

System Requirements – CTfastrak Application

TEAM V.T.D.

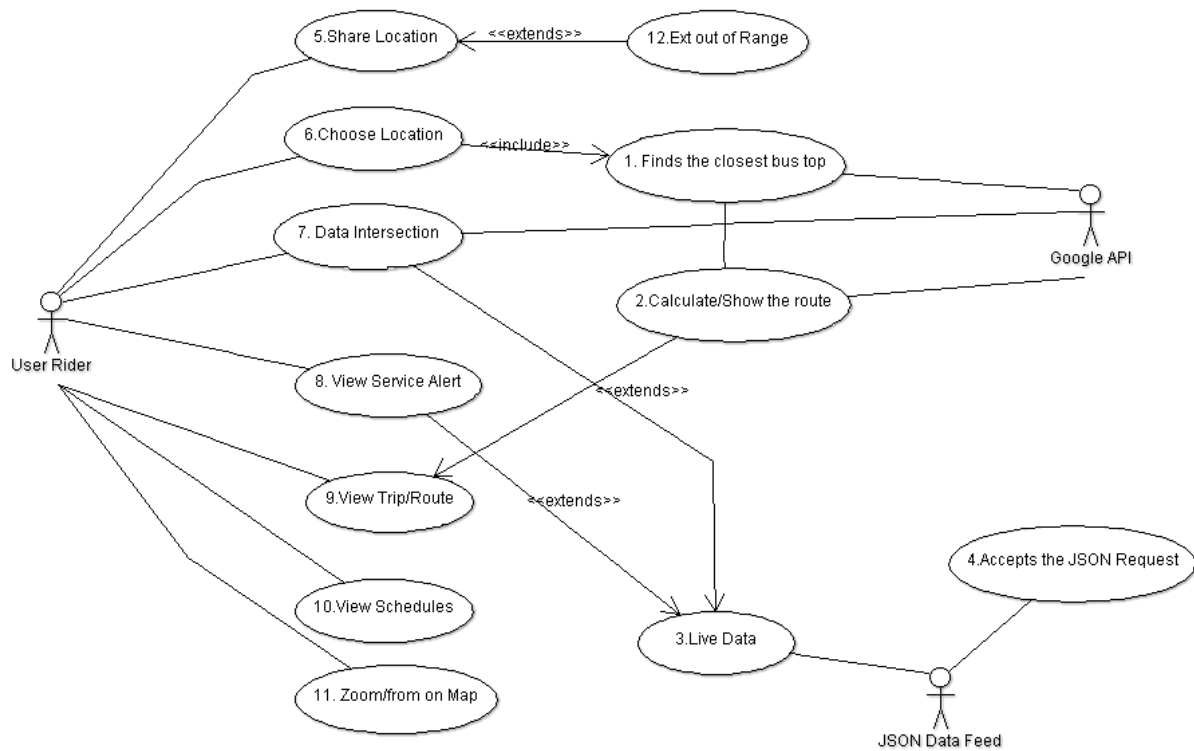
Functional Requirements

(1) Upon initialization, the application checks if the user's device is sharing its location. (2) If so, the application will test whether the location is within a reasonable distance of any CTfastrak bus stop. It does this by comparing the location coordinates to a bounding polygon that encapsulates the transit system area with a one mile buffer. (3) If the location is within the bounding polygon, the application prompts the user to accept it as their starting point or not. (4) If the user accepts, the application sets the user location as their starting point. (5) If the user rejects their location as their starting point, the user is outside of the bounding polygon or the user is not sharing their location, the application will not set a starting point at this time. (6) At any time, the user may choose a starting point by sharing their current location (if within the bounding polygon), picking a point on the map or entering an address. The map has a fixed extent so that the user may only select points on the map within a certain range (beyond the aforementioned bounding polygon) of the transit system. The latitude and longitude of an address entered are compared to this range and only accepted if they are within it. The user is notified when an address has been rejected. (7) A destination is chosen by picking a point on the map or entering an address. The same rules for accepting a starting point apply to accepting a destination. (8) Once a starting point and destination are set, the user submits them to the application which updates the map to emphasize the starting location and destination as well as the bus stops that are nearest to them and the CTfastrak route that connects those bus stops.

