Software Implementation and Testing Document

For

Group 11

Version 1.0

Authors:
John Torres

Chris D
Dan E

1. Programming Languages (5 points)

List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

We are using React Native with TypeScript instead of JavaScript to build the Front-End component. We have yet to decide what framework we will use for the back-end or the database but we believe that a google cloud database would be good for distributing to users.

2. Platforms, APIs, Databases, and other technologies used (5 points)

List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).

We are currently using the React-Native, along with its associated API Navgiator in order to move fluidly through pages and send different information between components. As for what database we are going to be using as previously mentioned we think a google cloud database would be best to store and distribute the data. Possibly use firebase as well.

3. Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

We have tested the login to make sure that it takes in the correct parameters and only shows the user the homepage if they are authorized. As of right now only the team is authorized to access the home page. We have also created dummy User Stories that are semi-interactable, where clicking expands them and takes the user to a new page.

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

We have not begun testing for non-functional components of the project. We are still in the early development cycle and would need a bit more time to have a better handle on how to handle testing our non-functional components. We do however, plan to create access tokens instead of the current system of simply having an array with appropriate names and passwords.

5. Non-Execution-based Testing (10 points)

Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).

I have personally talked with the team multiple times as to how everything works and how I implemented each component of the front-end. I have explained how to download the repository and all the dependencies. In addition, I have explained how to create basic components and have left them pages to create, style, and test to their hearts content. I will also add these steps for installing and running the program to the READ.ME for better understanding.