

MyMBTA Software Requirements Specification

Version 2.0

December 3, 2012

Project Team: Leighann Astolfi

Ryan Bigelow

Jeff Guion

Nick Labich

Contributors: Dr. Jessica Young Schmidt

Students of CS4500

1. Introduction

1.1. Purpose

The purpose of this document is to provide the requirements for a software that uses MBTA live data to plan trips on the MBTA subway lines and find the current location of all available trains in service. This document provides the overall and specific functionalities from the system and user perspective that should be included in the software.

1.2. Scope

This document contains a complete list of all functionality that needs to be included in the MyMBTA software. Overall product functions are listed by use case in Section 2.1 of this document. More specific requirements are provided in Section 3.2 of this document. The constraints and additional details can also be found in Sections 2 and 3.

1.3. Definitions, acronyms, and abbreviations

MBTA: Massachusetts Bay Transportation Authority

JSON: JavaScript Object Notation

2. Overall description

This software will use MBTA live data to display the current location of active trains. This software will display ways to get from one stop to another, providing maps and directions. Only the MBTA blue, orange, and red line train data will be available in this software. This software will not provide information about the MBTA green line train, bus schedules, or the commuter rail.

2.1 Product functions

This subsection provides the overall functionalities of the system based on use cases.

2.1.1. The user wants to know where she can go using the T.

Priority: Essential

1. The user launches the application.
2. The system displays a map of train lines.

2.1.2. The user wants to know the current location of all trains.

Priority: Essential

1. The user launches the application.
2. The system requests current JSON data.
3. The system displays a map of the train line showing the current location of each train in service.
4. The user hovers the mouse pointer over a train.
5. The system displays the time in seconds until the train reaches the next station.

2.1.3. The user wants to know when the next trains get to stop A.

Priority: Essential

1. The user selects the "Station Mode" option from the main screen.
2. The user clicks on stop A.
3. The system requests current JSON data.
4. The system displays at most five incoming trains in each direction to a station.

2.1.4. The user wants to know her options for getting from stop A to stop B.

Priority: Essential

1. The user selects "Route Mode".
2. The system displays a route menu.
3. The user clicks on a starting station and ending destination.
4. The system requests current JSON data.
5. The system displays the fastest route, unless an alternate route option is selected.

2.1.5. The user wants to know her options for getting to an ordered list of stops.

Priority: Desirable

1. The user selects "Route Mode".
2. The user adds stops to a route.
3. The system requests current JSON data.
4. The system displays the fastest route, unless an alternate route option is selected, to get to each stop in the order they were entered.

2.1.6. The user wants to know her options for getting to an unordered list of stops.

Priority: Optional

1. The user selects "Unordered Route Mode".
2. The user adds stops to a route.
3. The system displays "Set As Start Station" and "Set As End Station" fields.
4. The user does not select a start or end stations.
5. The system requests current JSON data.
6. The system displays the best logical route (fastest, unless other route option selected) through all of the stops.

2.1.7. The user wants to know her options for getting to an unordered list of stops with specified starting and/or ending points.

Priority: Optional

1. The user selects "Unordered Route Mode".
2. The user adds stops to a route.
3. The system displays "Set As Start Station" and "Set As End Station" fields.
4. The user selects a station from the list of added stations and clicks "Set As Start Station".
5. The user selects a station from the list of added stations and clicks "Set As End Station".
6. The system requests current JSON data.
7. The system displays the best logical route (fastest, unless other route option selected) through all the stops from start destination to end destination.

2.1.8. For any trip on the T, the user wants the option to specify departure and/or arrival times.

Priority: Desirable

1. The user selects "Route Mode" or "Unordered Route Mode".
2. The user adds stops to a route.
3. The system displays a "Time of Trip" drop down.
4. The user selects "Arrive By" or "Depart At" option.
5. The user enters desired arrival or departure time.
6. The system requests current JSON data.
7. The system displays the fastest route (unless other route option selected) ending or starting at the specified time.

2.1.9. For any trip on the T, the user wants to know the fastest route, the earliest departure, the earliest arrival, and fewest transfers.

Priority: Desirable

1. The user selects "Route Mode" or "Unordered Route Mode".
2. The user adds stops to a route.
3. The system displays a "Route Options" drop down.
4. The user selects desired route option.
5. The system requests current JSON data.
6. The system displays the a route according to the desired route option.

