Iteration 2 Report: CTFastTrak Live Map and Routes

Software Engineering CTFastTrak API Project
Bryan Davis, Robert Rotaru, Matthew Shafran, Brian Tardiff

Functionality

In the second iteration, our team was able to fully complete all of the goals set out during the planning of this iteration. The system has all of the functionality described in the user stories we planned to implement for Iteration 2. The system is able to determine the nearest bus and bus terminal location to the traveler on a specified bus route. The traveler is able to set both a departure location and a destination location, which is saved into our system and used to create travel plans and routing. The GTFS Interface accesses and processes all of the CSV format bus data, bus terminal data, bus route data, and any extraneous meta-data in order to provide JSON objects that are digested by our system to create a map of live CTFastrak information. Finally, the system is able to calculate and display a route for the traveler to take along a single bus line based on a destination that the traveler chooses and his or her origin point.

Implemented User Stories

2. As a traveler, I want to access the application's 'get nearest location' feature and have the system determine the bus location nearest to me so that I can plan my route accordingly.
Pre-condition: BRT System is accessible; Traveler is logged into the system; Locations are available
Post-condition: Traveler is able to retrieve nearest stop location

3. As a traveler, I want to access the application's 'set destination' feature and specify a destination I would like to travel to so that the system can accommodate my travel plans.
Pre-condition: BRT System is accessible; Traveler is logged into the system; Locations are pre-loaded into the system
Post-condition: Traveler is able to set a destination

4a. As a traveler, I want to access the application's 'pick route' feature and select from a list of calculated routes from point A to point B along the same bus line that the application has generated so that I can pick my preferred route to facilitate my travel needs.
Pre-condition: BRT System is accessible; Traveler is logged into the system; Routes have been calculated based on traveler's input
Post-condition: Traveler is able to pick a route

8. As the GTFS data Interface, I want to get the most up to date location of the bus terminals, so when the traveler requests a bus terminal location, I will be able to give accurate and relevant data.
Pre-condition: CTFastTrack provides bus terminal information in GTFS format
Post-condition: Bus terminal data gets sent to user

9. As the GTFS data interface, I want to get the most up to date location of the buses so that I can respond to the traveler's request for bus locations.
Pre-condition: CTFastTrack provides bus information in GTFS format
Post-condition: Bus data gets sent to traveler

User Story Changes
There were no user story changes for this iteration. Everything went smoothly as planned.

Lessons Learned
When working with the Google Maps Distance Matrix API we learned that we would not be able to query all stops without going over the free tier. In order to get around this, we had to calculate the distance from the route start coordinates to each bus stop. Then we only picked bus stops within a certain distance and sent those stops to the Matrix API in order to calculate how long it would take to walk to the bus stop. The other issue was that the Matrix API was only accurate down to the minute. This means that multiple stops could have the same time to get there, even though one is closer than the other. In order to compensate for this, we had to compare the distances and the times.

Remaining User Stories
4b. As a traveler, I want to access the application's 'pick route' feature and select from a list of calculated routes from point A to point B across multiple bus lines that the application has generated so that I can pick my preferred route to facilitate my travel needs.
Pre-condition: BRT System is accessible; Traveler is logged into the system; Routes have been calculated based on traveler's input
Post-condition: Traveler is able to pick a route

7. As a traveler, I want to be able to receive any updated information about the bus arrival times, ensuring that the data shown is as accurate as possible. The notifications will update any important changes (i.e. a delayed bus). This will help be alerted as soon as changes occur and be able to plan a better route.
Pre-condition: System is online and reachable. User is connected to the system. The user is either allowing automatic notifications or requests an update of JSON data.
Post-condition: The user will receive all up to date information from the JSON data, including any changes to route information.

10. As the JSON interface, I want to get live event data relayed from the CTFastrak API. I want to have this data as soon as it exists so that the user interface can be updated to alert a traveler via notifications of any route conditions or events
Pre-condition: JSON API is available. Interface is able to access JSON data.
Post-condition: BRT System listens and receives live event and delay information. This information is displayed in the user interface.

User Story Estimates

1	2	3	5	8
			#4b	
			#7	
			#10	

Next Iteration User Stories (to be implemented)

For our next iteration, we plan on completing our final three user stories, which add up to 15 points out of our original total of 45 (one third). The user stories to be implemented in Iteration 3 are #4b, #7, and #10. For user stories 7 and 10, our pre-conditions are that we have access to the live JSON data from our JSON interface, which we have implemented in our first iteration. We want to display a lot of available additional data provided by CTFastrak on the map, and user story 7 is to display live bus information, such as bus arrival times for bus terminals. User story 10 is similar, it is to display delay information for bus routes and conditions. Finally, we want to complete the routing feature we began implementing in Iteration 2. User story 4b is the second half of the routing feature, allowing us to route a traveler from any point on the map, through several bus lines, to their final destination.