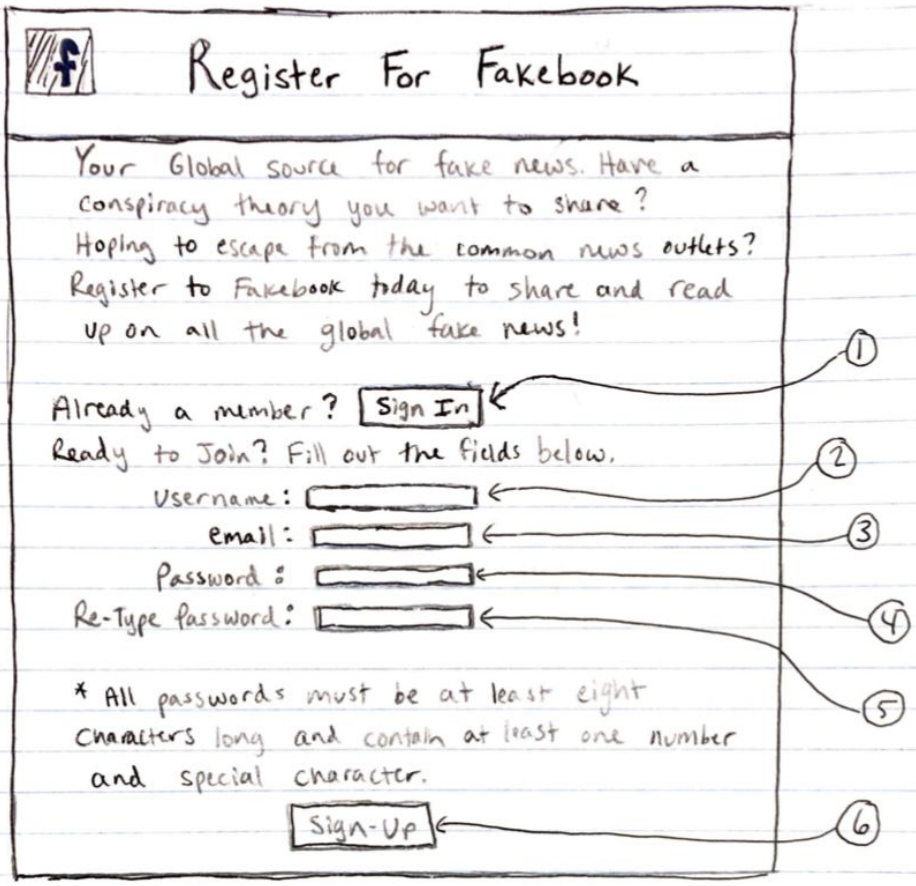

User
Interacts with
Application -
User Interface (website
User's Web Browser
User's Computer
Sends HTTP requests to
NodeJS Integration Layer
Modifies and loads
HTML
HTML files
JS
JavaScript Files
CSS
CSS files
Reads and writes data from/to
SQL
PostgreSQL Database
Stores: User data,
news links, login credentials
Hosted on: Heroku

- Front End Design



User Interactions:

