Requirements Document

Pinecrest People Mover

Members:

Ricardo Martinez

Maurice Pruna

Mentor:

Gabriela Wilson

Instructor:

Masoud Sadjadi

© 2014 by Senior Project – People Mover.

All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of Senior Project – People Mover.

Abstract

This document is divided into seven sections: Introduction, Description of Current System, Project Plan, Proposed System Requirements, Glossary, Appendix, and References. The introduction provides information in regards to the project. This information consists of explaining the problem definition, which includes the desire, expectation, and goal provided by the client, gives a description of the project background, and provides a list of definitions and acronyms used in the document.

The sections for the Requirements Document provide a description of any current system similar to our project, a proposition of the new system, the user requirements provided by the client, and any solutions that will aid in the development process. The project plan section provides an outline to the team personnel, the hardware and software resources needed for the project, and the work breakdown consisting of identifying tasks, milestones, and deliverables. The proposed system requirements consist of the scenarios and high-level requirements for the system. These requirements were agreed upon and specified by the client. The glossary contains any terms used in the system development of the project. The appendix consists of the complete list of use cases, UML diagrams, mockups and the diary of the team meetings. Lastly, the document will have a section for references.

Table of Contents

[1. Introduction 4](#_Toc405819088)

[1.1 Problem definition 4](#_Toc405819089)

[1.2 Scope of the System 4](#_Toc405819090)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc405819091)

[1.4 Overview of Document 4](#_Toc405819092)

[2. Description of Current System (Limitations and Constraints) 5](#_Toc405819093)

[3. Project Plan 6](#_Toc405819094)

[3.1 Project Organization 6](#_Toc405819095)

[3.1.1 Project Personnel Organization 6](#_Toc405819096)

[3.1.2 Hardware and Software Resources 7](#_Toc405819097)

[3.2 Identification of Tasks, Milestones and Deliverables (work breakdown) 8](#_Toc405819098)

[4. Proposed System Requirements 9](#_Toc405819099)

[4.1 Functional Requirements 9](#_Toc405819100)

[4.2 Analysis of System Requirements 12](#_Toc405819101)

[4.2.1 Scenarios 12](#_Toc405819102)

[4.2.2 Use case model 14](#_Toc405819103)

[4.2.3 Static model e.g., object diagrams, class diagram 14](#_Toc405819104)

[4.2.4 Dynamic model e.g., sequence diagrams or state machines 14](#_Toc405819105)

[5. Glossary 15](#_Toc405819106)

[6. Appendix 16](#_Toc405819107)

[6.1 Appendix A - Complete use cases 16](#_Toc405819108)

[6.2 Appendix B - Use case diagram using UML 24](#_Toc405819109)

[6.3 Appendix C - Static UML diagram 25](#_Toc405819110)

[6.4 Appendix D - Dynamic UML diagrams 34](#_Toc405819111)

[6.5 Appendix E - User Interface designs. 42](#_Toc405819112)

[6.6 Appendix F - Diary of meeting and tasks. 46](#_Toc405819113)

[7. References 54](#_Toc405819114)

# 1. Introduction

This section gives a brief idea about what is the problem presented by the community of Pinecrest using their trolley system called People Mover (PPM)

## Problem definition

Pinecrest government would like to design a Web tracker and a Mobile tracker to show residents routes, hours of operation, real-time trolley location (as a list and as an interactive map) and allow for automatic notifications for arrival at user’s favorite stops. The mobile tracker should work well on iPhone and android devices.

## Scope of the System

The Pinecrest People Mover is a free transit bus service operated by the Village of Pinecrest connecting their neighborhoods and schools. Middle and high school students who do not qualify for bus service from the school district mostly use it.

## Definitions, Acronyms, and Abbreviations

PPM: Pinecrest People Mover

SRS: Software Requirements Specification

UML: Unified Modeling Language

## Overview of Document

* Chapter 2: Describes the status of the current system.
* Chapter 3: Definition of the project plan, which consists of the project organization which assigns roles to the team members and the project schedule.
* Chapter 4: This chapter contains the proposed system requirements (definition of the scenarios, use case models, Static model, and Dynamic model).
* Chapter 5: Glossary with definitions of specific terms.
* Chapter 6: Appendix
* Chapter 7: References to other works.

# 2. Description of Current System

The village of Pinecrest has two mover routes that are currently operating as a free service (Palmetto Middle School and Palmetto Senior High School). Each one has a north and a south path. Pinecrest citizens mainly use the movers as alternative to get to the local schools. They have no way to know with precision the wait time for next mover while are waiting on a stop, or the nearest stop to their actual position. For that reason Pinecrest government is currently involved on the installation of GPS devices in the movers. A new system will be created to serve the clients desired purposes.

# 3. Project Plan

This chapter will explain the schedule for the development plan. The roles of each team member are outlined in this section including their assigned roles. The hardware and software resources needed for the development of this project are also provided. Tasks, milestones, and deliverables are also outlined.

## 3.1 Project Organization

The team implementing this project consists of two students. Being the fact that the team is small, each team member will require having multiple roles, which include project manager, developer, tester, etc. The team members will have to use the resources allocated to them and must be approved by the client in order for the development of project to be a success

### 3.1.1 Project Personnel Organization

The structure of the project organization can be considered hierarchal. Each student will be in constant communication with each other during the development process. The students will be presenting their progress to Masoud Sadjadi, the professor, and Gabriela Wilson, the client. Masoud Sadjadi will be providing feedback to Ricardo Martinez and Maurice Pruna in how to improve project.



|  |  |  |
| --- | --- | --- |
| Name | Roles | Contact |
| Gabriela Wilson | Client/Mentor | gwilson@pinecrest-fl.gov |
| Masoud Sadjadi | Professor | sadjadi@cs.fiu.edu |
| Ricardo Martinez | Front-End Developer, Tester | rmart071@fiu.edu |
| Maurice Pruna | Back-End Developer, Tester | mprun003@fiu.edu |

### 3.1.2 Hardware and Software Resources

**Required Hardware**

* iOS device
* Android device
* Server

**Required Software**

* Microsoft Word
* Sencha Touch
* Cacoo
* Mockup Builder
* Microsoft Project
* GitHub
* Trello
* SourceTree
* FIU Moodle

## 3.2 Identification of Tasks, Milestones and Deliverables (work breakdown)

|  |  |
| --- | --- |
| ID | Task Name |
| 1 | Meet with Partner |
| 2 | Meet with mentor |
| 3 | Create high level requirements |
| 4 | UI Mockup Designs |
| 5 | Create Use Cases |
| 6 | Create Schedule |
| 7 | Use Case Diagrams |
| 8 | Requirements Document |
| 9 | Project Plan Document |
| 10 | Feasibility Study |
| 11 | Retrieval of client server and database information |
| 12 | Map layouts with routes and trolley tracking |
| 13 | Begin Implementations |
| 14 | Design Test Cases based on use case criteria |
| 15 | Present to client product |
| 16 | Present project to class |
|  |  |

|  |  |
| --- | --- |
| Milestones | Projected Date |
| Finalized Requirements Document | 10/5/14 |
| Finalized Design Document | 10/13/14 |
| Completion of Implementation | 11/28/14 |
| Testing Phase Completed | 12/4/14 |
| Final Deliverable | 12/11/14 |
| Provide finished product to client | 12/12/14 |

# 4. Proposed System Requirements

This chapter will contain the detailed requirements for the application. Also there will be information about the different scenarios, uses cases model, class diagram, and sequential diagram.

## 4.1 Functional Requirements

1. **PPM\_01: Display Hours of Operation**

The system shall provide the user with the information in regards to the hours of operation.

* Usability: The user will access this function in 1 click from home.
* Reliability: Low mean time to failure since this function would be one of the main buttons on the menu
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_02: Trolley Alerts**

The system shall provide the user with any updates in regards to the trolleys. This can be a malfunction to a trolley, delayed departure, and not-in-service for that day

* Usability: This data doesn’t require any action of the user to get activated.
* Reliability: Low mean time to failure as this functionality would be very beneficial to the users
* Performance: Notifications should be sent as soon as information is entered by admin
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_03: Estimated Time for a Favorite Stop**

The system shall provide the estimated time for a selected stop marked as favorite.

* Usability: The user will access this function in 1 click from home.
* Reliability: Low mean time to failure since this is a main functionality of the app
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_04: Set Favorite Stop**

The system shall allow the user set a stop as a favorite stop.

* Usability: The user will access this function in the list stop view.
* Reliability: Low mean time to failure as user should be able to see estimated time for arrivla
* Performance: Information would be available within 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_05: Remove Favorite Stop**

The system shall allow the user remove the favorite stops entered.