2.1.10. The client wants to test the system with old data.

Priority: Essential

1. The user selects the "Test" radio button.
2. The system loads the original test data for all three lines.
3. The user clicks on one of the three color buttons to override that line's default test data.
4. The system displays a "Test File" information window.
5. The user enters a test file path.
6. The system loads the data from the test files.
7. The user can perform all functionality from Requirements 2.1.1-2.1.9 using test data.

2.2. User characteristics

The users of this system will be any user of the MBTA ranging from computer novice to computer expert with varying ages.

2.3 Constraints

1. The system shall be able to run on CCIS Linux machine.
2. The system shall be based on live MBTA data from <http://developer.mbta.com/lib/rthr/{{line}}.json>.
3. The system shall also be able to accept test data as input in order to facilitate the testing of the software.

2.4. Assumptions and dependencies

The system will be dependent upon the MBTA live data provided by the MBTA.

2.6 Apportioning of requirements

The requirements contained in Section 2 are higher-level requirements, while Section 3 contains the more detailed requirements. These requirements will be the basis for the design and implementation of the system. Each requirement has one of the following priorities: essential, desirable, and optional. The requirements will be implemented based on these priorities with the essential requirements implemented first, followed by the desirable requirements, and then the optional requirements.

3. Specific requirements

3.1. User Interfaces

The system should have a user friendly GUI for any technical skill level. It should be easy to understand and simple to use. The maps and directions should be interactive to help users get the most out of the software.

3.2. Functional requirements

This section provides specific details for the functionalities by use case from Section 2.1.

3.2.1. The user wants to know where she can go using the T.

Priority: Essential

1. The user launches the application.
 - DR 3.2.1.1.1. Main screen shall display a map when the user launches the application.
 - DR 3.2.1.1.2. The user shall be able to launch the application when she wants to see where she can go on the T.
2. The system displays a map of train lines.
 - DR 3.2.1.2.1. The system shall load a map of the Blue, Orange, and Red lines.
 - DR 3.2.1.2.2. The map shall have a white square representing each station on the appropriate line, in the order the station is on the line.

3.2.2. The user wants to know the current location of all trains.

Priority: Essential

1. The user launches the application.
 - DR 3.2.2.1.1. The user shall be able to launch the application option when she wants to see the current location of all trains.
 - DR 3.2.2.1.2. The main screen shall display a map when the user launches the application.
2. The system requests current JSON data.
 - DR 3.2.2.2.1. The system shall use Jackson to parse the incoming JSON from the live MBTA data or from local test file.
3. The system displays a map of the train line showing the current location of each train in service.
 - DR 3.2.2.3.1. The system shall have a train line map.
 - DR 3.2.2.3.2. There should be an individual train line map for each T line.
 - DR 3.2.2.3.3. The train line map shall be a straight line map with each stop labeled.
 - DR 3.2.2.3.4. The system shall use the JSON data to locate the current location of each train on all lines.
 - DR 3.2.2.3.5. The system shall display black squares at the current location of train on the selected train line map.
 - DR 3.2.2.3.6. The system shall load all train line maps and display them vertically: Orange Line on top, then Red Line, and Blue Line at the bottom.
4. The user hovers the mouse pointer over a train.
 - DR 3.2.2.4.1. The user shall be able to mouse over the black squares representing trains when she wants to see the time until that train reaches the next station.
5. The system displays the time until the train reaches the next station.
 - DR 3.2.2.5.1. The system shall display a tool tip when the user moves the mouse over a black square.
 - DR 3.2.2.5.2. The tool tip shall contain the name of the next station in which the train will arrive.
 - DR 3.2.2.5.3. The tool tip shall contain the time in minutes and seconds until it reaches the next station.

3.2.3. The user wants to know when the next trains get to stop A.

Priority: Essential

1. The user selects the "Station Mode" option from the main screen.
 - DR 3.2.3.1.1. The system shall have a "Station Mode" option.
 - DR 3.2.3.1.2. The user shall be able to select the "Station Mode" option when she wants to see when the next trains get to stop A.
 - DR 3.2.3.1.3. The system shall fill in the radio button when the option is selected by the user.
2. The user selects a stop A.
 - DR 3.2.3.2.1. When in Station mode, the user shall be able to click on a Station on the map when she wants to see incoming trains to the selected Station.