(9) In addition to the trip planning mode outlined above, the user may explore the map to get real-time information about the CTfastrak transit system at any time. All buses, routes and bus stops are displayed as map objects. The bus locations will be updated every thirty seconds to reflect their real world positions. By clicking on a map object, the user is presented with available data for the corresponding system component.

(10) In addition to real-time data, links to static documents that show planned bus schedules will be available.

Use Case Scenario Diagram



User Stories

[1] Find Closest Bus Stop

Scenario: The Google API finds the closest bus stop for the current location.

User Role: The Google API which is integrated into the application

Goal: Google API that is integrated into our application communicates with its map service by processing the rider's location query and finds the closest bus stop.

Benefit: Our Application use this Google API to take advantage of or extend the functionality of the existing map services that is capable of finding the closest bus stop.

Pre-Condition/ Post-Condition: Rider has shared location or entered destination are pre-Condition and there is no post-condition.

Further Breakdown: The user story does not need any further breakdown.

[2] Show/Calculate Route

Scenario: The Google API calculates and show the route for the current location towards the closest bus stop.

User Role: The Google API which is integrated into the application

Goal: Google API that is integrated into our application communicates with its map service by processing the rider's location query to find the closest bus stop and also to display and calculate the route.

Benefit: Our Application use this Google API to take advantage of or extend the functionality of the existing map services which is capable of showing and calculating the route from the current given location to the closest bus stop for the riders.

Pre-Condition/ Post-Condition: Riders have shared their location or destination and the Google API has found the closest bus stop are pre-Condition and there is no post-condition.

Further Breakdown: This story could be broken down to smaller steps for the route calculation

[3] Provide Live Data

Scenario: The JSON provide live data feed on the CTfastrak bus service's

User Role: The JSON Live data feed that is integrated into our application.

Goal: JSON feed that is integrated into our application to provide live data about the CTfastrak bus routes.

Benefit: Our application use this external JSON data feed to provide CTfastrak bus points or information on the map for the rider's.

Pre-Condition/ Post-Condition: The Application should have done JSON request for when the rider's request for view bus information are pre-Condition and there is no post-condition

Further Breakdown: This story does not need any further break down.

[4] Accept JSON Request

Scenario: JSON data feed accepts the request from our application for user features like viewing bus route.

User Role: The JSON Live data feed which is integrated into our application.

Goal: Allow the JSON request into our application to further retrieve the live data feeds.

Benefit: Our application use this external JSON data feed to provide CTfastrak bus points or information on the map for the riders.

Pre-Condition/ Post-Condition: Riders should have opted to view bus information through our application are pre-Condition and there is no post-condition.

Further Breakdown: This story could be broken down to receive request and process request

[5] Share Location

Scenario: The user shares their current location.

User Role: Riders of CTfastrak who are using this application.

Goal: The rider's location is necessary for the application to provide detailed information, route estimates, and the location of the nearest bus stop. With share location the user can automatically share GPS/location data.

Benefit: The rider benefits from getting personalized and more detailed information when they provide their location. This includes the ability to find the closest bus stop and view trip/route information.

Pre-Condition/ Post-Condition: There are no pre-conditions. The ability to view find the closest bus stop and view trip/route information are both post-conditions.

Further Breakdown: This user story may be broken down into the methods of input. Depending on the variety and differences between the input types it may be recommended to split this user story.

[6] Choose Location

Scenario: The user selects a location.

User Role: Riders of CTfastrak who are using this application.

Goal: The rider's choice in location is necessary to plan the route for any trip. This story is required for the destination and may be required for the user's current location.

Benefit: The rider benefits with detailed route information such as transfers, distance, fare, and the closest bus stop to the destination.

Pre-Condition/ Post-Condition: The pre-condition is to not be out of range. The ability to find the closest bus stop and view trip/route information are both post-conditions.

Further Breakdown: This user story may also be broken down by method of input. It may be necessary to split this story depending on the types of input.

[7] Data Intersection

Scenario: The rider wants to view real-time attributes of a CTfastrak map object.

User Role: CTfastrak rider, Mapping API, real-time data service

Goal: Display a formatted text area that contains the latest data available for a given a CTfastrak map object.

