A list of program components with role of each specified. - FARHAN

- User Experience
    - Login Page/ Registration
        - Users Register with Name, Email, and Password
        - Users can Login with username or userID
- Can ask browser to remember the user using sessions
    - Homepage
        - Users see most recent posts with corresponding comments ordered by date
        - Users can create a new post, or add a comment to existing posts
        - Posts and Comments will be objects in python
        - Post Object will contain an ordered list of comment objects
- Login.py
    - Contains Login and Register functions
- Post.py
    - Posting and Commenting, Rendering specific post with comments
    - Contains Constructor for Post and Comment Objects
- SQL.py
    - Contains basic SQL functions, used as a util file
- Homepage.html
- Login.html

Explanation of how each component relates to the others. **Component map visualizing relationships between components.** - GABE

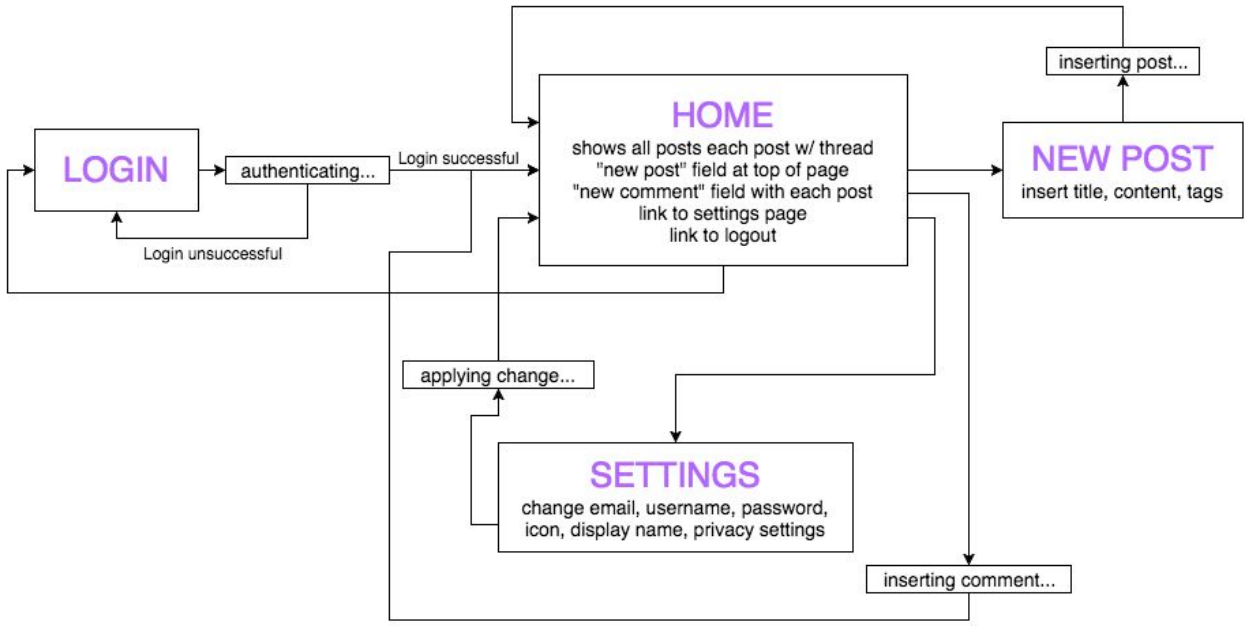
Database Schema - ZIYAN
User table: ID, name, hash pass
: Once an account is created, the user will be given an ID and his/her name and hashed password will be saved in this database. This is also the database the login page checks.

Posts table: (user) ID, date, content, pID (-> postID.txt)
: This database keeps track of all the posts that were made (not limited to the user). Whenever a user makes a post, the user's ID, the date, the content of the post, and a generated post ID will be saved in this database. The post page will access this database and display the posts according to their date (the most recent one goes on top). The ID keeps tracks of the user that made each post. The pID is used to link the comments saved in the comment database to their specific posts.

Comments table: user ID, post ID, comment ID, (-> commentID.txt)
: This database keeps track of all the comments in all of the posts (slightly chaotic but this is a small project so it shouldn't slow things down by much). Whenever a user makes a comment, the user's ID, the post ID, and a generated comment ID will be added to this database. The post page will access this database along with the post table database to generate the post page. Comments will be matched with their corresponding posts with their post ID and ordered using their ID. The user ID will identify who made the post.

Site map for front end - NALA

- Represent each page you envision for your site.
- Show linkages conveying user pathways for traversing site.



Description/Breakdown of how to divide labor amongst group

Ziyan: Registration and Login

- Develop structure for adding new users (assigning IDs), authenticating logins

Nala: Posts Management

- Implement write-to-file stuff so that post content and comments are stored and linked appropriately

Farhan: SQL

- Write Python scripts to create database tables and select appropriate data from them

Gabe: Frontend Design

- Design HTML templates for 4 main pages (login, home, new post, settings) — see site map