Software Requirements and Design Document

For

Group <1>

Version 2.0

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

1. Overview

This application will consolidate FSU's various transportation apps into one easy-to-read app. The app uses Starmetro's Transloc api for FSU bus data and FSU's parking api for parking garage data. The data will be represented on an Android app with two main views. The first view is for parking garage information and the second view is for bus route information. The Google Maps api will also be integrated for the parking view. This will allow users to easily identify garages on a map and find the quickest route from their current location. The bus view will show all relevant FSU busses and their predicted time until a particular stop (specified by the user). We will also have a pattern recognition feature that will take data currently being collected to give information on what times are expected to see high volumes in traffic.

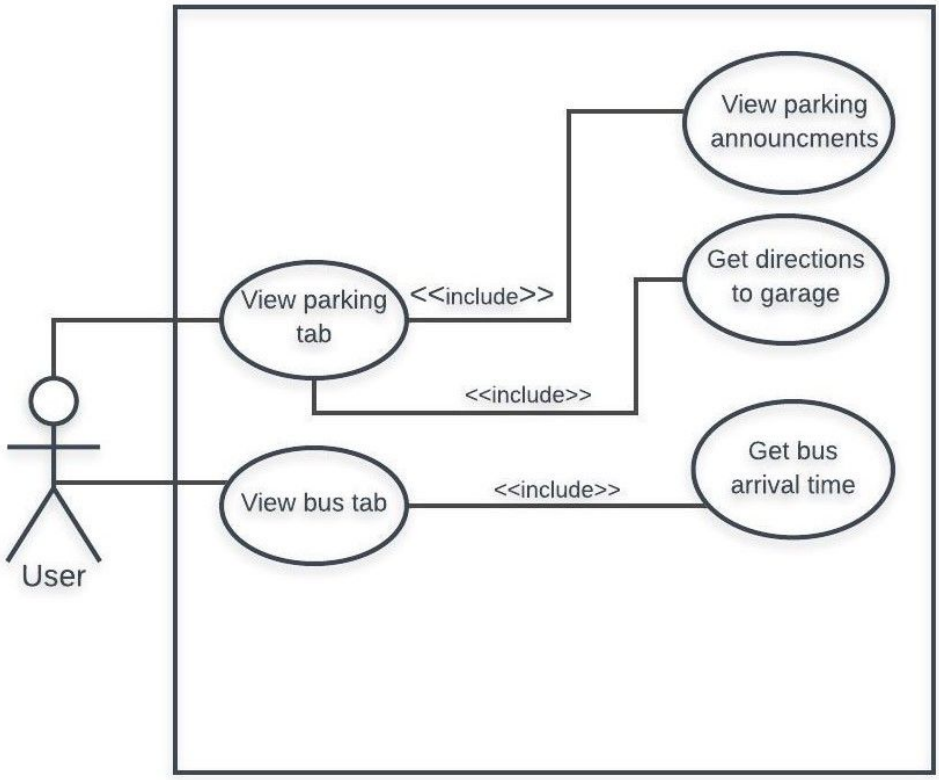
2. Functional Requirements

The app will display real-time information on how many parking spots are available at a given garage. We're getting the parking data from an FSU Transportation JSON source, which is simply a web page that updates garage availability data every five minutes (this is a high priority requirement). A user will also be able to find a given garage on a google maps tab, and then will have the options to get directions to a given garage from their current location (this is also a high priority requirement). We also have a dedicated bus route tab that allows a user to see FSU bus route information, such as how far a given bus from a particular bus stop (this is also a high priority requirement). We would also like to implement a pattern recognition feature that will use both previous and current parking information to give users an idea as to what times during the day does a particular garage see high volumes in traffic and show if there is a certain window where parking spots open up; such as the time in between classes, lunch time, and days where student traffic might be particularly light due to a school event (This is a medium priority requirement).

