Proposed Implementation:

- app.py
    - calls login()/logout() in utils logging in and logging out
    - calls register() to manage registration
    - calls poster() in utils to manage posting and commenting
    - calls homeDisplayer() in utils to manage displaying the home page
    - calls blogDisplayer() in utils to manage displaying posts
    - calls comment() in utils to manage posting and storing comments
- Data/
    - users.db
        - stores all account info (usernames and passwords) in a single table
    - text.db
        - stores all blog posts as individual tables; each table is named after the post
        - each table has a field for username, text, and time
        - the first record contains the original poster's name, content, and post time
        - also contains a main table called "POSTS" which contains the usernames of original posters along with their post names
- Templates/
    - homeIn.html
        - home page that is displayed when the user is logged in
        - logout button in corner
        - create post button in other corner
        - displays posts in chronological order (newer → older)
    - homeOut.html
        - home page that is displayed when the user is not logged in
        - login/register form in the upper right-hand corner
    - createPost.html
        - textbox for writing the post in centre of screen
        - submit button directly below
    - post.html
        - template for an individual blog post page
        - allows logged-in users to comment on the selected post
- Utils/
    - Python file named utils.py with functions for various operations
        - function "register()" first goes through the single table in users.db to see if a username is registered. If it is, it redirects to the main page where an error will be displayed, otherwise, it will add the username and a hashed password to the table in users.db.
        - function "login()" takes data from the form on the homeOut.html page as a username and password. If the username exists in the table and the hashed password matches the one in the table, then the user is logged into the session

and redirected to the homeIn.html page. Otherwise, they are redirected back to the homeOut.html page with an appropriate error message.

- function “homeDisplayer()” is called each time either the homeOut.html page is loaded or the homeIn.html page is loaded.
- function “blogDisplayer()” is called whenever a certain blog post is selected from the home page. Depending on the title of the post, the corresponding content is formatted/displayed, including all comments that were made.
- function comment() is called whenever someone selects the comment button on an individual blog post page. If the user is logged in, once the comment button is clicked, the page is reloaded to reflect the additional comment. The comment text is stored in the appropriate table within text.db. If the user is not logged in, the comment button will redirect them to the homeOut.html page with an error message asking them to login before posting
- function poster() is called when the user clicks the submit button on the createPost site, taking the contents of the textbox and filing as a new post in text.db. It also timestamps the post and retrieves the username from the session and puts them into the database as well

Division of work:

Nicholas - Back end stuff

- Write app.py
- Write utils.py

Reo

- write out all the html templates

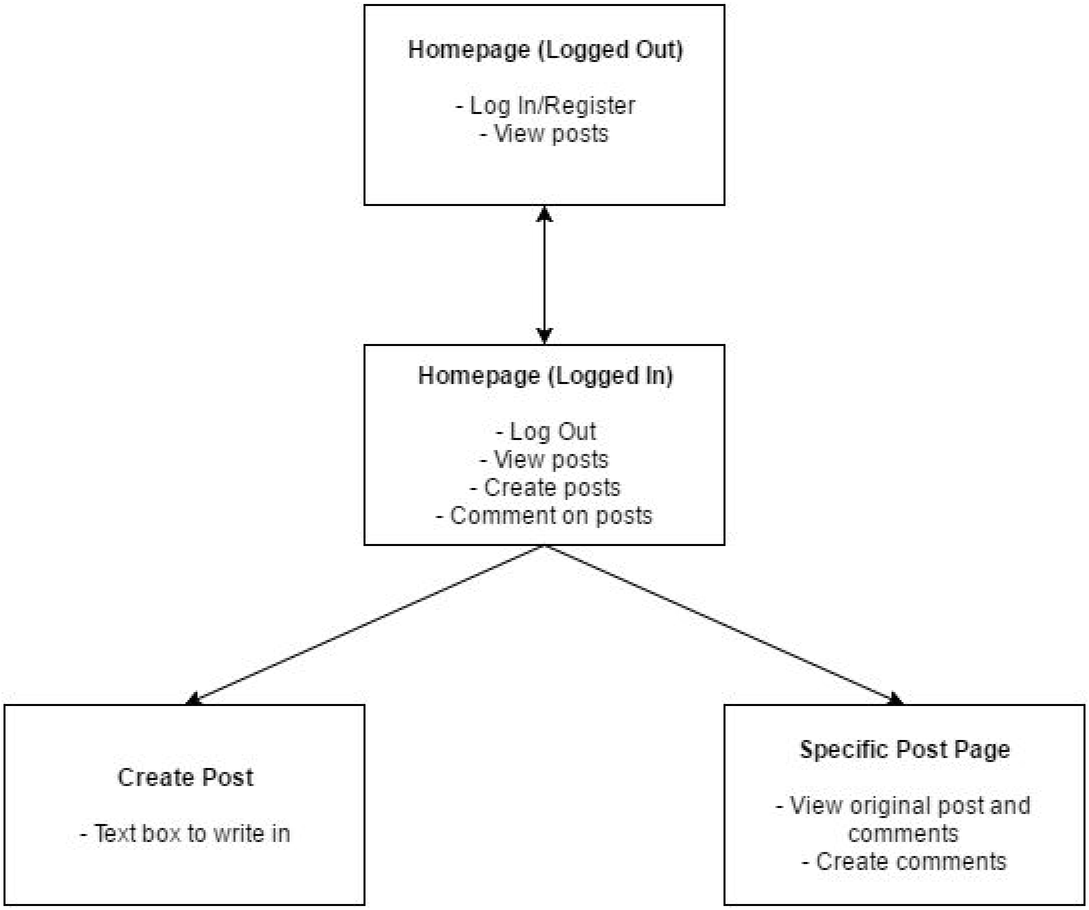
Jason

- work on authentication system (database - python interactivity)

Jonathan

- work on commenting/post system (database python interactivity)

Front End Site Map



Database Schema

users.db

USERS

Username	Password

text.db

POSTS

Username	Post Name

<POST NAME>

Usernames	Text	Time
<ORIGINAL POSTER>	<POST CONTENT>	<TIME OF POST>