Home Hub - Your Personal Home Display
015-02

Member	Github ID
Zuojun Lin	Zuojunlin1
Caleb Lowe	caleblowe17
Dawson Davis-Pounders	Poundie72
Collin Lowe	colo3663
Pranav Kaliaperumal	pranavkkp4
Reese	

Project tracker - github project board

VCS - repository

Video Demo

Project Description: HomeHub - Your Personal Home Display is the central command center for the interconnected modern home. This application offers an intuitive graphical user interface that allows users to seamlessly manage and integrate their daily tasks and important information into one customizable dashboard. Key features include:

Editable Daily Calendar and Shopping List: Users can easily organize their schedules and shopping tasks by adding, editing, or removing entries on a responsive daily calendar and a dynamic shopping list, both accessible directly from the HomeHub dock.

Weather Display: HomeHub provides real-time weather updates and forecasts tailored to the user's location, offering convenience and preparedness as part of the daily routine

Health Tracker Information: By displaying health data from user input, like total exercise and sleep data, HomeHub encourages a healthy lifestyle and enables users to monitor their physical activity and well-being with ease.

Traffic Displays: Commuters can stay ahead of the game with real-time traffic updates and personalized route displays, helping to optimize travel times and reduce the daily commute stress.

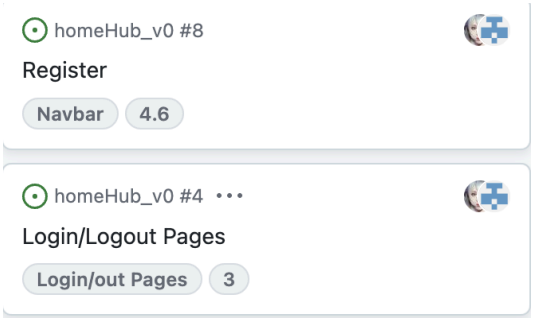
To-Do Lists: HomeHub helps users stay on top of their tasks with customizable to-do lists that keep important tasks in clear view, ensuring nothing gets missed in the hustle of everyday life.

Useful profile info: The profile page provides users with a date and time that updates every second, as well as a randomized quote to welcome them. There is also a change password button that redirects them to a page that allows them to change their password. Here, the user has to enter their old password and then enter their new password twice, and the form can't be submitted unless the new password and confirm password match. If the old password is wrong, the user gets a message after submitting it.

Contributions:

- Dawson Davis-Pounders
    - I created the base file directory and template including our nav bar, initial login/register/logout which was amended by a different member, as well as implementing the health tracker functionality. The health tracker included an exercise calculator that totaled calories burned, as well as a sleep debt tracker. I also helped debug our group's code as well as implementing all of the features into the final page so all modules were visible and usable.
- Zuojun Lin
    - I've completed the login registration and logout functionality with Dawson , and in discover I've completed the traffic conditions using the Google Maps API as well as the queryable

weather display using the openweather API.



- Caleb Lowe
    - I’ve completed the full calendar and its functionalities. It includes being able to add events and well as display the events on the right-hand side of the calendar container. You are also able to select a specific date if needed, as well as jump back to the current day. The events are stored and saved when the user logs out and logs back in. I also worked on user authentication, which allows only logged-in users to access the Discover page. If you are not logged in, clicking on the Discover page will redirect you to the login page.
- Collin Lowe

  I've completed the shopping list functionality, allowing the user to add and delete as many individual items as they wanted to, also provided the ability to show what items were added to the list by updating the list with the item that was added each time the user input a new item.
- Pranav

  I completed the profile page. This included the implementation of two API’s, the quotes API and the Date and Time API. The quotes API was implemented in the backend using Axios. The Date and Time API was implemented on the front end, and I used string splicing to display the information in the format I wanted. I also implemented the change

profile function. It uses event listener to check the confirm password with the new password using keyup. To check the old password, it was implemented in the background to check and if it does not match a message is sent using handlebars,