3. The system requests current JSON data.
-See DR 3.2.2.2.1
4. The system displays the incoming trains to a station.
 - DR 3.2.3.4.1. The system shall display an information window when the user clicks on a Station in Station mode.
 - DR 3.2.3.4.2. The information window shall contain the time in minutes and seconds until the next train arrives at the selected Station in both directions.
 - DR 3.2.3.4.3. The information window shall contain the direction in which the next train is travelling in each direction.
 - DR 3.2.3.4.4. The information window shall display the time in minutes of up to 4 additional trains going in both directions.

3.2.4. The user wants to know her options for getting from stop A to stop B.

Priority: Essential

1. The user selects "Route Mode".
 - DR 3.2.4.1.1. The main screen shall have a "Route Mode" radio button.
 - DR 3.2.4.1.2. The user shall be able to select the "Route Mode" option when she wants to find out how to get from A to B.
 - DR 3.2.4.1.3. The system shall fill in the radio button when the option is selected by the user.
2. The system displays a route menu.
 - DR 3.2.4.2.1. The system shall have a menu on the right side.
 - DR 3.2.4.2.2. The system shall have an "Add/Update" button.
 - DR 3.2.4.2.3. The system shall have an "Clear" button.
 - DR 3.2.4.2.4. The system shall have a "Get Directions" button.
3. The user clicks on a starting station and ending destination.
 - DR 3.2.4.3.1. The user shall be able to click on a Station when in Route mode.
 - DR 3.2.4.3.2. The system shall highlight the selected Station in yellow when in Route mode.
 - DR 3.2.4.3.3. The user shall be able to click "Add/Update" when she wants to add a selected stop to a route.
 - DR 3.2.4.3.4. The user shall be able to select another Station and add it to the route.
 - DR 3.2.4.3.5. The system shall keep the last stop added highlighted yellow.
 - DR 3.2.4.3.6. The system shall add the name of the selected Stations to the route menu when the user clicks "Add/Update".
 - DR 3.2.4.3.7. The system shall clear the stops from the route menu when the user clicks the "Clear" button.
 - DR 3.2.4.3.8. The user shall be able to click the "Get Directions" link when she wants the system to find her trip.
4. The system requests current JSON data.
-See DR 3.2.2.4.1
5. The system displays the fastest route, unless an alternate route option is selected.
 - DR 3.2.4.5.1. The system loads the fastest route, unless an alternate route option is selected, when the user selects "Get Directions".
 - DR 3.2.4.5.2. The system displays a directions information window.

- DR 3.2.4.5.3. The system displays the time in minutes and seconds when the train leaves the first added Station in the information window.
- DR 3.2.4.5.4. The system displays the time in minutes and seconds when the train arrives at the selected destination in the information window.

3.2.5. The user wants to know her options for getting to an ordered list of stops.

Priority: Desired

1. The user selects "Route Mode".
 - DR 3.2.5.1.1. The user shall be able to select the "Route Mode" option when she wants to find out how to get to a list of stops.
 - See DR 3.2.4.1.1,3
2. The user adds stops to a route.
 - DR 3.2.5.2.1. The users adds stops to a route as in DR 3.2.4.3.1 - 4.
3. The system requests current JSON data.
 - See DR 3.2.2.4.1
4. The system displays the fastest route, unless an alternate route option is selected, to get to each stop in the order they were entered.
 - DR 3.2.5.4.1. The system shall load the fastest route through all of the stops in the order the user has entered them, unless an alternate route option is selected, when the user clicks "Get Directions".
 - See DR 3.2.4.5.2 - 4.
 - DR 3.2.5.4.2. The system shall repeat 3.2.4.5.3,4 for each remaining stop on the route.

3.2.6. The user wants to know her options for getting to an unordered list of stops.

Priority: Optional

1. The user selects "Unordered Route Mode".
 - DR 3.2.6.1.1. The main screen shall have a "Unordered Route Mode" radio button.
 - DR 3.2.6.1.2. The user shall be able to select the "Unordered Route Mode" option when she wants to find out how to get to an unordered list of stops.
 - DR 3.2.6.1.3. The system shall fill in the "Unordered Route" radio button when the option is selected by the user.
2. The user adds stops to a route.
 - DR 3.2.6.2.1. The users adds stops to a route as in DR 3.2.4.3.1 - 4.
3. The system displays "Set As Start Station" and "Set As End Station" fields.
 - DR 3.2.6.3.1. The system shall display "Set As Start Station" and "Set As End Station" buttons below the "Add/Update" button.
4. The user does not select a start or end stations.
 - DR 3.2.6.4.1. The user shall not click either the "Set As Start Station" or "Set As End Station" buttons.
5. The system requests current JSON data.
 - See DR 3.2.2.4.1
6. The system displays the best logical route (fastest, unless other route option selected) through all of the stops.

