**Software Implementation and Testing Document**

**For**

**Group 5**

Version 1.0

**Authors**:

Benjamin Zech

Corin Bradley

Gregory Garmen

Ian Carter

Wesley Harris

# Programming Languages (5 points)

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

**Java** - We choose to use Java in the entire program because most of us are more familiar with android development in Java then other languages.

# 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).*

* Android Studio - Used to develop the app
* Google Maps API - Used to locate businesses during the business signup process
* Back4App - Used to store the users information that was used during the sign up process
* Android emulators - Used to test

# 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).*

**Increment 1**

Account creation and storage-We created Student & Business accounts then logged in to make sure we could create an account, login and that Back4App was storing the information properly.

Google Maps API- Testing was implemented by searching for a myriad of businesses in Tallahassee using the search bar in our app itself. When the business we were searching for appeared in the results shown, we clicked on that business which showed its location on an embedded Google map and an image of the business provided by Google. We confirmed accuracy by either being personally familiar with the business and its location or entering the constructed API URL into a real web browser and confirming the JSON data returned matched what was seen in our app.

**Increment 2**

Direct Messaging- Testing was done by live messaging other members in our group and ensuring it acted like a real direct messaging feature.

Profile search- Testing was done by simply typing the names of other user’s real names on the app and watching live populating of profiles appear on screen.

Event creation- Testing done by entering fields to create event and saving the event. Save confirmed by checking backend.

Instagram linking-Simply clicking the button on a user’s profile to load the user’s Instagram page confirmed success

Account credentials changing- Entering new information and observing changes on backend as well as in the app confirmed success.

# 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 ran the code on 2 different Android phone emulators to make sure that the program would actually run on multiple Android phones.

Tested signing up with a non FSU email address to make sure it only allows FSU email addresses to sign up

# Non-Execution-based Testing (10 points)

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

We walked through the code together to know how the different account types interacted with the profile page. Example, how would the profile look if a student is signed in opposed to a business