* Usability: Easy as using pressing a button. No training required
* Reliability: Low mean time to failure since user should be able to set and remove favorite stops.
* Performance: Information would be deleted within 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_06: List Stops**

The system shall provide the user a list of all the stops given a route and direction.

* Usability: No training required. Can be seen in two clicks from the home screen.
* Reliability: Low mean time to failure since this information is retrieved from the database.
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_07: Map Routes**

The system shall provide the user a map given a route and direction.

* Usability: No training required. App opens up to map view.
* Reliability: Low mean time to failure since this is one of the main functionalities of the app
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_08: Estimated Arrival Time given Stop**

The system shall provide the estimated time for a selected stop.

* Usability: The user will access this function in 1 click from home.
* Reliability: Low mean time to failure as this is a main functionality of the app.
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_09: Find Nearest Stop**

The system shall provide the nearest stop to the user’s geolocation.

* Usability: The user will access this function in 1 click from home.
* Reliability: Low mean time to failure since this depends on the information tracked by the GPS
* Performance: Data will load in less than 2 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_10: Arrived to Location Alert**

The system shall send an email notification to user once trolley has arrived or is near to arrive (given the time defined in for a favorite stop) to selected location

* Usability: No training required.
* Reliability: Low mean time to failure since this depends on the information tracked by the GPS
* Performance: Notification should be sent no more than 1 minute.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_11: Login**

The system shall check if the username and password of a user are correct and show favorite stops.

* Usability: No training required. Login will contain input boxes.
* Reliability: Low mean time to failure as this functionality will allow user many functionalities.
* Performance: Login should take no longer than 10 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_12: Register**

The system shall save the username and password of a user.

* Usability: No training required. Registration view will contain input boxes.
* Reliability: Low mean time to failure as this functionality allows system to have users.
* Performance: Registration should take no longer than 20 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_13: Request Stop**

The system shall provide the user with a form view allowing the user to request a stop. They system should send an email to the pertinent authority and reply with an alert box saying ‘Email Sent’. The Village of Pinecrest reviews routes twice a year (Aug and Jan) based on suggestions from interested residents. Please review route information regularly to see if we were able to accommodate your request’

* Usability: No Training required. The view will contain input boxes where the user can type in information.
* Reliability: Low mean time to failure as this functionality allows interaction from the user.
* Performance: Request should take no longer than 20 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_14: Report Problem**

The system shall provide the user with a form view allowing the user to report a problem. They system should send an email with the problem message in question and/or a picture documenting the problem to the pertinent authority and reply with an alert box saying ‘Email Sent’ .

* Usability: No Training required. The view will contain input boxes where the user can type in information.
* Reliability: Low mean time to failure as this functionality allows interaction from the user.
* Performance: Request should take no longer than 20 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

1. **PPM\_15: Send Feedback**

The system shall provide the user with a form view allowing the user to send feedback. They system should send an email with the feedback message in question to the pertinent authority and reply with an alert box saying ‘Email Sent’ .

* Usability: No Training required. The view will contain input boxes where the user can type in information.
* Reliability: Low mean time to failure as this functionality allows interaction from the user.
* Performance: Request should take no longer than 20 seconds.
* Supportability: This functionality should be supported by any internet browser, iOS and Android Devices

## 4.2 Analysis of System Requirements

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

### 4.2.1 Scenarios

1. **Scenario - PPM\_01: Display Hours of Operation**
   1. The user has downloaded the app and wants to check what time the Pinecrest People Mover is operating. The user clicks on the Service Time button located in a tab panel on the Main Menu. The system retrieves the information from the database and displays information on the interface. The user will not be able to see at what time the trolley is working.
2. **Scenario – PPM\_02: Trolley Alerts**
   1. The user has the app downloaded on the phone device and wants to verify that the trolleys are operating on schedule. The user clicks on the ‘More’ button and when system has redirected to another view, he/she presses the Trolley Alerts button. The system retrieves any information saved in the database and displays on the interface that the trolley for Palmetto High Route is not functioning.
   2. The system will send notifications to all users in the app indicating that the movers are out-of-service. When the user opens the app a message will appear indicating that there has been a problem with the movers. If the user has provided an email or cell phone, a notification will be sent to them ahead of time
3. **Scenario – PPM\_03: Estimated Time for Favorite Stop**
   1. The user has selected a certain stop. The system should provide the user with information about the closets bus with that stop in its route. Such information should include the ID number, route information, the mover’s next stop, and estimated time of arrival to the user’s selected stop. The estimated time of arrival should be in minute format.
4. **Scenario – PPM\_04: Set Favorite Stop**
   1. The user knows of someone using the north route for the Palmetto Senior High mover. Once the mover, has arrived to the Palmetto Senior High stop, the system will send a notification to the user indicating that the mover has arrived safely to the location. This will require email and cell phone information so the notifications can be sent.
5. **Scenario – PPM\_05: Remove Favorite Stop**
   1. The user wants to remove their current favorite stop from the list. They have realized that they no longer us that stop, so they decided to click on the favorites button on the home page. They click on their favorite stop and a view is showing, where an option appears and displays ‘Remove from Favorites’. The user selects the option and the stop no longer appears on the list.
6. **Scenario –PPM\_06: List Stops**
   1. The user is using the application and wants to see all the stops a mover will make in a certain route. In the menu there will be a ‘List route’ button which will direct the user to another page indicating to choose which destination, Palmetto Middle School or Palmetto High School. He/she then chooses the route. Once the route has been selected, a list of all the stops is now being shown to the user
7. **Scenario – PPM\_07: Map Routes**
   1. The user is using the application and clicks on the view map button. The system will direct the user to a site with a Google map implemented. This map will have four different colors, which will be used to identify the different routes, and a People Mover icon, which will represent the actual location of the bus. The map will be considered real-time by showing the current location of the mover, so if the mover is driving to the next stop, the icon will be moving as well.
8. **Scenario – PPM\_08: Estimated Arrival Time Given Stop**
   1. The user is currently looking at the stop list for a given route. The list will also display the estimated arrival time for each stop. The user now knows the status of the trolley and around what location the trolley is currently located.
9. **Scenario – PPM\_09: Find the Nearest Stop**
   1. The user is on the Pinecrest People Mover application and clicks on the ‘Near Me’ button. The system should use the GPS feature to determine the user’s location and direct the user to a pin indicating the closest stop to their current location. They system will provide the user with the address of the stop and its route information.
10. **Scenario – PPM\_10: Arrived to Location Alert**
    1. The user has selected their favorite stop and the menu for time of alert appears. This menu give you options to either get instant notification when the trolley has arrived and ranges to up to 15 minutes before trolley arrives. The user selects the instant notification. Once the trolley has arrived at the favorite spot, they receive a SMS text notification and email notification indicating the trolley has arrived.
11. **Scenario – PPM\_11: Login**
    1. The user has already registered for the app. They want the app to retrieve all of their information and allow alert functionalities for their favorite spots. They click on the ‘More’ button and select the Sign in button. They enter their username and password and now the system will allow the alerts functionalities and their selected favorite spots will now appear.
12. **Scenario – PPM\_12: Register**
    1. The user wants to have the alert functionalities and keep the list of the favorite spots. They click on the ‘More’ button and select the Register button. They will enter their username, password, email, and phone number. Once they have selected the register button they will receive an email confirmation and can now use that information to sign in.
13. **Scenario – PPM\_13: Request a Stop**
    1. The user has moved location but there is no stop for the trolley near by. He/she wants to send a request so when they click on the ‘More’ button, they then click on the request a stop button. They will fill out the needed fields and submit the request. They will then see a confirmation message.
14. **Scenario – PPM\_14: Report Problem**
    1. The user see a tree blocking the way. He/she wants to report the problem to the transportation department. He/she click on the ‘More’ button, they then click on the report problem button. They will fill out the needed fields and submit the report . They will then see a confirmation message.
15. **Scenario – PPM\_15: Send Feedback**
    1. The user wants to state how great the app is. He/she wants to send the feedback to the transportation department. He/she click on the ‘More’ button, they then click on the send feedback button. They will fill out the needed fields and submit the feedback . They will then see a confirmation message.

### 4.2.2 Use case model

This sections is geared to point out the different steps within our engineering process where there is interaction between the system and the different actors that use the system. It provides with an overall idea of what the system needs to meet the requirements. The use case model gives a high level outlook of the system.

