Software Requirements and Design Document

For

Group 3

Version 2.0

Authors:

Tommy Chong

David Markel

Victor Marques

Dylan Gire

Abraham Beltran

1. Overview (5 points)

The backend of the web application is composed of Python using Django Framework to connect to the frontend. The frontend uses ReactJS with an Axios component to connect to the backend. We are using SQLite for our database.

2. Functional Requirements (10 points)

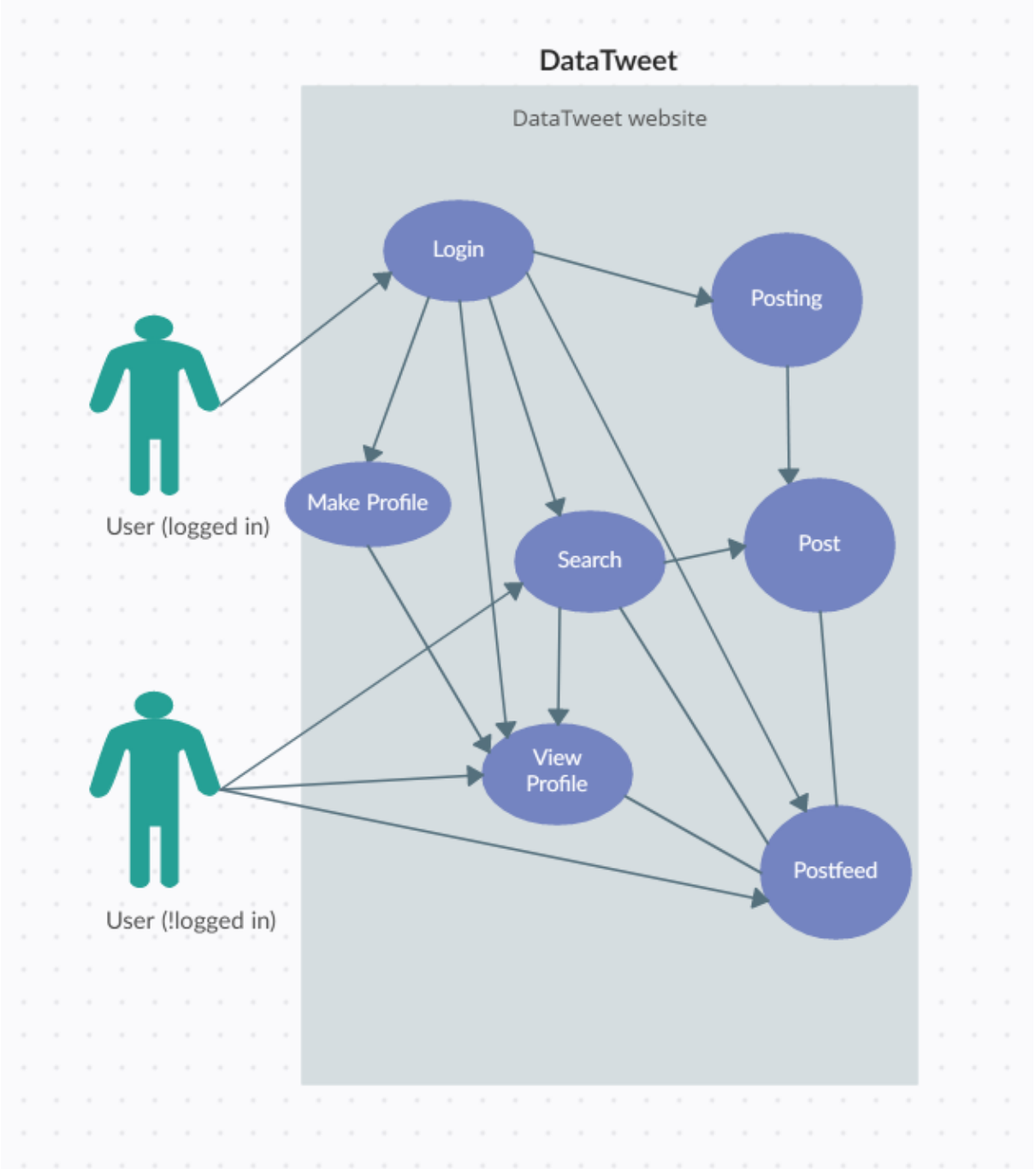
1. Login/Registration System (HIGH priority): A user is able to register with their username and email, along with the password for the account. A user is also able to log in to the page with credentials that are validated with our database.
2. Identify if a user is logged in (MEDIUM priority): The web application should redirect a user that is not logged in or whose logging time has expired to the login page. If the web application identifies that the user is logged in, it will redirect them to the home page.
3. Posting (HIGH priority): Users that are logged in can post from the home page. When posting, the user has to write a description of the data they are posting and upload the csv file in their posting. The web app will convert the csv to a graph, visualizing the data.
4. Posts Feed (MEDIUM priority): The homepage is also composed of posts that were posted by you or other people. It will be in chronological order, the posts that were most recently posted will appear on top of others.
5. Posts (MEDIUM priority): Clicking on the graph in the posts will be dynamic, meaning, it won't just be a JPG image generated. The posts will also have a number of likes, dislikes, and comments. Clicking on the post will take the users to a thread where users can see the comments made for the current post.
6. Profile (LOW priority): Users clicking on the profile tab in the navbar will take them to the profile page where they are going to be able to edit various personal information such as their first name, last name, etc… and also they are going to be able to see a feed of all the posts that the user logged in has posted.
7. Search (LOW priority): Users searching for specific posts based on keywords in a tab linked on the navbar. If possible, the user should be able to search by user or content to lead them to the person's profile and be able to look through their other posts. If the user wants to look specifically at that post, they should be able to see the comments, likes/dislikes, and other interactions.
8. The data visualization tool itself (HIGH priority). Users will be able to select what types of inputs they have and we will create a visualizer for them. This will be implemented after the basic website is made.