- Reese : N/A

Test Results -

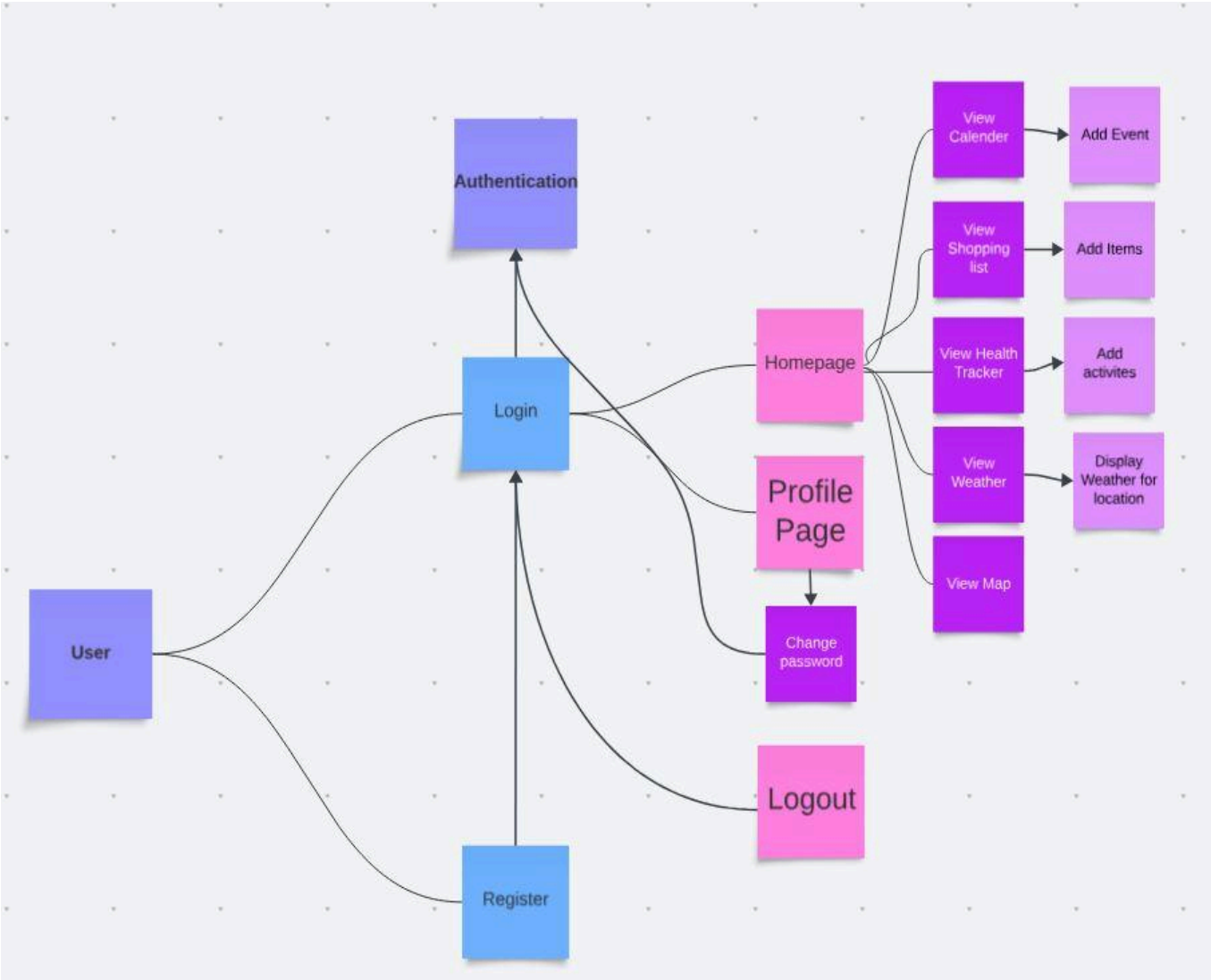
- Login - positive case - Attempt to login with valid user/password
    - Test Results: When valid login credentials were provided, the test case passed and the user was redirected to the main hub page, now logged into their account. The test was passed.
    - Observations: Existing users were able to easily login to their account given they provide valid credentials.
- Login - negative case - Attempt to login with invalid user/password
    - Test Results: When invalid credentials were provided, the user was redirected to the register page. The test was passed
    - Observations: Non valid credentials lead to a place where a new account can easily be created.
- Register - positive case - Attempt to Register with valid user/password
    - Test Results: When a non-empty and unused username/password were provided, the page redirects to log in where you can login with the new credentials. The test was passed.
    - Observations: New users are able to easily register if they provide non-empty, unique credentials.
- Register - negative case - Attempt to Register with invalid/empty user/password
    - Test Results: When an empty username/password was used, you cannot click submit, and a pop-up box prompting you to fill the field is displayed. When a pre-existing username is used, it will redirect you to an error page. The test was passed.

