Software Requirements and Design Document

For

Group <1>

Version 3.0

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 three main views. The first view is for parking garage information, the second view is for bus route information, and the third view is a maps tab with the garages displayed on a map fragment. The Google Maps api is used to display the map, and allow users to find a particular garage based on location, occupancy, or other reasons (The user's current location is displayed on the map). The bus view will show all relevant FSU busses and their predicted time until a particular stop (specified by the user). We also have a pattern recognition feature that will take data currently being collected to give information on if a particular time is expected to see a high volume in traffic

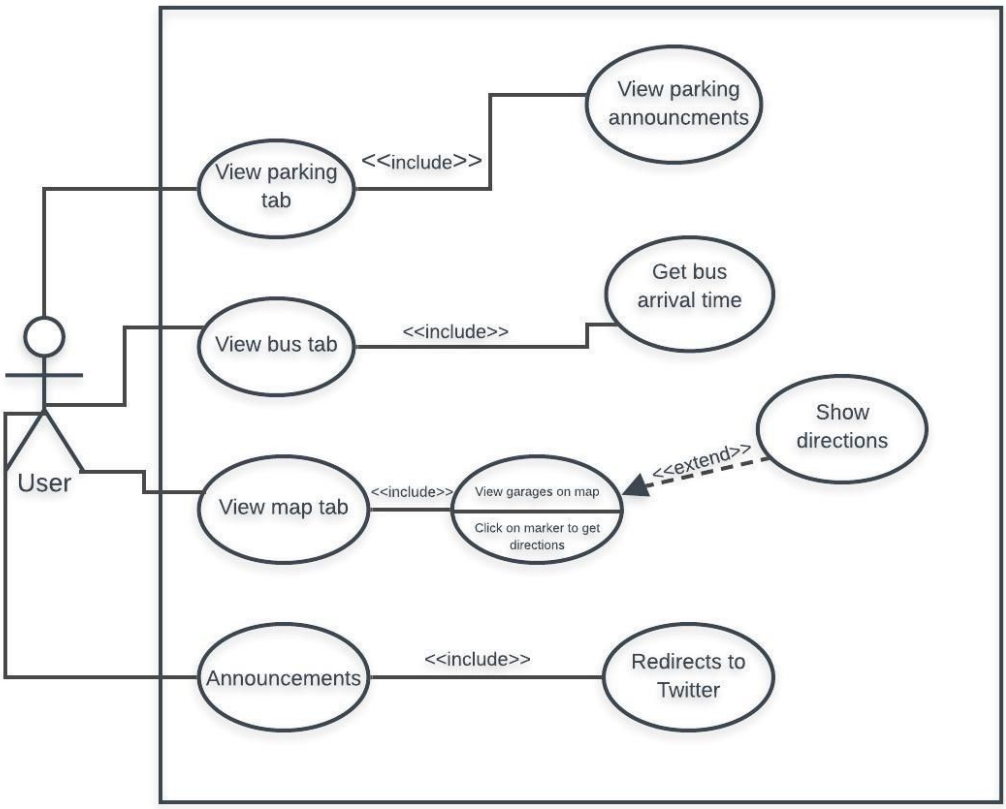
2. Functional Requirements

The app displays 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 have also implement a pattern recognition feature that will use both previous and current parking information to give users an idea as to if a given time period is best to try to park at a particular garage (This is a medium priority requirement).