- DR 3.2.6.6.1. The system shall load the fastest route, unless other route option selected, through all of the specified stops.
-See DR 3.2.4.5.2 - 4, DR 3.2.5.4.2.

3.2.7. The user wants to know her options for getting to an unordered list of stops with specified starting and/or ending points.

Priority: Optional

1. The user selects "Unordered Route Mode".
- See DR 3.2.6.1.1 - 3.
2. The user adds stops to a route.
DR 3.2.7.2.1. The users adds stops to a route as in DR 3.2.4.3.1 - 4.
3. The system displays "Set As Start Station" and "Set As End Station" fields.
- See DR 3.2.6.3.1.
4. The user selects a station from the list of added stations and clicks "Set As Start Station".
DR 3.2.7.4.1. The user selects her desired start station from the route menu and clicks the "Set As Start Station" button when she wants to set a start station.
DR 3.2.7.4.2. The system shall display "Start Station" next to the selected station in the route menu when the user clicks "Set As Start Station".
5. The user selects a station from the list of added stations and clicks "Set As End Station".
DR 3.2.7.5.1. The user selects her desired end station from the route menu and clicks the "Set As End Station" button when she wants to set an end station.
DR 3.2.7.5.2. The system shall display "End Station" next to the selected station in the route menu when the user clicks "Set As End Station".
6. The system requests current JSON data.
-See DR 3.2.2.4.1
7. The system displays the best logical route (fastest, unless other route option selected) through all the stops from start destination to end destination.
DR 3.2.7.7.1. The system shall find the best logical route (fastest, unless other route option selected) routes from Start to the End, or just from Start or to End, depending on user input, visiting each other stop using the most direct route.
-See DR 3.2.4.5.2 - 4, DR 3.2.5.4.2.

3.2.8. For any trip on the T, the user wants the option to specify departure and/or arrival times.

Priority: Desirable

1. The user selects "Route Mode" or "Unordered Route Mode".
- See DR 3.2.4.1.1, DR 3.2.4.1.3, DR 3.2.5.1.1 or DR 3.2.6.1.1-3.
2. The user adds stops to a route.
DR 3.2.8.2.1. The users adds stops to a route as in DR 3.2.4.3.1 - 4.
3. The system displays a "Time of Trip" drop down.
DR 3.2.8.3.1. The main screen shall have a "Time of Trip" drop down.
DR 3.2.8.3.2. The main screen shall have an incremental text box "Time" field containing hours and minutes defaulted to the current time.
DR 3.2.8.3.3. The "Time of Trip" drop down will contain the following options, defaulted to "Leave Now":

- Leave Now
 - Arrive By
 - Depart At
4. The user selects "Arrive By" or "Depart At" option.
 - DR 3.2.8.4.1. The user shall use the "Time of Trip" drop down to choose desired option when she wants to arrive by or depart at a certain time.
 - DR 3.2.8.4.2. The system shall display the option in the drop down field when selected by the user.
 5. The user enters desired arrival or departure time.
 - DR 3.2.8.5.1. The user will use the "Time" text box or incremental arrows to select the desired time of trip.
 6. The system requests current JSON data.
 - See DR 3.2.2.4.1
 7. The system displays the fastest route (unless other route option selected) ending or starting at the specified time.
 - DR 3.2.8.7.1. The system shall find the q fastest route (unless other route option selected) at the specified time.
 - DR 3.2.8.7.2. The system shall display the route with unknown times if the user selects "Depart At" and there is no data available to leave at the time specified by the user.
 - DR 3.2.8.7.3. The system shall display the message "No route can arrive at the specified time. Here is the closest route to your arrival time:" and display a route leaving now, if no route is available to arrive by the time specified by the user.
 - See DR 3.2.4.5.2 - 4, DR 3.2.5.4.2.
- 3.2.9. For any trip on the T, the user wants to know the fastest route, the earliest departure, the earliest arrival, and fewest transfers.**
- Priority:** Desirable
1. The user selects "Route Mode" or "Unordered Route Mode".
 - See DR 3.2.4.1.1, DR 3.2.4.1.3, DR 3.2.5.1.1 or DR 3.2.6.1.1-3.
 2. The user adds stops to a route.
 - DR 3.2.9.2.1. The users adds stops to a route as in DR 3.2.4.3.1 - 4.
 3. The system displays a "Route Options" drop down.
 - DR 3.2.9.3.1. The main screen shall have a "Route Options" drop down.
 - DR 3.2.9.3.2. The "Route Options" drop down shall contain the following options:
 - Fastest Route
 - Earliest Departure
 - Earliest Arrival
 - Fewest Transfers
 4. The user selects desired route option.
 - DR 3.2.9.4.1. The user shall use the "Route Options" drop down to select the desired property of her trip.
 5. The system requests current JSON data.
 - See DR 3.2.2.4.1
 6. The system displays the a route according to the desired route option.