- Observations: While successful, the error messages could use some clarification and styling.

Deployment instructions-

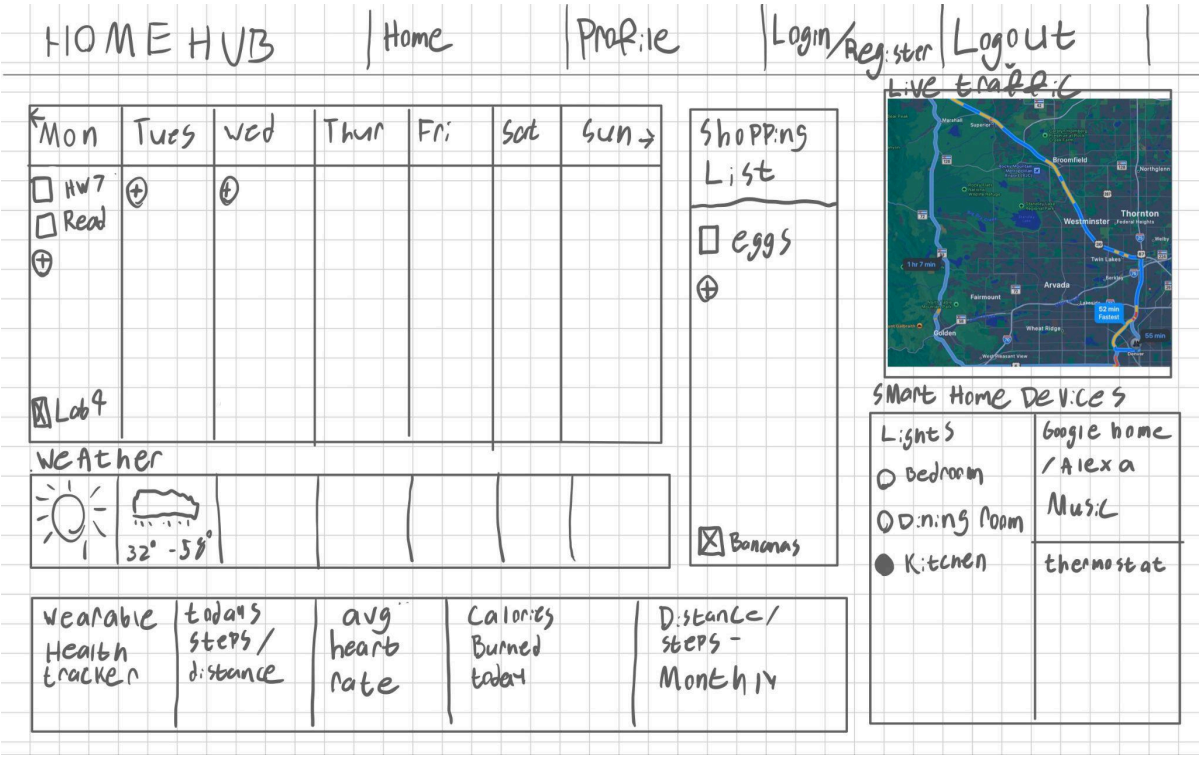
1. Clone Repository: Clone the HomeHub repository to your local machine.
2. Install Dependencies: Ensure you have all necessary dependencies installed. Use npm install to install dependencies listed in package.json.
3. Configuration: Modify configuration files (if necessary) for APIs, database connections, and environment variables.
4. Build Application: Use the appropriate build commands to build the application for deployment.
5. Deploy: Deploy the built application to your preferred hosting service following their deployment instructions.
6. Testing: Perform thorough testing to ensure all features function as expected in the deployed environment.
7. Monitoring and Maintenance: Regularly monitor the application for any issues or updates, and perform maintenance tasks as necessary to keep it running smoothly.