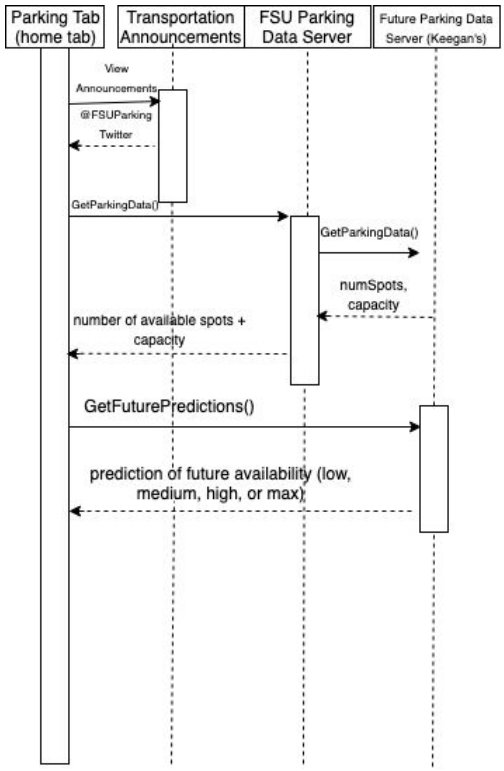
3. Non-functional Requirements

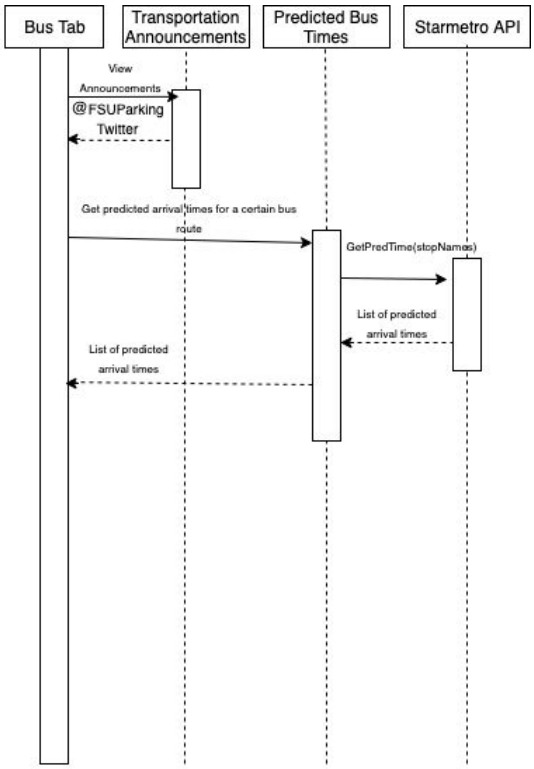
As far as system security and safety go, there are no real concerns because the application isn't collecting any sensitive user information, nor providing any. When it comes to software security, there are a couple of factors to consider. Since we are relying on information provided by the school, if 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. Although our app can be refreshed immediately by simply switching views or clicking on the tab the user is currently on, the data we're pulling from is refreshed every 5 minutes.

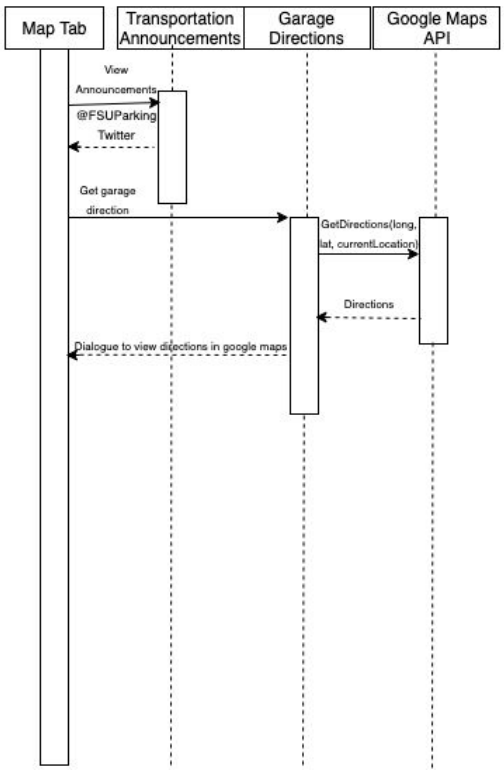
4. Use Case Diagram



5. Sequence Diagram:







6. Operating Environment

The system will be operating on Android OS. It also requires the device have Google Play Services installed, as well as Google Maps installed for the map features/directions to work properly.

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.