- DR 3.2.9.6.1. The system shall find the route through the selected stops that best matches the user selected route option.
-See DR 3.2.4.5.2 - 4, DR 3.2.5.4.2.

3.2.10. The client wants to test the system with old data.

Priority: Essential

1. The user selects the "Test" radio button.
 - DR 3.2.10.1.1. The system shall have a "Test" radio button.
 - DR 3.2.10.1.2. The user shall be able to select the "Test" radio option when she wants to test the application with old data.
 - DR 3.2.10.1.3. The system shall fill in the "Test" radio button when the option is selected by the user.
2. The system loads the original test data for all three lines.
 - DR 3.2.10.2.1. The system shall not access the MBTA live data.
 - DR 3.2.10.2.2. The system shall use the original test files provided from the client by default.
3. The user clicks on one of the three color buttons to override that line's default test data.
 - DR 3.2.10.3.1. The system shall have three buttons marked "Orange", "Red", and "Blue".
 - DR 3.2.10.3.2. The user shall be able to click on one of the three buttons marked "Orange", "Red", and "Blue" when she wants to enter the path of a JSON file to override that line's default test data.
4. The system displays a "Test File" information window.
 - DR 3.2.10.4.1. The system shall display a "Test File" information window when the user clicks on one of the three buttons marked "Orange", "Red", and "Blue".
 - DR 3.2.10.4.2. The "Test File" information shall have a text field.
5. The user enters a test file path.
 - DR 3.2.10.5.1. The user shall be able to enter the fully qualified path of a JSON file to override that line's default test data in the "Test File" window.
 - DR 3.2.10.5.2. The JSON files must be in the same format as the MBTA live data.
 - DR 3.2.10.5.3. The JSON files should contain valid data.
6. The system loads the data from the test files.
 - DR 3.2.10.6.1. The system shall not access the MBTA live data.
 - DR 3.2.10.6.2. The system shall only query the data from the files specified by user or the default files from the client.
 - DR 3.2.10.6.3. The system shall be able to interpret the old data JSON files just as it would current MBTA live data.
7. The user can perform all functionality from Requirements 2.1.1-2.1.9 using test data.
 - DR 3.2.10.7.1. The user shall be able to use the software with the test data files.

3.3. Performance requirements

1. Each screen should load within 10 seconds.
2. User selection that requires new fields to be displayed should be done so within 10 seconds.
3. Maps and directions should also be loaded within 10 seconds.

3.4. Design constraints

1. The system shall be created using Java.
2. The system shall use Jackson to interpret the MBTA JSON data.
3. The system should only query the MBTA data once every 10 seconds to avoid losing access to the data.
4. The GUI should be created using only the core Java libraries, including Swing.

4. Document Revision History

Version	0.1
Name(s)	Leighann Astolfi
Date	September 27, 2012
Change Description	Created original document from client user stories.

Version	1.0
Name(s)	Leighann Astolfi, Ryan Bigelow, Jeff Guion, Nick Labich
Date	October 22, 2012
Change Description	Revised original document with comments from Dr. Schmidt. Added, removed, and modified after discussion with project team.

Version	2.0
Name(s)	Leighann Astolfi, Ryan Bigelow, Jeff Guion, Nick Labich
Date	December 3, 2012
Change Description	Revised updated document with comments from Dr. Schmidt. Revised and updated based on decisions made during implementation to match the final implementation.