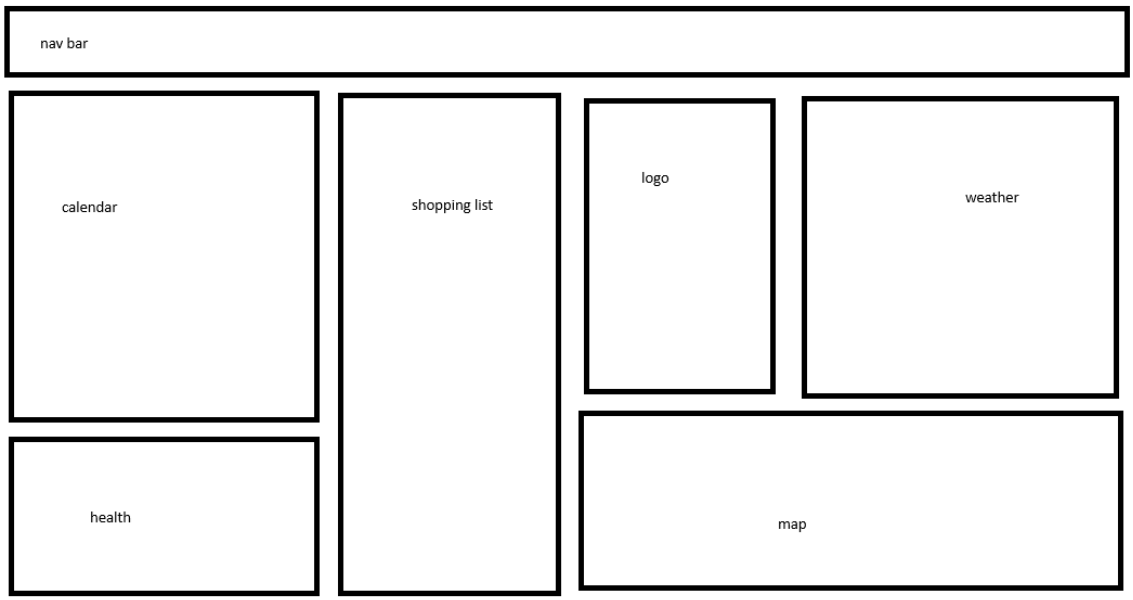
Use case Diagram -



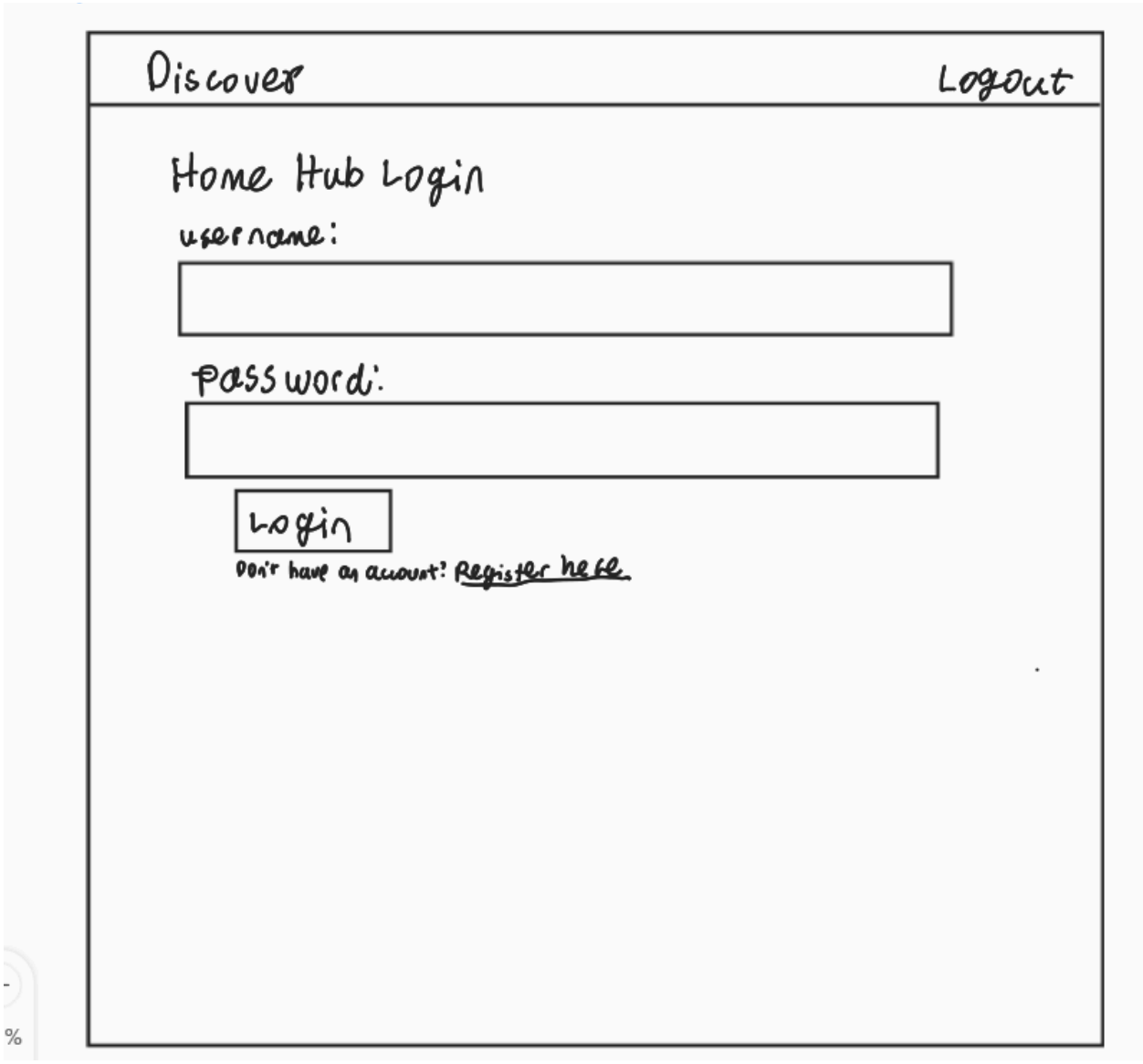