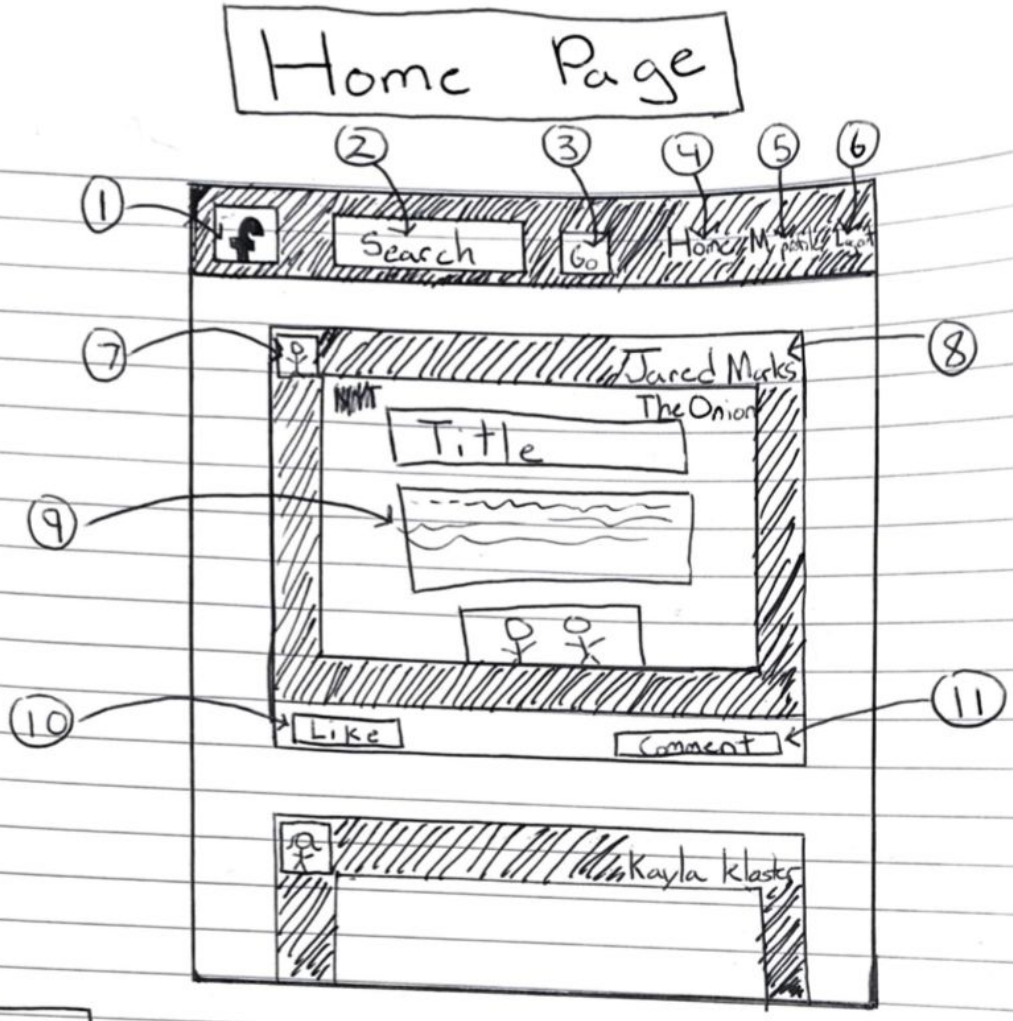
1. User enters registered username
2. User enters password
3. User presses "Login" and if the credentials provided match a registered user they will be re-directed to their homepage. If not, an error message will display "Incorrect Username or password" above the username entry field.
4. If the user is not registered they can click this button to be re-directed back to the registration/sign-up page



User Interaction:

1. If the user is already registered, it will take them to the login page where they can enter their credentials, and if they match a user in the database they'll be re-directed to their homepage
2. User enters a username of their choice, and if they press "enter" or "tab" it will take them to the next field entry
3. User enters their valid email address.
4. User enters a password that passes the given requirements
5. User must re-enter the password to ensure the user entered their password correctly
6. Once the user has finished entering their information, they can

Click this "Sign-Up" and if all the information is valid, they will be re-directed to the Homepage. However, if their password doesn't the requirements, or if the two passwords don't match, the registration page will reload with the corresponding error message in red at the top of the sign-up form.



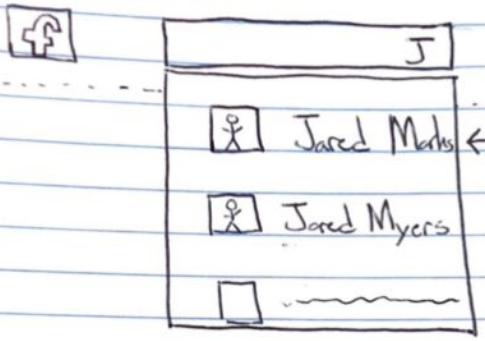
Defines

1. Webpage Logo - click to reload page
2. Search Bar - Search for friends / Add friends
3. Button for Search - onClick event
4. Nav Bar - Home - takes user to home page
5. Nav Bar - My Profile - takes user to personal profile
6. Nav Bar - LogOut - takes user to log in page
7. Image of article poster - click to goto their profile
8. Name of article poster - click to goto their profile
9. Article Overview - click to goto actual article web page
10. Like button - notifys poster that you like post
11. Comment Section - Add comment to article posting

