MetroBike - Requirements Outline

Product Description

MetroBike is an Android app that provides travel directions for people who ride bicycles and take public transit. Therefore, the target audience of our product is anyone that wants to combine public transit and bicycling in order to minimize their travel time. The app is essentially a new “travel mode” option for Google Directions. Google Transit currently only provides public transit directions under the assumption that the user will be walking. This is not ideal for cyclists because they can travel faster than someone who is walking. Given starting and ending locations, such as two addresses, our app will find optimal routes based on the shortest travel time given some combination of taking public transit and riding a bike. Our app will enable cyclists to take advantage of their speed to find the bus routes that a pedestrian would not be able to make in time.

Software Toolset

Android Maps and Google Web Services APIs

We will be using the Android Maps API in order to draw the user and their current location on the map. We will also be using the Google Web Services API in order to fetch the bicyclist directions and the public transit directions. We chose to use these libraries because many people are familiar with Google’s map interfaces, and we figured this would make our application the easiest for them to use. Furthermore, using these APIs fulfills the data source requirement of the project. It would have also been impossible to complete this project in the given time frame if we had started from scratch.

Git and GitHub

About half of the team has a basic understanding of Git while one of the team members is an advanced user. The team is planning to keep the usage of Git relatively simple so that the team members that are less familiar with Git will not have as much ramping up to do. Only two of the team members have used GitHub before, so they will need to spend some time getting familiar with the tool.

Java

We chose to create the project in Java because everyone on the team is familiar with this language. There will be no ramping up on this.

Feature Set

The core feature of the application is, given starting and ending locations, finding an optimal route that combines public transit and bicycling directions provided by the Google Directions API. The optimal route will be displayed to the user as a map where the route is highlighted as well as in list form describing each step of the route. Another major feature is providing step-by-step directions that guide the user through the route as they travel. By default the directions will be calculated based the user leaving immediately. However, another feature will give the user the option of inputting a specific departure or arrival time when asking for directions. A minor feature of the app is returning multiple possible routes to the user so they can choose one based on an overview each route. Other minor features include a history list that saves recent requests for directions and advanced route-finding features that enable the user to set the maximum bicycling distance or the maximum number of bus transfers.

Profitability

We plan to make money on this app by having two versions, free and premium. The free version will have only the basic route finding capabilities while the premium version will allow the user to view their route finding history and set advanced route finding options. These advanced options would include features like setting a maximum bicycling distance or a maximum number of bus transfers. If the user tries to use one of the premium features they would be prompted to purchase the premium version of the app for a few dollars.

Alpha, Beta, V1 Breakdown

Alpha Version

For the Alpha version we plan to have a system that demonstrates that the pieces are in place for the major functionality of the app. The Alpha version of the app will allow the user to input starting and ending locations and then get back the Google Transit directions for the route. The directions will be visualized on a map and also in list form. We decided not to include any bicycling directions at this point because we want to focus on plotting a route using the Android Maps API given directions from the Google Directions API. We believe this will be a significant part of the coding work and will set us up well to incorporate bicycle directions in the Beta version. For this version we will have started implementing the feature of displaying the optimal route but it will not be complete due to a lack of incorporating bicycle directions.

Beta Version

For the Beta version we will add in functionality to incorporate bicycle directions from the Google Directions API. We will first develop the most basic route-finding algorithm which simply replaces the walking parts of the transit directions returned by the Google Directions API with bicycling directions by making one or more calls to the Google Directions API. We also plan to fully implement the step-by-step directions feature to walk the user through the route. Finally, we will also complete the history feature that allows the user to view routes searched for previously. We chose to implement the most basic route-finding algorithm and the other major user interaction features before working on making a more sophisticated algorithm because there is less risk involved. In summary, for the Beta release we will have completed two out of the three major features and one of the minor features.

Version 1.0

For Version 1.0 of our app we will have implemented all of the major and minor features of the app. The last major feature that we will complete during this phase is allowing the user to select a departure or arrival time when searching for directions. One of the minor features that will be completed is the ability for the user to see multiple route possibilities after searching for directions. The other minor features we plan to complete is giving the user the ability to limit the total bicycling distance and set the maximum number of bus transfers. Finally, we plan to work on implementing at least one sophisticated route-finding algorithm that enables the app to potentially find faster routes. We decided to work on improving the algorithm last because it is difficult to predict if it will substantially improve route-finding and we feel it is most important to have a fully functional app before working on optimization.

Risks, Cuts, Adjustments

If we are short on time the features we plan to cut is either the history feature or adding the options that enable the user to limit the maximum bicycling distance and number of bus transfers. We will decide which to cut based on our assessment of how difficult each one is to implement. We can also cut out doing work optimizing the route-finding algorithm. One idea we have for optimizing the route is replacing one or more legs of routes that involve taking multiple buses with bicycling directions instead of busing to check if it would be faster. We will pursue this implementation or explore other possible algorithms to further optimize route-finding only if time permits it. Another idea we have if we are falling behind schedule is to put less of an emphasis on the UI. We can do this by moving one or two people off the UI team and having them put 100 percent of their effort towards the area that is the most at risk for failure.

Group Dynamics

Jaylen VanOrden will be the Project Manager for MetroBike. We have decided to split the work up into 4 parts. They are:

- UI
- backend
- algorithm
- documentation

Each piece of project has a team leader that is responsible for making sure that they have a deliverable and that it is production quality. Some of the team members will work on multiple sub-teams based on their workloads. Some of the minor tasks like updating the wiki are assigned to individual team members. The documentation group changes based on the phase and the workload of the respective members. Given this, there is no official team lead, but team members will be assigned tasks that they will be responsible finishing. The current group breakdown/tasks is as follows:

- Jaylen VanOrden: Project Manager, Backend Lead
- Corey Harmon: Wiki-Posts, Backend
- Adrian Laurenzi: Algorithm Lead
- Qinyuan Wan: Backend
- Sam Wilson: UI
- Shuo Wang: UI

- Xinyun Chen: UI Lead

These roles were chosen by the individuals during our Tuesday meetings. If a disagreement arises, the involved team members should try and work through it by themselves. If they are unable to reach a consensus or resolve the conflict, they should then involve the PM. If the disagreement involves the PM and a team member they should try and work through it together. If they are unable to come to a conclusion then they should bring it up with the TA or professor.