Testing Report Memo

To: Professor Pisano, Professor Alshaykh, Professor Hirsch

From: ConnectBU (Hussain Albayat, Yousuf Baker, Benjamin Chan, Nadim Elhelou, Damani Phillip)

Team: 3

Date: 11/22/20

Subject: Prototype Testing Report

Equipment and Setup

For the prototype testing, we screen shared the demonstration via Zoom. We showed all the web application's functionalities that have been implemented so far, including but not limited to navigation across pages, Google Single Sign On, Redux state management, etc.

If you wish to test the prototype on your local machine, complete the following setup:

1. Go to ConnectBU's GitHub repository and clone the repository.
2. Install or update npm if you haven't already.
3. In the cloned repository, navigate to the frontend folder and install the project's node dependencies by running the following:
   ```
   $ npm install
   ```
4. Run the following command to run the web application
   ```
   $ npm run start
   ```
5. In your browser, navigate to "http://localhost:3000" to view the web app.

Functions Tested

We tested several functions of our webapps. These included:

1. App/Google login
2. Routing / User Flow
3. Page Interaction / Button Functionality
4. Dynamic page sizing
5. MySQL (hosted on AWS RDS) Relational Database entry
6. Human-computer Interaction Experience
7. Techno-psychological motivation behind the design

We employ a rigorous testing protocol in order to demonstrate these functions. For the first function, we show the google sso login capability using both previously registered and

unregistered user accounts and show that the two different sets of users would be directed to different starting pages. This leads directly into the testing of the second and third metrics, the routing and user flow, and button functionality. Once logged in, we show that the different buttons and menus allow the user to navigate through the webapp frontend as the buttons intend. With this, we show that users remain "logged in" as they navigate through the different pages of the web application unless specifically logged out. This is by going to different pages and demonstrating that the navigation bar still indicates that the user is logged in with the ellipse button: if the login state is not preserved, then a login button would display instead. Finally, dynamic page sizing is shown by resizing the page, and then also showing the mobile view of the page through the developer tools console in the browser. The database functionality is tested by entering a new user, and then querying the database to show that the user is added. The last two metrics are more qualitative design choices that are explained verbally during the walkthrough of the webapp and the testing of the above functionalities. The discussion of these two metrics includes discussing UX/UI and user-flow design choices, as well as justifications for which core functionalities of the webapps are chosen to be implemented in the MVP.

Conclusions

From our testing, we conclude that all of the above functions performed at the standard set for this round of testing. By employing the testing protocol outlined above, we adequately demonstrated the functionalities set out for testing. By going through the function testing protocol outlined above, we showed that users that have previously registered are directed directly to the profile page, whereas new users are directed to the sign up page. In this process we also demonstrated the functionality of the buttons, the page routing/user flow, and an initial (to be tested in an alpha distribution) efficacy of the user flow UX/UI design. Along with this, we demonstrated the functionality of the database, which implemented the database tables and schema as they were designed.

However, there are a few caveats to be worked on for future iterations. For example, we need to make sure that all the data (other than the student name, id, and major), are stored in the correct tables in the database. This also includes implementing and testing the ability for students to remove their affiliation from clubs, labs, and other fields. Finally, while the front end of the search and results pages were implemented, we need to connect these to the back end to allow for functional searching, and to allow for the results displayed to be pulled from the database rather than placeholders. Although, this functionality is largely streamlined because of the design and implementation of the relational database, which is cleanly organized and allows for quick search times because all searches will be done through queries.

Project

Item	Points	Comments
1. Summarizes <u>equipment and setup</u> (consistent with test plan?)		
2. Describes the detailed <u>measurements taken</u>		
3. Discusses <u>conclusions</u> based on test data		
4. Extra Credit (1 pts max)		
Total score		

0-not present; 1-present, treated marginally; 2-appropriately discussed