Once finalized, this system will provide a solution for Pinecrest’s citizens to get information about the town trolley system. The Use Case Diagram shown in Appendix B provides with all the functions that an actor could perform to get the requirements that are part of the system. For this scheme we identified two actors and 15 use cases that will be implemented. The interaction between the actors and the uses is cases is defined as follow:

Actor 1 Regular User:

* **Display Hours of Operation**
* **Trolley Alerts**
* **List Stops**
* **Map Routes**
* **Estimated Arrival Time Given Stop**
* **Find the Nearest Stop**
* **Request a Stop**
* **Report Problem**
* **Send Feedback**

Actor 2 Registered User:

* **Display Hours of Operation**
* **Trolley Alerts**
* **Estimated Time for Favorite Stop**
* **Set Favorite Stop**
* **Remove Favorite Stop**
* **List Stops**
* **Map Routes**
* **Estimated Arrival Time Given Stop**
* **Find the Nearest Stop**
* **Arrived to Location Alert**
* **Login**
* **Register**
* **Request a Stop**
* **Report Problem**
* **Send Feedback**

### 4.2.3 Static model e.g., object diagrams, class diagram

A static model is a representation of the system itself. It doesn’t represent the time or the sequence of events involved on the system. The object diagram is geared to display a set of objects with attributes using the relation among them. By the other hand a class diagram show the overall structure of the system using representation of the classes, attributes and methods and the relation among the classes as well. All these diagrams are displayed on the appendix C of this document.

### 4.2.4 Dynamic model e.g., sequence diagrams or state machines

Dynamics models are a representation of the system taking to account the time. The most used ones are the sequence diagrams and states machines. The sequence diagrams are representations of the interaction between objects and classes given by a sequence of events along a time line which helps to display the functionality. By the other hand states machines displays a class object and the iteration of this object through all the system. The principal use of these diagrams is to display the transition events of the objects through the system in a manner that developers could understand easily how the interaction of the user using with the system takes place. These types of diagrams could be found in the Appendix D of this document.

# 5. Glossary

**Sencha Touch**: Sencha Touch, a high-performance HTML5 mobile application framework, is the cornerstone of the Sencha HTML5 platform. Built for enabling world-class user experiences, Sencha Touch is the only framework that enables developers to build powerful apps that work on iOS, Android, BlackBerry, Windows Phone, and more.

**Spring**: Spring helps development teams everywhere build simple, portable, fast and flexible JVM-based systems and applications

**Cordova**: free and open source framework that allows you to create mobile apps using standardized web APIs for the platforms you care about.

# 6. Appendix

## 6.1 Appendix A - Complete use cases

|  |  |
| --- | --- |
| Use Case ID | PPM-01 Display Hours of Operation |
| **Description** | Allow users to view the hours of operation specified by the client |
| **Actor** | User |
| **Pre-conditions** | 1. The hours of operations have been inputted into the database by the client 2. The user has downloaded and opened the app 3. The user has clicked the more button, and clicks on ‘Hours of Operations’ button |
| **Main success scenario** | 1. The system retrieves the information from the database 2. The system sends the information to the controller 3. The controller sends information to the view and displays information to user |
| **Post-conditions** | 1. The view now contains the hours of operations |
| **Exceptions** | If any error has occurred, the system will display a message indicating the error |

|  |  |
| --- | --- |
| Use Case ID | PPM-02 Trolley Alerts |
| **Description** | Allow user to receive alerts about any issues or problems in regards to the trolleys |
| **Actor** | User |
| **Pre-conditions** | 1. The user has opened the app |
| **Main success scenario** | 1. The system checks in database if any current issue is present 2. System shows alert box with alert message |
| **Post-conditions** | 1. The user can now make changes to plans |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-03 Show Estimated Time to Favorite Stop |
| **Description** | The system updates the user with the estimated time of the user’s favorite stop |
| **Actor** | User |
| **Pre-conditions** | 1. The user has set a favorite stop in the application |
| **Main success scenario** | 1. The user clicks on the favorites stop tab 2. The system retrieves the location of the trolley through the API. 3. The system retrieves the user’s favorite stops information. 4. The system sends the information to the view 5. The user can now see the estimated time of arrival for the stop(s) |
| **Post-conditions** | The user can now plan accordingly |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-04 Set Favorite Stop |
| **Description** | Allows user to define their favorite stop |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system |
| **Main success scenario** | 1. The user has chosen a route 2. The system displays all the stops for that given route 3. The user clicks the add to favorites button 4. The system asks for the timer notification for that stop 5. The user selects the timer option 6. System sends confirmation alert |
| **Post-conditions** | The user will now receive alerts for their selected stop(s) |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-05 Remove Favorite Stop |
| **Description** | Allows user to remove their favorite stop |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system |
| **Main success scenario** | 1. The user has selected the favorite stops tab 2. The system retrieves the user’s favorites information 3. The system displays the stop(s) to the user in the view 4. The user selects the ‘Remove Stop’ button 5. System sends a an alert 6. User acknowledges actions 7. System removes stop from the database |
| **Post-conditions** | The stop will longer exist for the user’s favorite stop(s) list |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-06 List Stops |
| **Description** | Allows user to remove their favorite stop |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system |
| **Main success scenario** | 1. The user has selected the route 2. The system retrieves the stops for that route in the database 3. The system displays all of the stops for that route |
| **Post-conditions** | 1. The user can now set stop(s) as favorites 2. The user can see the stops for the routes |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-07 Map Routes |
| **Description** | Allows the user to see all of the routes for the trolley |
| **Actor** | User |
| **Pre-conditions** | 1. User has selected the map button |
| **Main success scenario** | 1. The system retrieves the routes information 2. The map displays color coated routes and stops |
| **Post-conditions** | The user can now see all of the routes and stops on a map |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-08 Show Estimated Time to Stop |
| **Description** | The user can see the estimated time of arrival for each stop |
| **Actor** | User |
| **Pre-conditions** | 1. The user has selected a route |
| **Main success scenario** | 1. The system retrieves the stops for the given route 2. The user is looking at the list 3. The system displays the estimated time of arrival for each stop |
| **Post-conditions** | The user can now plan accordingly |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-09 Find Nearest Stop |
| **Description** | Allows user to find the closest stop to their location in the Village of Pinecrest |
| **Actor** | User |
| **Pre-conditions** | 1. The user has opened the app |
| **Main success scenario** | 1. The user clicks on the ‘Near Me’ button 2. The system retrieves the user’s location 3. The system find the closest location using the API 4. The application shows the user where the closest stop is |
| **Post-conditions** | The user can now go to that stop |
| **Exceptions** | The user is now in the Village of Pinecrest |

|  |  |
| --- | --- |
| Use Case ID | PPM-10 Arrived to Location Alert |
| **Description** | The system sends an alert when the trolley has arrived to users favorite stop |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system 2. User has set the favorite stops with the alert timer |
| **Main success scenario** | 1. The system checks the database to see if the user has any stops chosen 2. The system retrieves the location of the trolley through the API 3. The system sends the user a notification that the trolley has arrived to their selected stop |
| **Post-conditions** | 1. The user is now notified of the arrival of the trolley to the favorite stop |
| **Exceptions** | The user is not logged on the system. |

|  |  |
| --- | --- |
| Use Case ID | PPM-11 Login |
| **Description** | The system allows the user to login to retrieve information |
| **Actor** | User |
| **Pre-conditions** | 1. User has opened the app 2. User selects the login option |
| **Main success scenario** | 1. The system displays the login panel 2. The user enters the username and password 3. The system retrieves all of the favorite stop information 4. The system retrieves all of the alerts information |
| **Post-conditions** | 1. The user can now receive alerts |
| **Exceptions** | The user types in wrong credentials |