Wireframes-

Discover Page

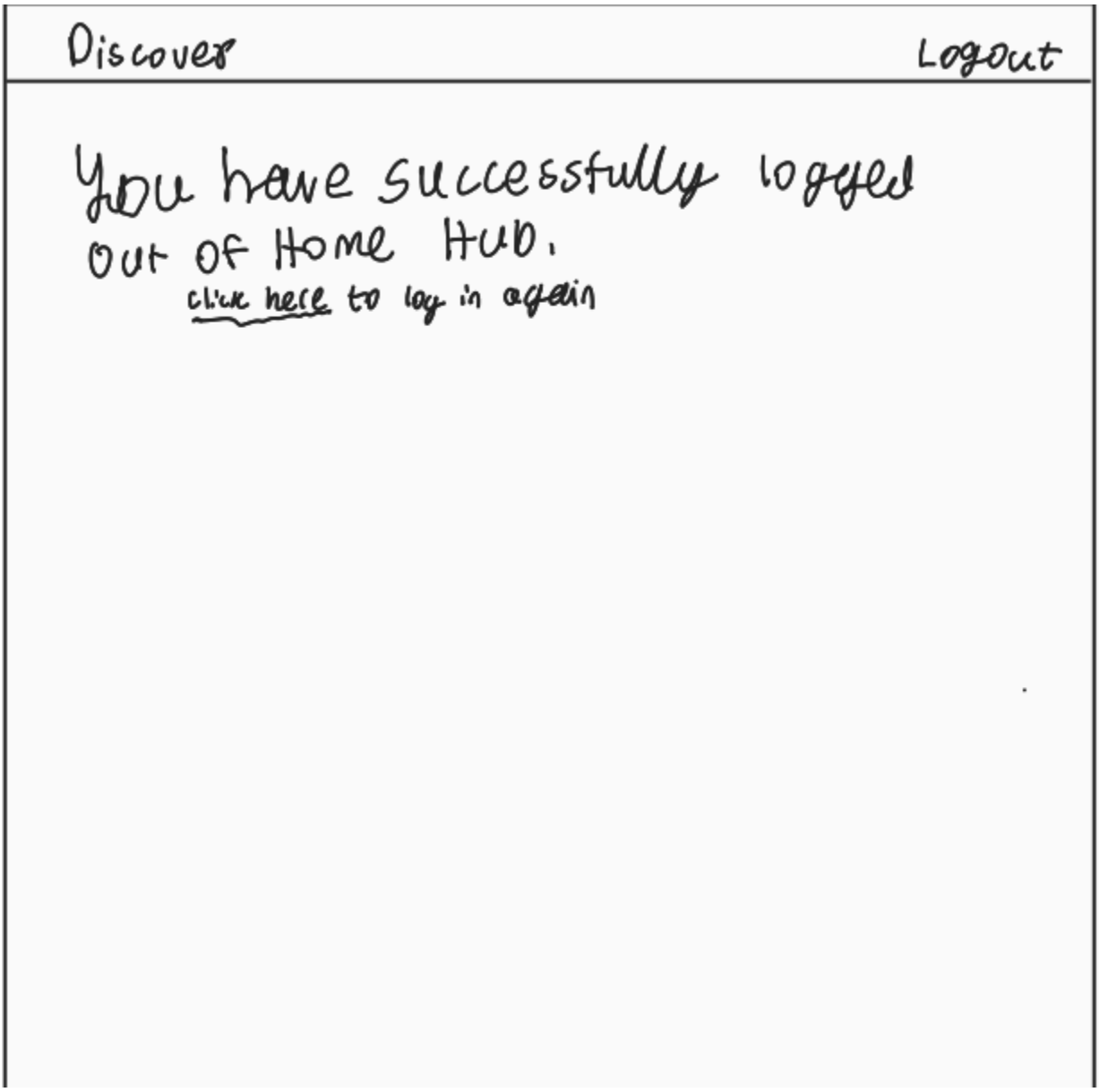




Login


Discover
Logout
Home Hub Login
username:
password:
Login
Don't have an account? Register here.