Home Page

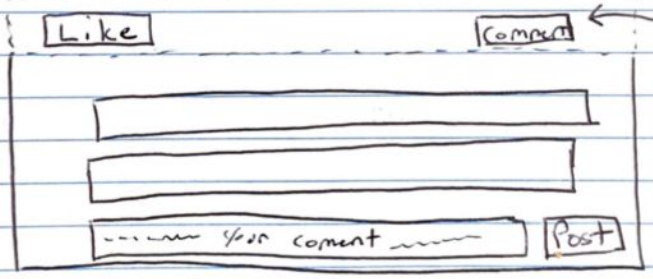
In-Depth

(2) Search Bar



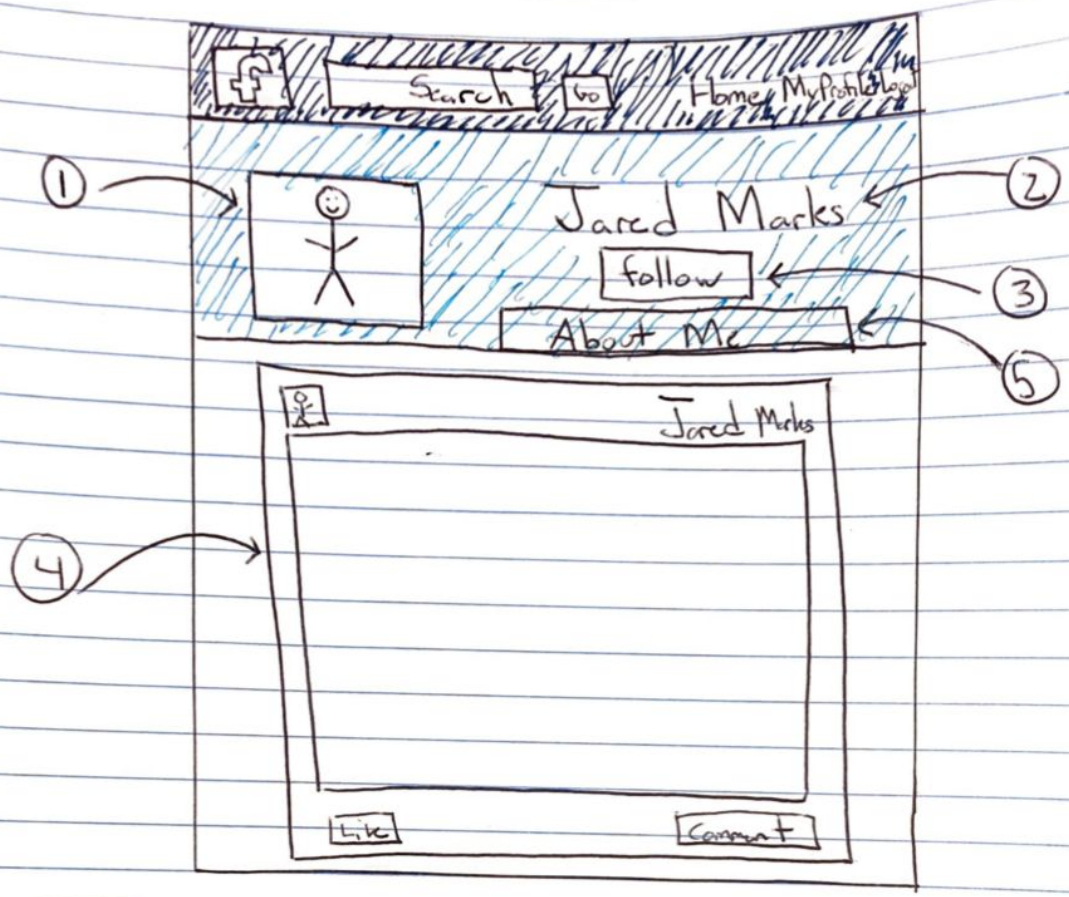
Display Search result in pop down menu, click to go to user profile