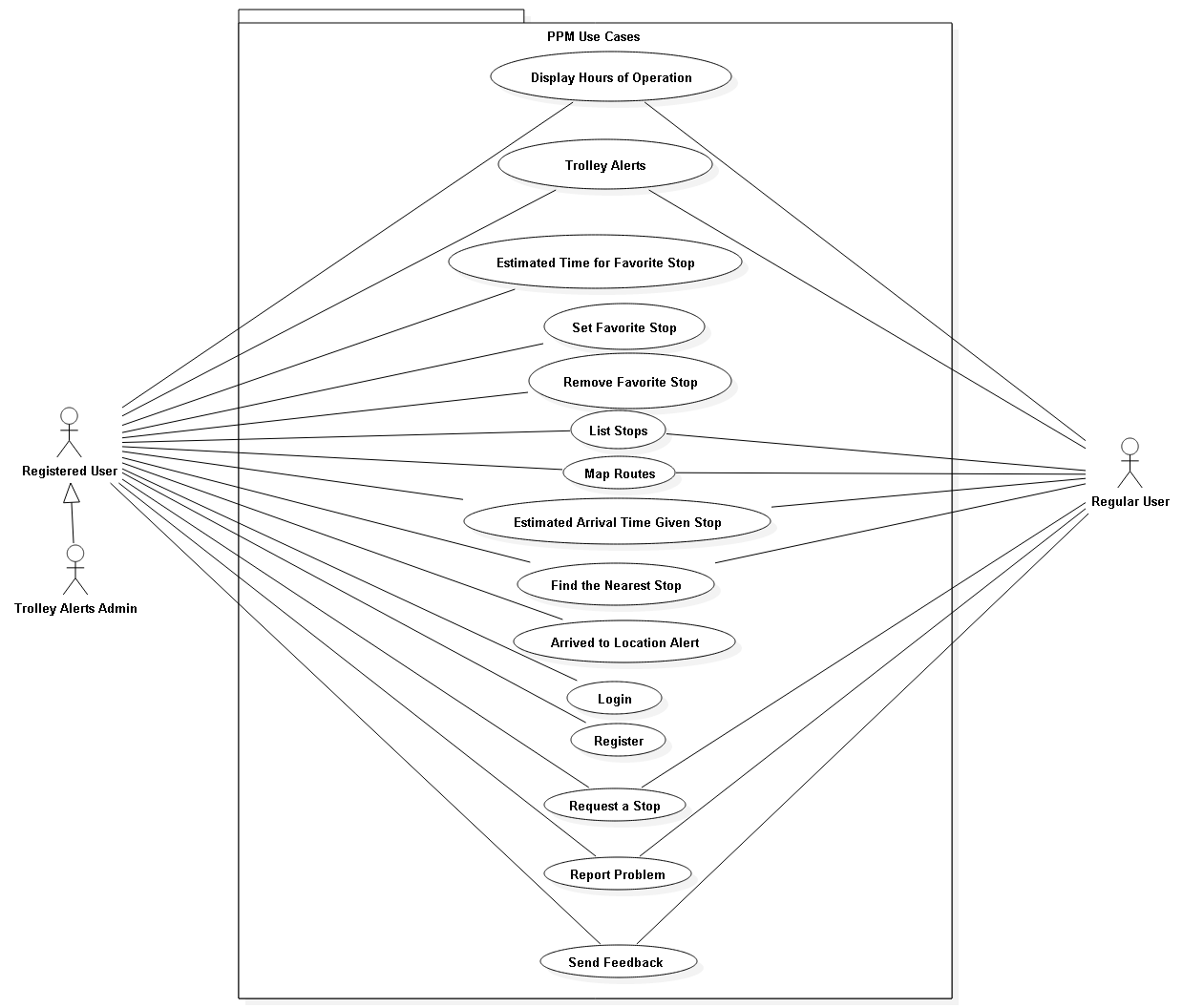
|  |  |
| --- | --- |
| Use Case ID | PPM-12 Register User |
| **Description** | Allows user to Register on the system |
| **Actor** | User |
| **Pre-conditions** | 1. The user has opened the app 2. The user selects the register button |
| **Main success scenario** | 1. The system validates that the filed username is populated with an email. 2. The system validates that the password and confirm password entered are equal. 3. If validations are correct the system save the email address and password introduced. |
| **Post-conditions** | The system enabled options related to the management use cases |
| **Exceptions** |  |

|  |  |
| --- | --- |
| Use Case ID | PPM-13 Request a stop |
| **Description** | Allow user to send messages to the system. In this case to request a stop |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system to automatically capture sender email 2. User enters email to the system. |
| **Main success scenario** | 1. The system creates a message with a formatted subject with one of the stated topics 2. Send an email to the corresponding authority. |
| **Post-conditions** | 1. The email is sent to the corresponding authority |
| **Exceptions** | The user has typed in wrong email or incorrect email format |

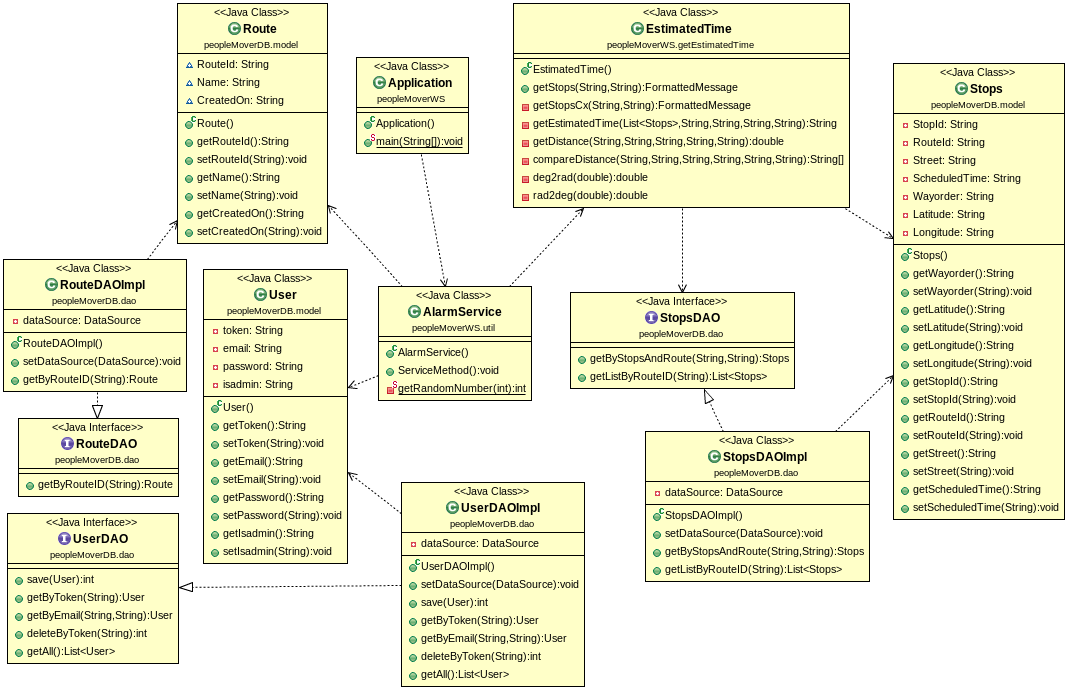
|  |  |
| --- | --- |
| Use Case ID | PPM-14 Report a Problem |
| **Description** | Allow user to send messages to the system. In this case to report a problem |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system to automatically capture sender email 2. User enters email to the system. |
| **Main success scenario** | 1. The system creates a message with a formatted subject with one of the stated topics 2. Send an email to the corresponding authority. |
| **Post-conditions** | 1. The email is sent to the corresponding authority |
| **Exceptions** | The user has typed in wrong email or incorrect email format |

|  |  |
| --- | --- |
| Use Case ID | PPM-15 Send Feedback |
| **Description** | Allow user to send messages to the system. In this case to send feedback |
| **Actor** | User |
| **Pre-conditions** | 1. User is logged on the system to automatically capture sender email 2. User enters email to the system. |
| **Main success scenario** | 1. The system creates a message with a formatted subject with one of the stated topics 2. Send an email to the corresponding authority. |
| **Post-conditions** | 1. The email is sent to the corresponding authority |
| **Exceptions** | The user has typed in wrong email or incorrect email format |

