Software Implementation and Testing Document

For

Group <1>

Version 3.0

Authors:
Andres Gonzalez
Joey Jimpie
Keegan Webster
Robby Jones
Nicholas Ford

1. Programming Languages

We're using Java and Python for the majority of this project. Java in Android Studio and Python for scrubbing the data and pattern recognition. Python is best for prototyping, is intuitive, and easy to use. Throughout the project, there is a mix of use in Java libraries as well as Android Java libraries. We also use the org.json Java library for use in JSON parsing.

2. Platforms, APIs, Databases, and other technologies used

- Google Maps API
    - Used for the parking tab. The application allows users to select a garage to get directions to, and the Google Maps API will be used for accessing those directions.
- Transloc API
    - Used for the bus tab to get information about bus routes in the Tallahassee area. We focus primarily on FSU bus transportation. The Transloc API gives us time and location data for the various stops a bus can make in a route.
- FSU Transportation API
    - This API is used in the parking tab for parking garage data. This is a simply a web link that returns a list of parking garages and data in JSON format. This information is being logged for use in pattern recognition to determine which garage is most appropriate to park at a certain time of day.
- Android SDK
    - This is how our application is being developed. Our application targets Android mobile phones. The entirety of our application is being developed using the SDK, with help from the aforementioned APIs.

3. Execution-based Functional Testing

We have tested the Google Maps feature to make sure it picks up the device location correctly, and made sure that the garages are correctly marked on the map activity. We initialized the views for the bus, parking, and maps tabs and executed the code to test the GUI layout. The Bus_Routes class was added into the android project and was successfully tested for scraping the bus route stop names. Similarly, the parking garage information was initialized in the parking tab and modified using execution-based testing. The redirection to Google maps has also been implemented and tested.

4. Execution-based Non-Functional Testing

We've checked to see if the data being parsed is correct and up to date. We've had moments where the FSU parking service was offline and the data was being incorrectly displayed. We've also made sure to prevent the application from reading false data from the Starmetro API. This improved the stability of the app and prevents crashes.

5. Non-Execution-based Testing

We've gone through all of the code for the three main tabs (Garage, Bus, and Maps tabs). The bus route calculations have also been reviewed by the group, as well as the data analysis code for determining if a given time is best to park at a particular garage.