(11) Comments



Pop down comment menu onClick, view others comments and post your own

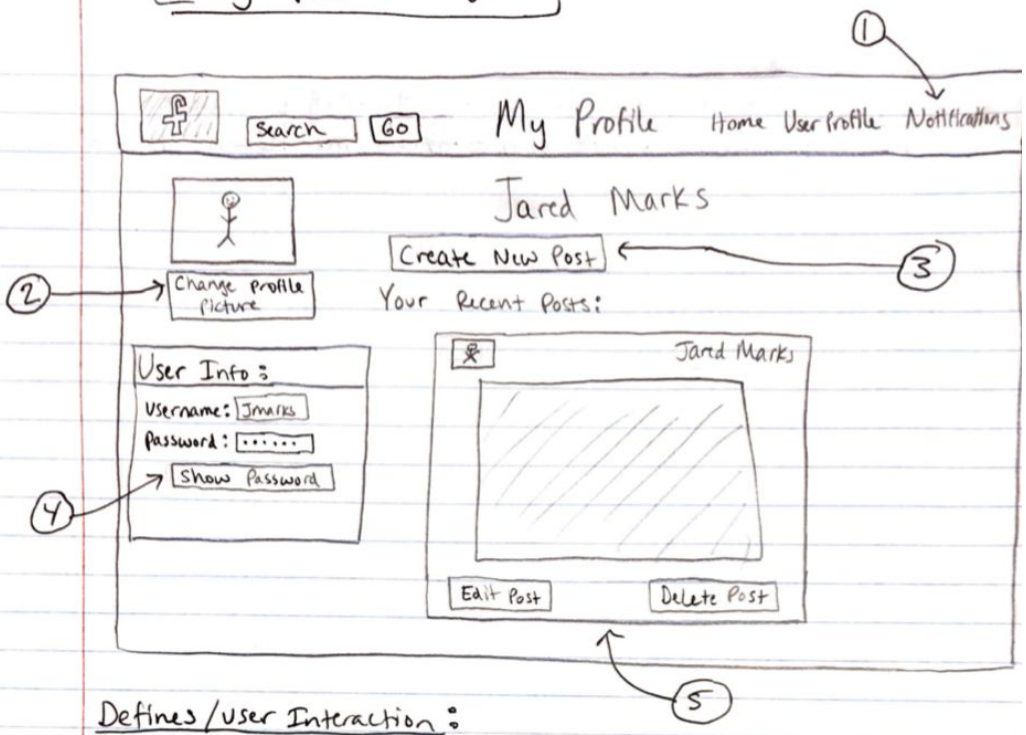
User Profile



Defines

1. User Profile Pic - Display image of the user
2. Users Name
3. Follow button - onClick() transitions to "following"/"Unfollow"
4. Recent Posts from user - Display 5-10 most recent posts from the user
5. About user section - The user can put a small bio to describe themselves to the followers

My Profile Page

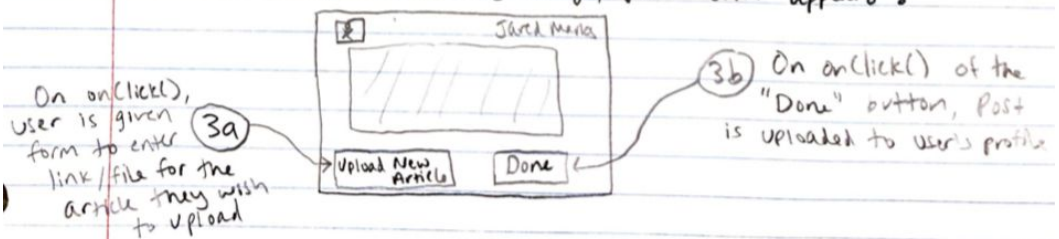


Defines / User Interaction:

① Notifications will show who has liked the user's post and other updates/new actions regarding the user's posts.