## 6.2 Appendix B - Use case diagram using UML

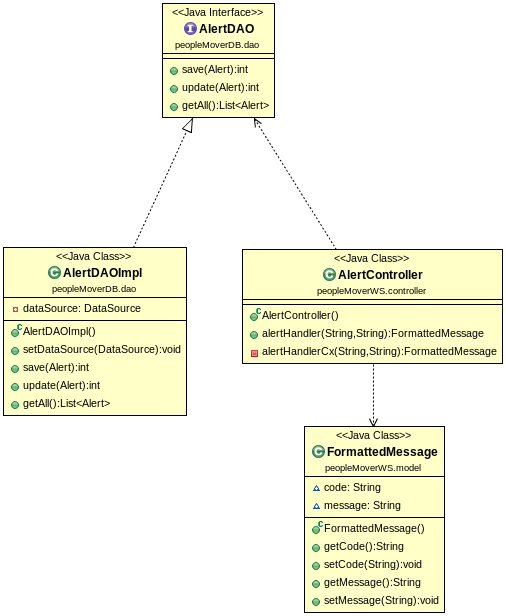


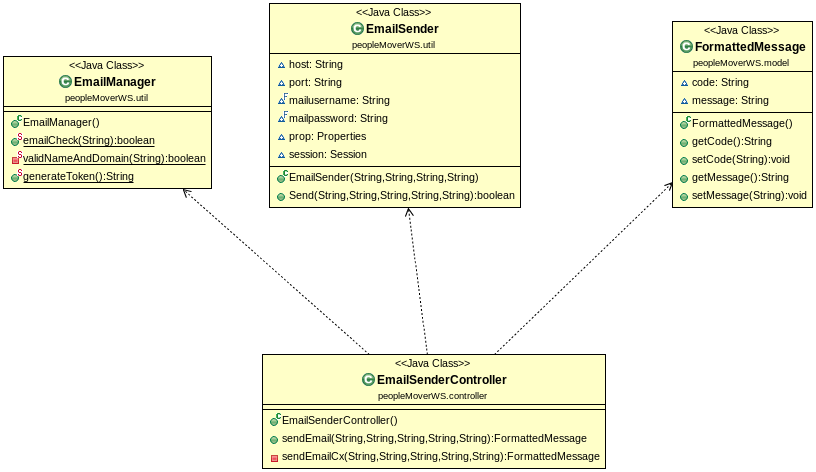
## 6.3 Appendix C - Static UML diagram



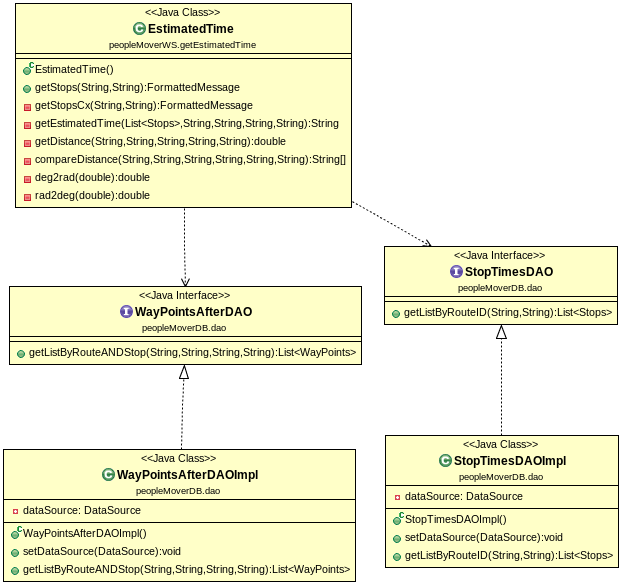
Alarm Service

Alert Subsytem

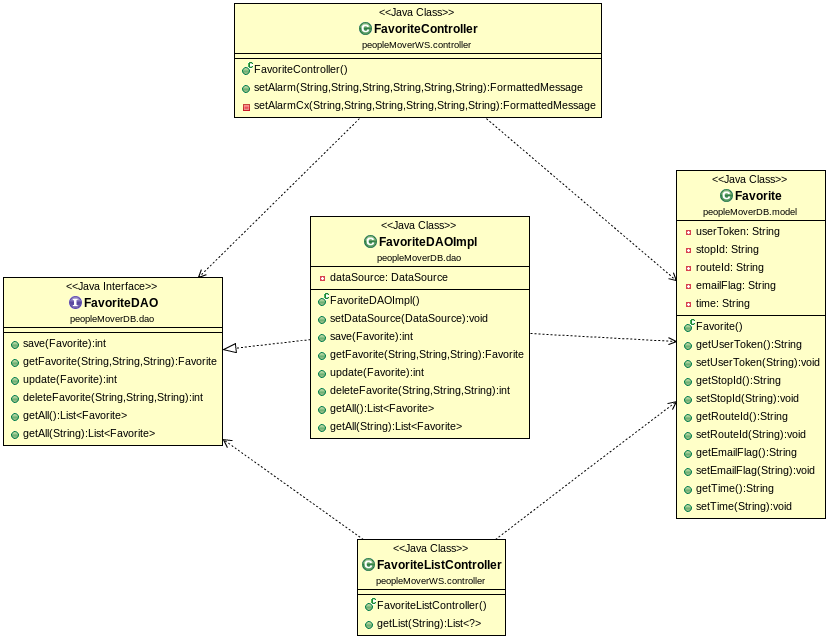


Email Subsystem