3. Non-functional Requirements (10 points)

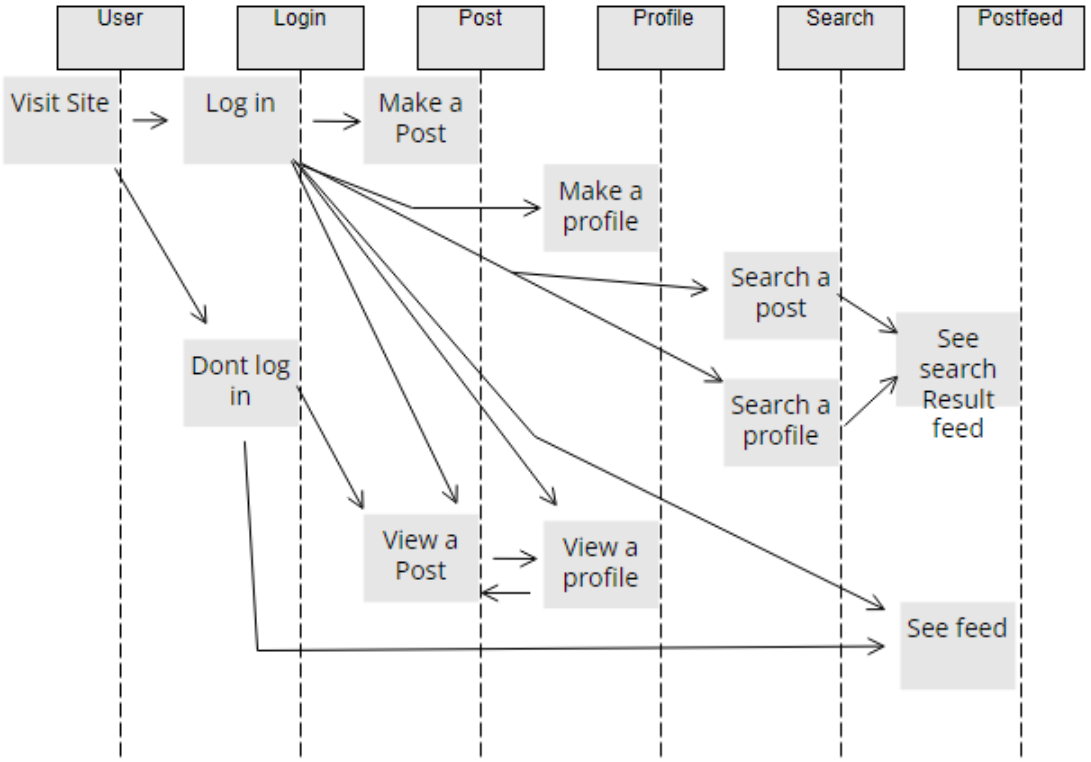
1. Login: In order for our web app to be scalable, users that log in will need some unique identifier for them so that a lot of users can log in at the same time. Thus, we can't use sessions for login.

2. Register: To make our web app more secure, registering an account will encrypt the username and password in the database. Once it is needed in the web app, it is going to decrypt it in the web app. This enforces security on the user accounts.
3. Posts: To be able to identify posts, there is a unique ID auto-generated by SQLite to make posts unique and the username associated with each post will be a foreign key from the Users table.
4. To have the website truly utilize react. This means that components only get rendered when they are updated as opposed to every time the page changes.
5. Reliability and speed of access when using the database associated.

4. Use Case Diagram (10 points)



5. Class Diagram and/or Sequence Diagrams (15 points)



6. Operating Environment (5 points)

The software will operate on any machine that is able to install Python3.8+ and is able to install ReactJS 16+. The machine also requires a web browser to access the web application. Specifically, Firefox and Chrome will be supported. Safari, edge, and explorer may not be tested. The software can be run in any operating system as long as it has the correct version of Python and React with the required libraries and modules.

7. Assumptions and Dependencies (5 points)

The program assumes that you have python3+ and React 16+ installed in your system and its dependencies (use “npm install” in the frontend folder to install all its dependencies). It also assumes you have npm installed in your system.

To run the program we assume you have your own copy of ‘sharedKeys.py’, if not please receive it from group 3.

Will consider hosting this website so that it is publicly accessible, but if not, then will give clear instructions on how to build and connect to the database.