**Software Implementation and Testing Document**

**For**

**Group 2**

Version 2.0

**Authors**:

Matthew Kolnicki

Jalal Jean-Charles

Noah Shaffer

Randy Toyberman

Alexandre Jean

# Programming Languages (5 points)

We are using swift and swiftUI as the programming languages for our project. These languages were chosen because they are the most compatible with iOS development and have integration with XCode allowing us to simulate our app within the XCode IDE*.*

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

We are using the Firebase API to authenticate and store user information. We wrote our source code using XCode and used iOS Simulator to test our app.

# Execution-based Functional Testing (10 points)

Functional testing is a type of testing that seeks to establish whether each application feature

works as per the software requirements. When testing our program we would cross check our

implemented features with our desired features. At each check we would ask ourselves if the

Desired features still are feasible to implement and make sense. We reworked our software

Requirements on more than a couple occasions to realign our project with our new or preexisting

Goals.

# Execution-based Functional Testing (10 points)

Test execution is the process of executing the code and comparing the expected and actual results. We used xcodes preview feature for much of our testing through early production of a feature in the project. When implementing a feature before adding it to the project. Xcode allows you to run a preview of the view. This includes a simulation of all the functionality of the view, which proved very useful to the Early error testing to cultivate something close to what we needed before ever really adding it to the Project. After testing through preview. We would implement it in the project by adding a redirect to the View. Then we would do more intensive error testing by intentionally trying to break the app. Running the Actual project and giving it values that we know are not what it is expecting and ensuring we get appropriate Output.

# Execution-based Functional Testing (10 points)

We did not implement this form of testing.