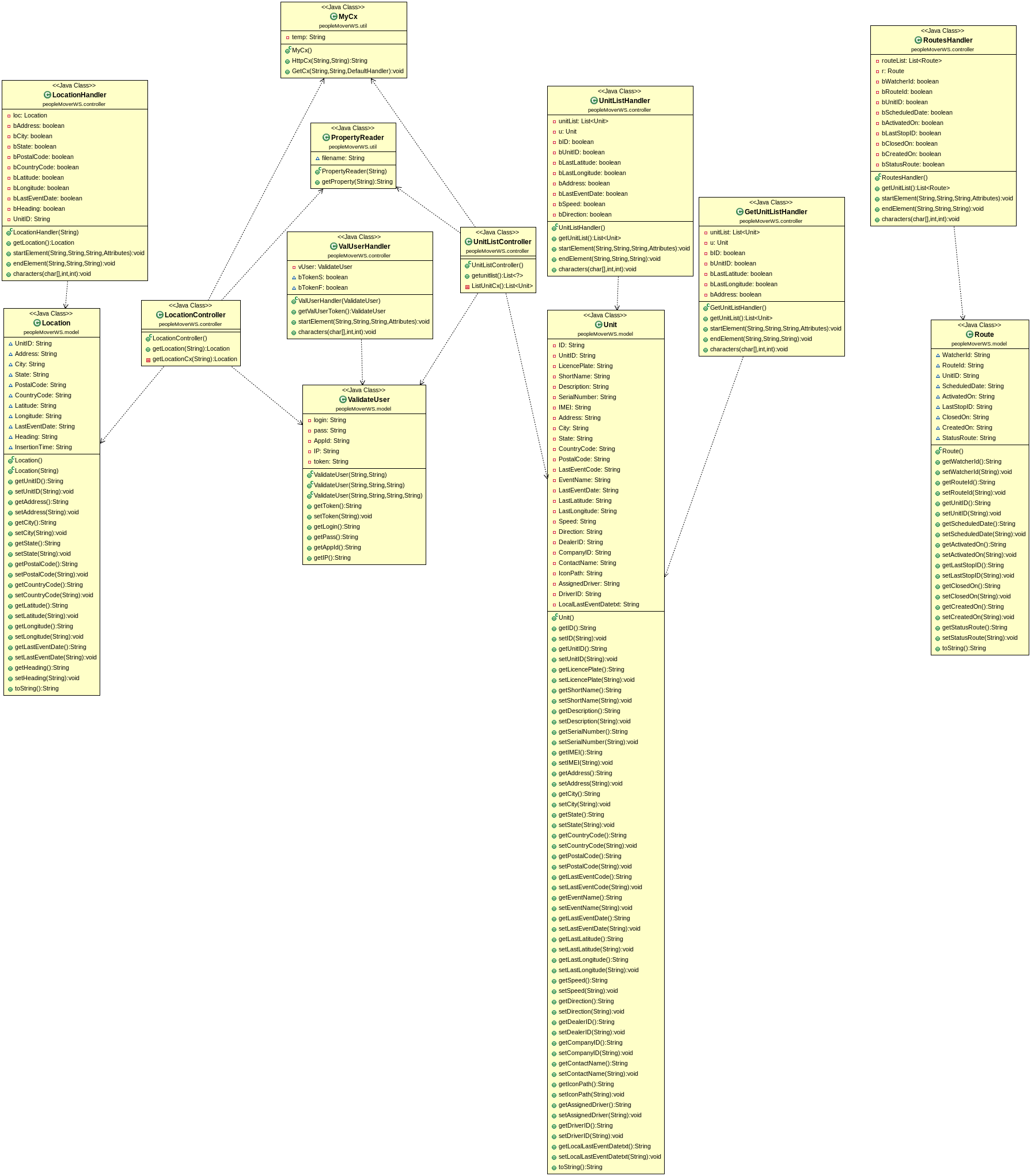
Estimated Time



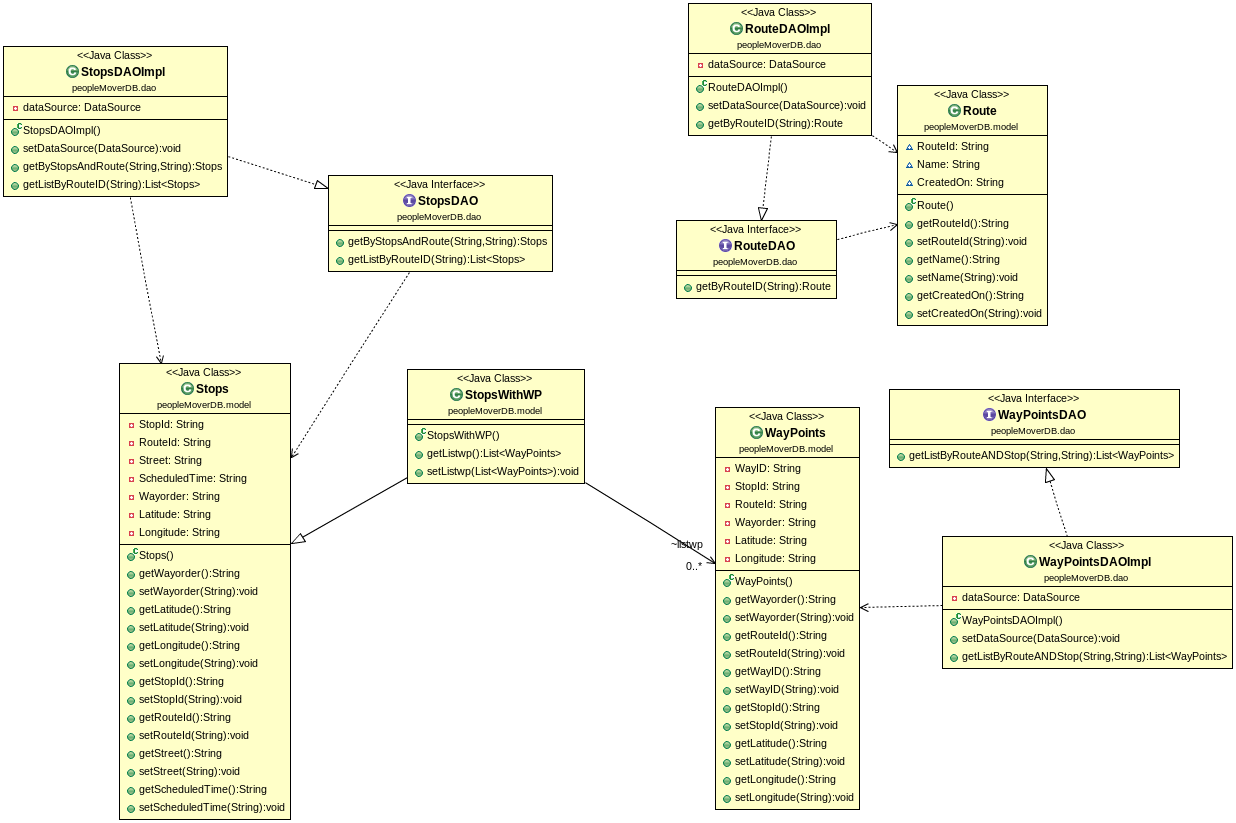
Favorite Definition

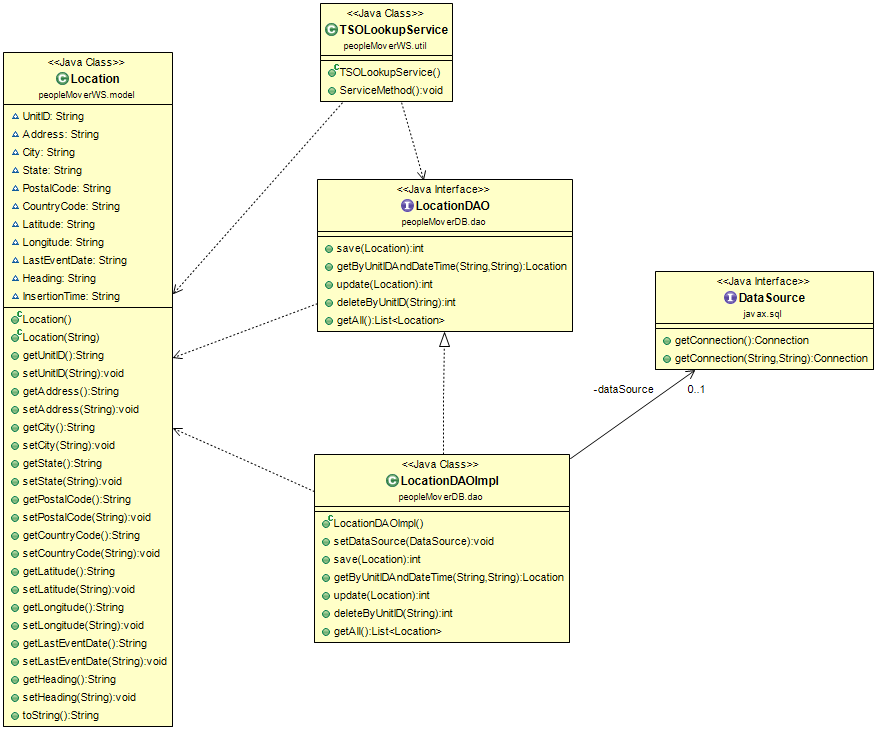


Location, Unit and Route

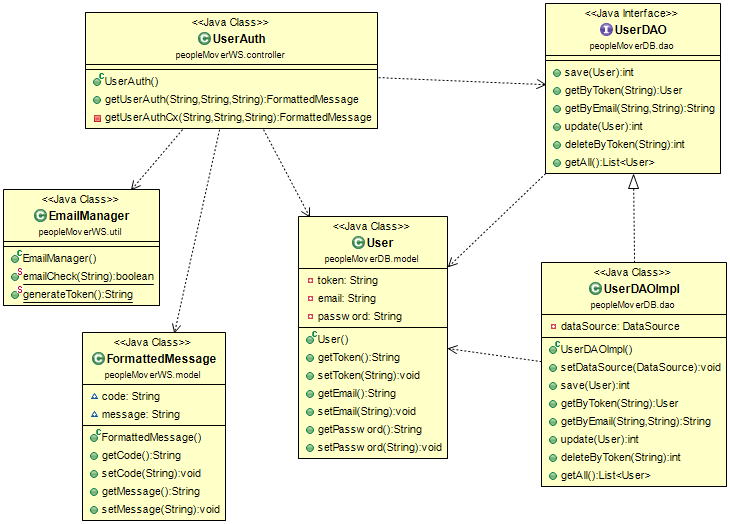


Stops, Route, and Waypoints

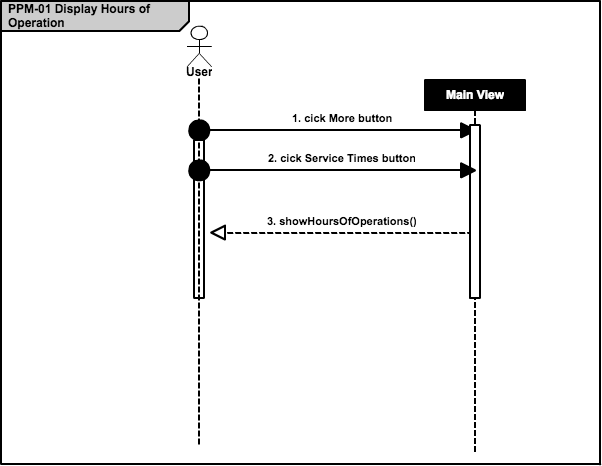


TSOLookUpService (Historical Location)