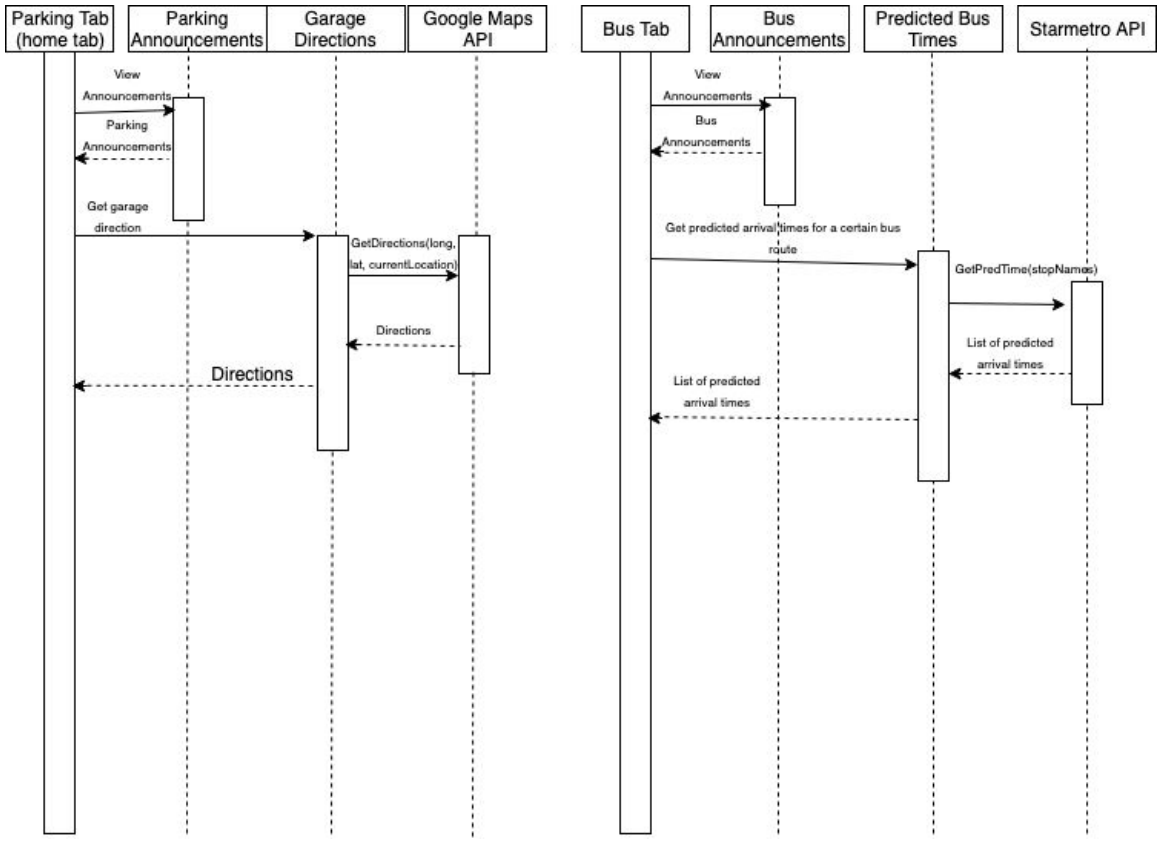
3. Non-functional Requirements

As far as system security and safety go, there are no real concerns because the application isn't collecting any sensitive user information, nor providing any. When it comes to software security, there are a couple of factors to consider. Since we are relying on information provided by the school, if for whatever reason the transportation API is down for maintenance or for some other reason, then our information will be out of date until the school's services are up and running. Our information is also limited to how often the information is updated by the school, which in this case is about 5 minutes. Additionally, we want to ensure that the data being retrieved from the APIs is as efficient/responsive as possible. The information presented should be the most up-to-date data regarding the API.

4. Use Case Diagram



5. Sequence Diagram:



6. Operating Environment

The system will be operating on Android OS. It also requires the device have Google maps in order for some features to work

7. Assumptions and Dependencies

Besides the case where the FSU Parking API is down for reasons beyond our control, there is the possibility that we reach our limited quota for map request, as the free Google API has a cap on the number of requests it'll take before requiring a commercial license.