# Software Implementation and Testing Document

# For

# Group 11

Version 1.0

**Authors**:
John Torres
Mark Mori
Wilfredo Huertas
Juan Dangon

## 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. For the database we are using Firebase, and as of right now we are not currently using a back-end framework. This might be implemented once we decide to distribute the app to multiple 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 Navigator in order to move fluidly through pages and send different information between components. For the database we are using Firebase to easily handle database interaction as well as allow for high scalability.

## 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 and register pages with our database to make sure that it takes in the correct parameters and only shows the user the homepage if a user's information has been authenticated by the database. The team is authorized to access the home page, where they can see a variety of posts made by other members of the team. Each user story is interactable with a functioning like button. Clicking on a user story expands them and takes the user to a new page. The team can create new posts, adding these user stories to the feed.

## 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 tested the speed of writing to the cloud database and have determined it is very fast at around 100 milliseconds. We have also tested increasing the latency between the user inputting text into a form and the overall application and have been able to reduce by about 20% since the initial implementation. We have also implemented testing for the security of an account by testing the FireBase cloud against inputting a very easy password such as 123 and confirmed it won't allow you to make an account with those types of parameters.

## 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.
Wilfredo has done code reviews and walkthroughs with members of the group primarily pertaining to database functionality and a bit of how data is moved throughout the project.