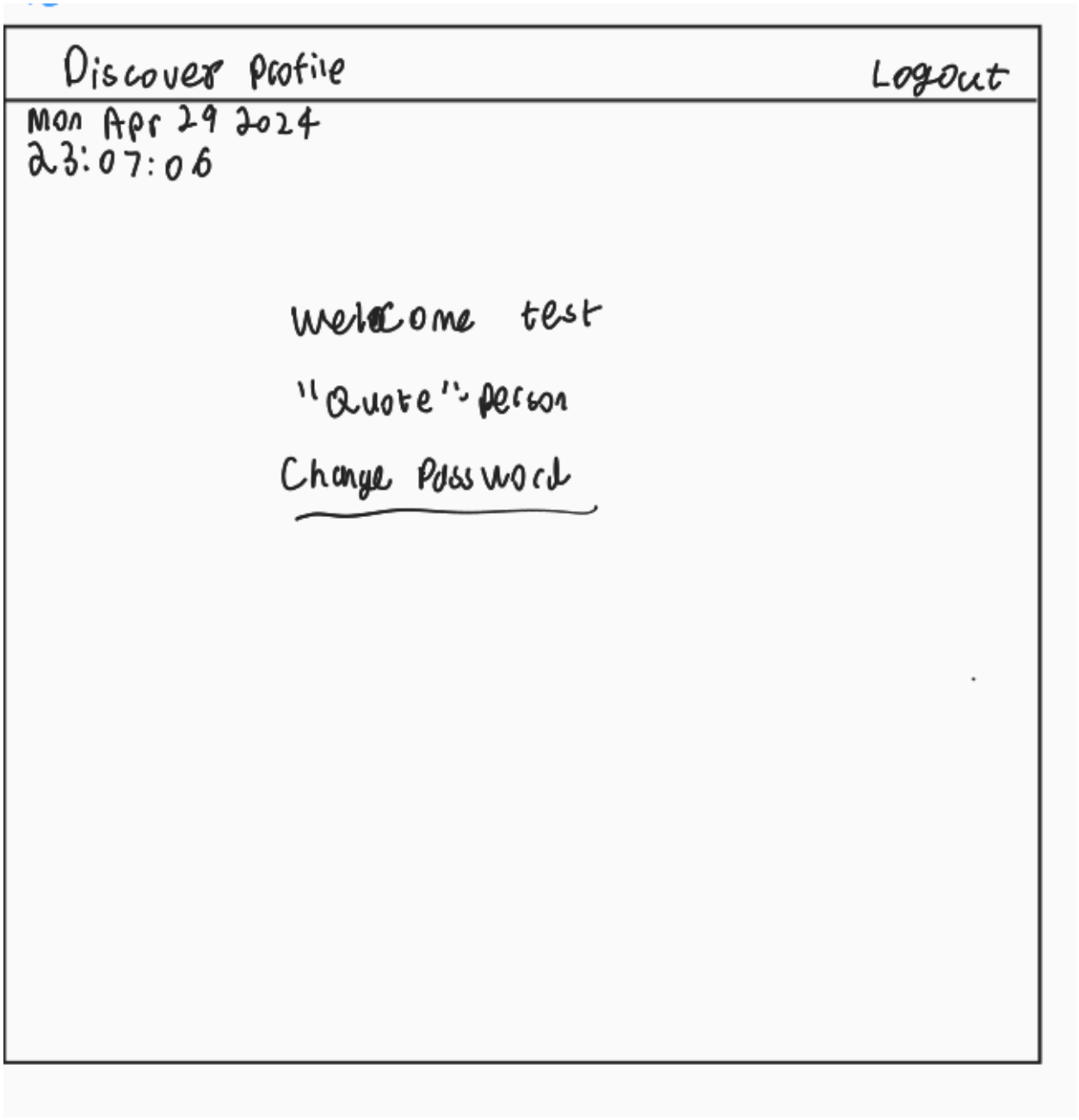
Logout


Discover
Logout
You have successfully logged out of Home Hub.
click here to log in again

Register


Discover
Logout
Home Hub Register
username:
password:
Register

Profile


Discover Profile
Logout
Mon Apr 29 2024
23:07:06
Welcome test
"Quote" - person
Change Password