Where's My Bus? Process Description

Becky Leslie (beleslie), Bryce Martz (bmartz),
Daniel Houtsma (dhoutsma), David Li (lidav6),
Nicolas Bissiri (nbissiri), Dylan Whitlow (whitldy)

October 2016

1 Software Toolset

Programming Languages:

- Java in Android Studio - the group is already familiar with Java and some with Android Studio.
- Ruby on Rails on Heroku - great for rapid deployment, members have varying familiarity with these tools already and we expect Rails to have libraries and off-the-shelf solutions for server and database architectures.

Data Sources:

- OneBusAway API - buses and bus stop location and time data
- Android Location Services API - for various mapping features
- PostgreSQL database - default database for Heroku Rails, used for back end data such as the forum for route and neighborhood alerts.

Version Control and Bug Tracking:

- Git/Github - we expect to use the built-in bug tracking supplied by Github, more advanced bug tracking software feels out-of-scope

2 Group Dynamics

Manager: Becky
Design will generally be done as a group and implementation will be done by the following sub-groups, with the testing sub-group also overseeing integration of the ends.
Front End Team: Becky, Bryce, Daniel
Back End Team: David, Daniel, Dylan, Nick
Testing Team: Nick, Bryce, Dylan

The roles of specific members may change as we get further into the process and understand the developmental needs of the project more clearly. However, we would like to keep a structure of the teams such that everyone works on either the front end or the back end of the system and at least one member of either of those teams performs integration and testing of the two sides to keep the system unified.

Group disagreements will be settled by discussion and voting with the manager having the final say. We will be using Discord (group messaging and VoIP), a Facebook group, and email for communication outside of our regular meetings.

3 Schedule

Release	Delivered Features	Front End	Back End
Software Design Specification	Design	Designing, learning Android Studio	Designing, learning Ruby on Rails and Heroku
Zero Feature	Able to query then display some data	Client and basic placeholder GUI	Server, basic database queries, learning data APIs
Beta Feature	Basic forum features (alerts about routes and neighborhoods), searching route by name, map view of bus location	GUI, user state	Correct handling of client requests and database storage
Feature Complete	Location-bound alerts, automated transit-agency alerts, re-route detection, bus specific alerts and forums, favorites, help button	Adding features, testing and fixing	Adding features, testing and fixing
Release Candidate	No bugs, correct error handling and help information	Testing and fixing	Testing and fixing
Final	As usable as possible		

For the beta feature release, the delivered features listed above are the main features of our app. We will focus on the core functionality of our system first to ensure our releases satisfy the main requirements we have specified for our product. Then, once we have our main features working, we will focus on some of our more supplementary features, like starring favorite routes and neighborhoods, as well as our stretch features, like displaying reroute maps, and include these in our feature-complete release. By completing our main features early in the schedule, we will give ourselves more time to finish at least some of the other features we would like to offer our users.

4 Risk Summary

Testing our app in the real world is our biggest concern. Many concurrent test users would be ideal, but difficult to coordinate. We can force the back and front ends to interact with fixed data (by providing a specific location in the Android emulator for example), but at some point we will need to test the application with actual buses. To help mitigate this, we plan to develop basic debug interfaces to start testing the system early and often (such as when a group member would already be using the bus).

Coordinating between group members working on different ends is another concern. To reduce the risk of implementing incompatible interfaces that necessitate rewriting, we plan to focus on this aspect of development first: using dummy data where data would be needed from unimplemented components.

One source of risk is in implementing automated information collection from transit agencies. Automatically formatting this information and posting it as alerts is a feature we definitely want, but this information may not be uniform enough to be parsed by a computer. We plan to write one or more basic parsers for transit agency information early on to test the feasibility of this feature. In the worst case, we will cut it.

To help identify risks inherent in the chosen software tools and interfaces, we will focus some of our efforts on learning the tools group members are unfamiliar with while working on the Software Design Specification. In particular, the front end team will be familiarizing themselves with Android Studio, the back end team with Ruby on Rails and Heroku.

As additional risk assessment, we will pursue external user evaluation at various points throughout our project. Before the zero feature release, we will perform paper prototyping with people who do not necessarily know computer science to ensure that general users can navigate our user interface. In addition, we will ask for user feedback after our beta feature release to ensure that users appreciate and understand the core functionality of our app, arguably the most important part of our system. Finally, we will ask users to test our system after the feature complete release to verify that our system works and potentially uncover any bugs.