Benefit: The rider is kept informed of the status of the CTfastrak transit system.

Pre-Condition/ Post-Condition: The rider has selected on a CTfastrak map object. The post conditions are that the rider has been provided the latest information concerning the map object of interest and the display has changed to show this information.

Further Breakdown: This user story relates all the actors in this system and can be broken down as such; the rider who is querying the system, the real-time data service which receives requests and returns data about CTfastrak, and the mapping API which bridges the other two by interpreting the rider's map click as coordinates to be used when querying the real-time data service and formatting the real-time data that is returned.

[8] View Service Alerts

Scenario: The user wants to view the detailed service alert.

User Role: Riders of CTfastrak who are using this application.

Goal: Allow users to view news that may affect service of the CTfastrak line.

Benefit: Users can adjust their plans if the service alert affects their trip.

Pre-Condition/ Post-Condition: Accept JSON request and provide live data are pre-conditions. No post-conditions are required.

Further Breakdown: This story should not require further breakdown.

[9] View Trip/Route

Scenario: The user can view trip information and a representation of the route on the displayed map.

User Role: Riders of CTfastrak who are using this application.

Goal: Gives user detailed trip information and provide a visual representation of the route.

Benefit: Users can see a visualization of the route, distance, direction, and detailed information about their planned trip.

Pre-Condition/ Post-Condition: Share location, choose location, find closest bus stop, and show/calculate route are all pre-conditions. No post-conditions are necessary.

Further Breakdown: This story is simple enough that further break-down would lead to defining the implementation.

[10] View Schedule

Scenario: The user wants to view the static bus schedules.

User Role: Riders of CTfastrak who are using this application.

Goal: Allow easy access to scheduling information pertinent to the users input.

Benefit: Users are given a convenient link to scheduling information without the need to search for a link.

Pre-Condition/ Post-Condition: View bus information is a pre-condition. There are no post-conditions.

Further Breakdown: This story is simple enough that further break-down would go beyond the usefulness of a user story.

[11] Zoom/Pan Map

Scenario: The user wants the display to show a different map extent.

User Role: CTfastrak rider, Mapping API

Goal: Update the map to show a different extent.

Benefit: The rider has control over how much map detail is displayed.

Pre-Condition/ Post-Condition: There are no pre-conditions. The post-condition is that the map has changed.

Further Breakdown: Zoom and Pan are not considered as two separate functions of the map because technically speaking they are doing the same thing which is requesting new map image tiles.

[12] Out of Range

Scenario: The user's coordinates (through GPS or geocoding) are outside of the area of interest.

User Role: CTfastrak rider, Mapping API

Goal: Reject as input a location that is not within a reasonable distance from the CTfastrak transit system.

Benefit: The system is restricted to realistic implementations. The CTfastrak transit system remains the focus of the map.

Pre-Condition/ Post-Condition: The pre-conditions are that the rider is either offering their current location as input or has entered an address and that the location or address are not near the CTfastrak area. The post-condition is that the trip variable (starting point or destination) that was being manipulated by the rider has not changed. The rider is informed of invalid input.

Further Breakdown: This user story is an extension of [6] Choose Location and may be split into a starting point scenario and a destination scenario but the workflow is the same for each.

Non-functional Requirements

- The application must be intuitive.
- The application must be easy to use.
- The application must not be intrusive.
- The application must be fast.
- The application must be reliable.

- The application must be responsive.
- The application must be robust.
- The application must be mobile friendly.
- The application must protect privacy.
- The application must be secure.
- The application must follow legal documentation.

Glossary

API – Application Programming Interface. A set of programming tools that extend the functionality of applications such as this one.

CTfastrak - An express bus line system in central Connecticut.

Geocoding – The process of converting an address into a point on a map.

GPS – Global Positioning System. A constellation of satellites that locate receivers on the Earth.

JSON - A.K.A JavaScript Object Notation is a data object used to transmit live data.

Map Service – A cartographic service provided by Google API.

Riders - Users of the CTfastrak express bus line.

Service alert - News relevant to CTfastrak riders provided by a JSON.

Static - Information that does not change regularly.