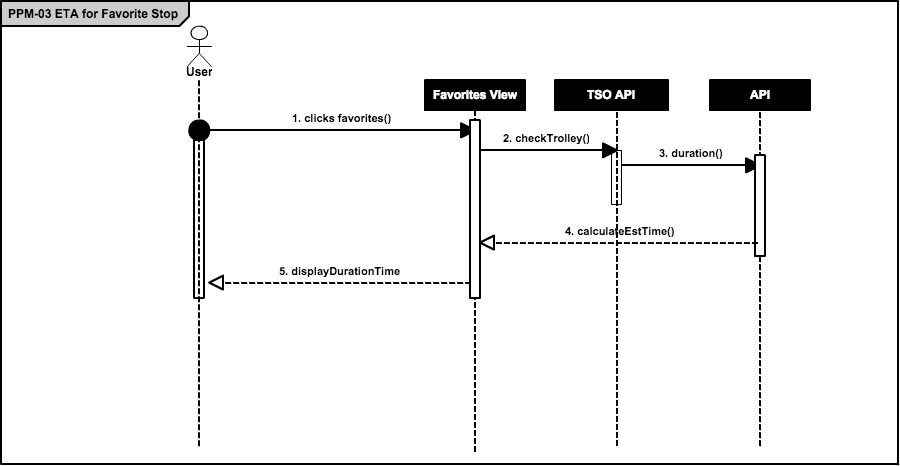
Authentication Service

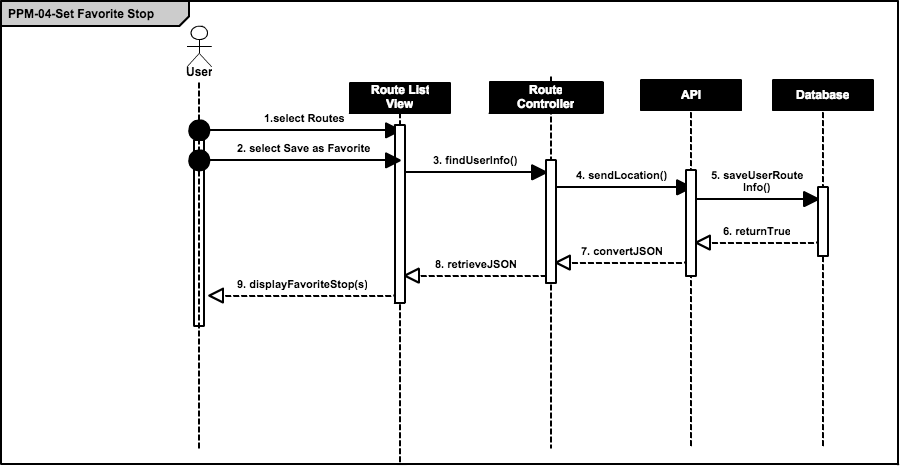


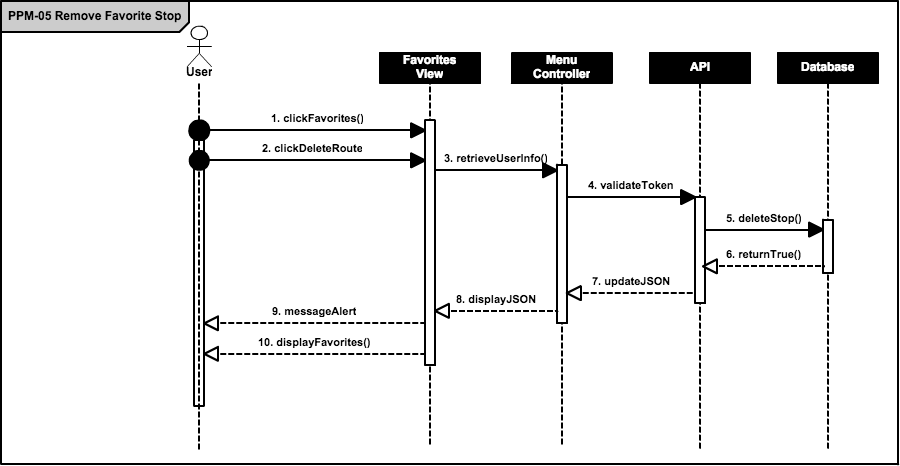
## 6.4 Appendix D - Dynamic UML diagrams