② Change picture Button – Allows the user to change their profile picture and uploads a new image chosen by the user

③ New Post Button – onClick() creates a new post that the user can upload a new article to, and when clicked the following image/post form appears:

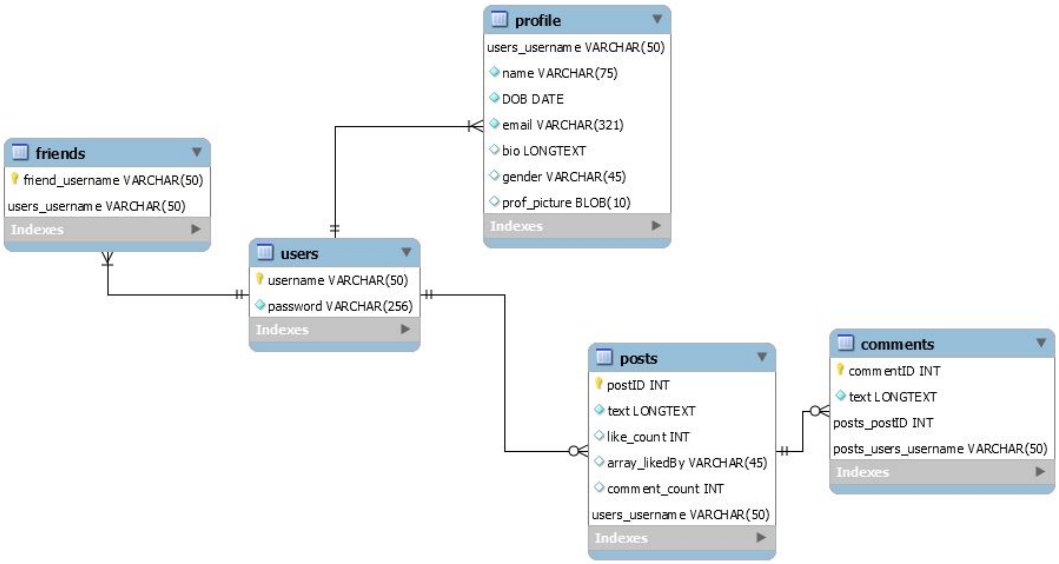


④ "Show Password" Button ~~always active~~ – onClick() transitions password field from "hiding" / "showing" user's password

⑤ User is given the option to edit or delete their posts. Posts are listed from ~~the~~ newest to oldest.

- Web Service Design
    - Not using any Web Services through APIs

- Back End Design
    - To be built in postgreSQL



Revised List of Features	This is an updated list of your FEATURES inventory (from Milestone 2). It is normal for feature lists to change during the course of a project. Some features may have been dropped. Some features may have been added. This revised features list should reflect these changes. This revised features list should identify the PRIORITY order of how the features will be developed.
Architecture Diagram	This deliverable is a picture or diagram that shows each architectural component of your application. The diagram should identify how your application's front-end, integration layer, and backend processes will be hosted. This diagram should identify the flow of data from one layer to another. This diagram should identify the protocols being used to/from each component layer.
Front End design	This deliverable is a series of diagrams that show the basic design of your application's front end. Typically, this design is most easily presented in terms of a WIREFRAME. This may be hand drawn, or created using a web page wireframe drawing tool. The front end design should identify each major feature of the application's front end and the
Web Service Design	If your application is using Web Services via APIs, this deliverable should list the Web Service being used along with a description of the Web Service's API including the data being passed to and received from the API.
Back End design	This deliverable provides a summary design of your application's database. The design document should identify each type of data being stored in your database. This may be documented in terms of a schema definition, showing data entities ("files") and attributes ("fields"). This may be documented via an Entity Relationship Diagram showing database tables and columns. The document should identify the specific DBMS technology being used to store your application data (PostgreSQL, MySQL, Firebase, etc.)