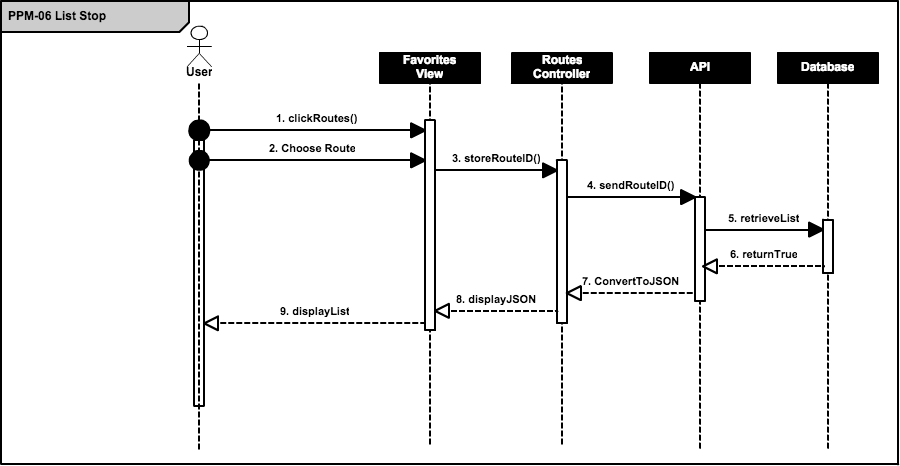


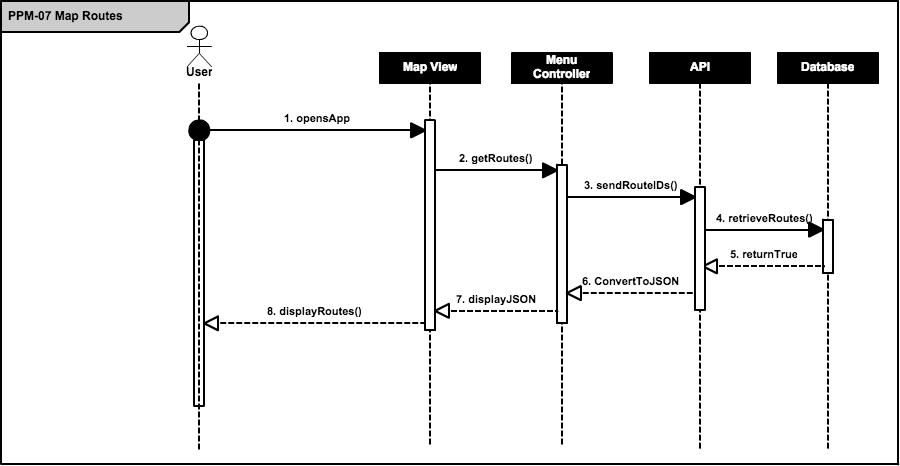


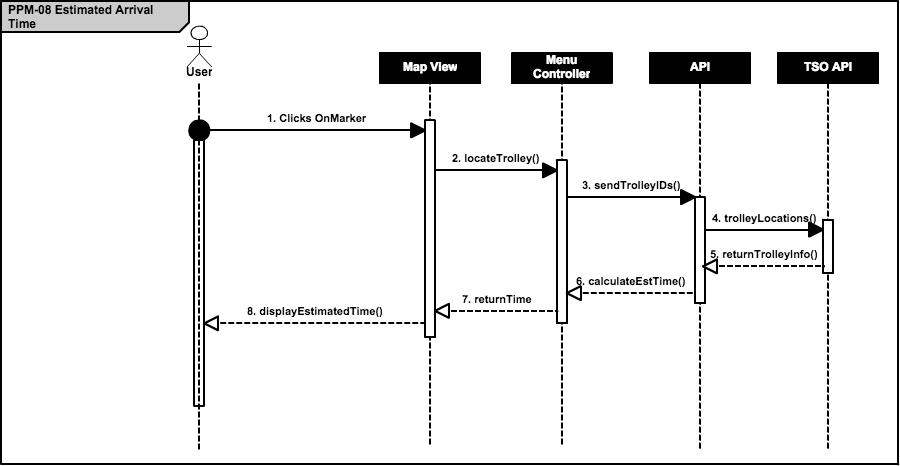


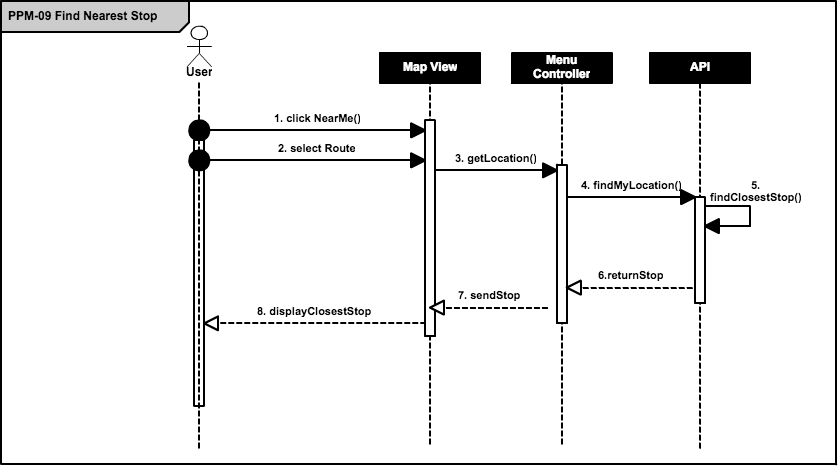


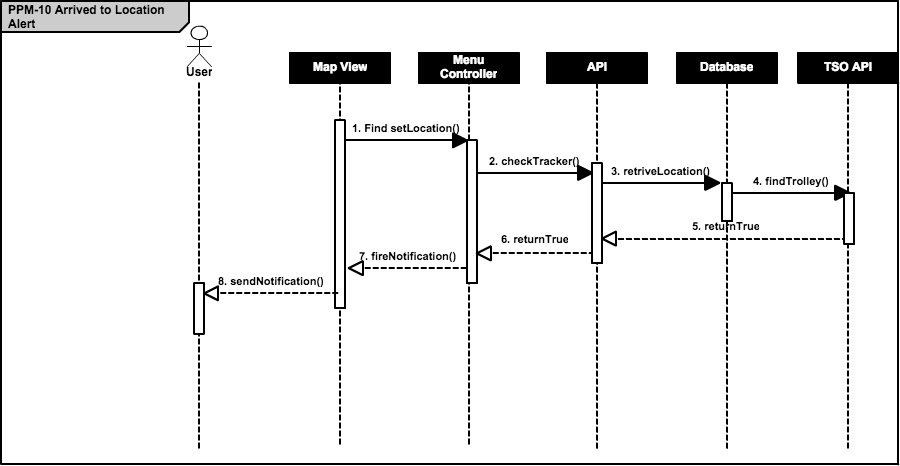


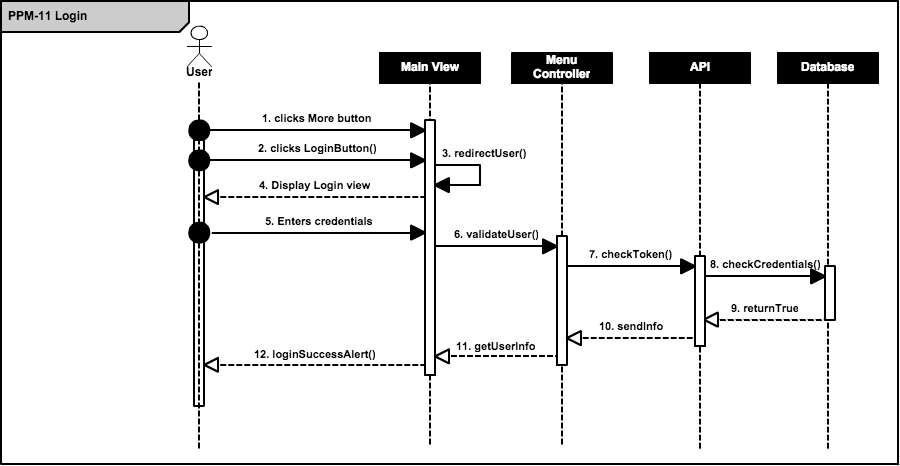


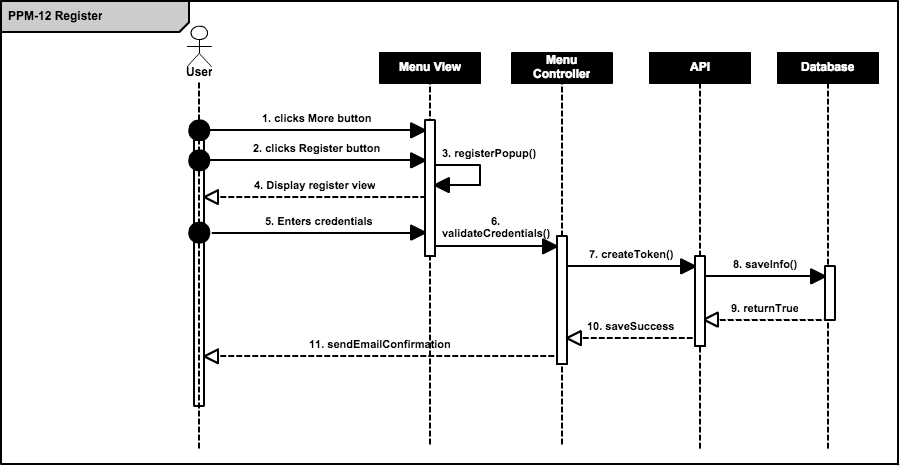


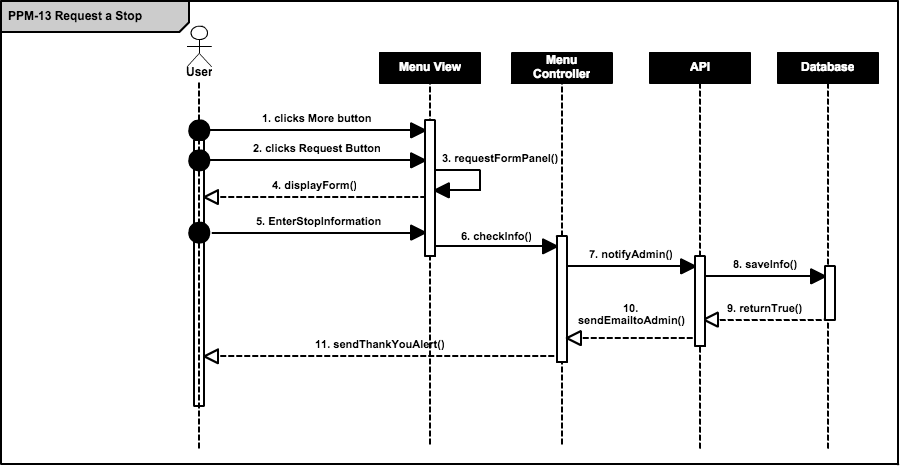


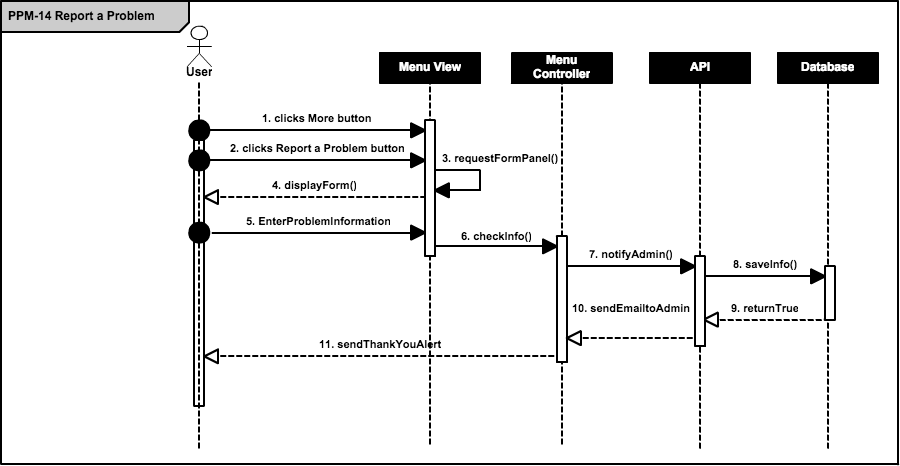


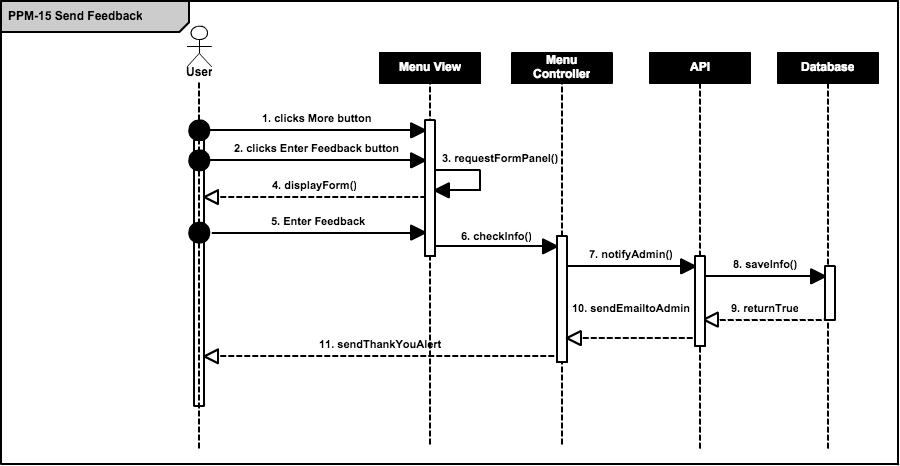












## 6.5 Appendix E - User Interface designs.

Figure E-1 : Loading Image



Figure E-2: Main Menu and Insert Favorite

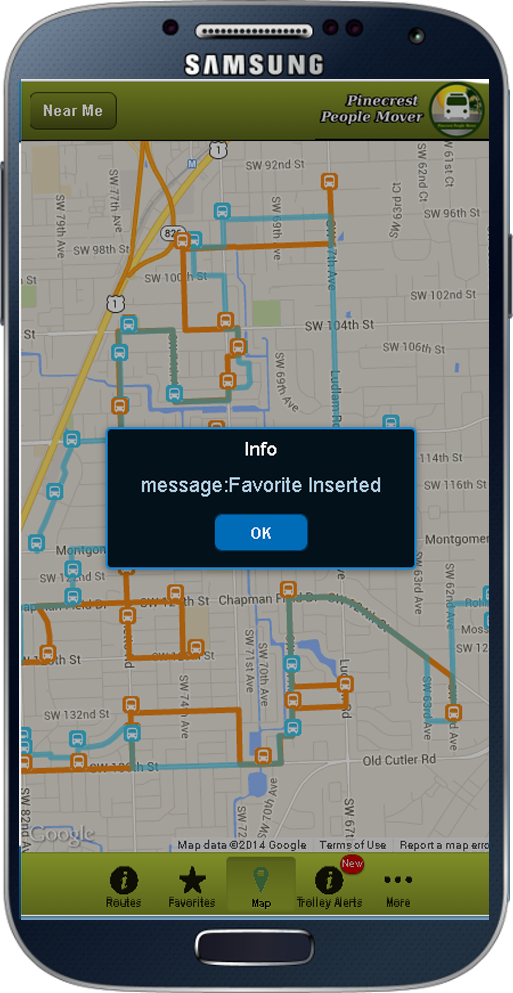


Figure E-3: Stop List

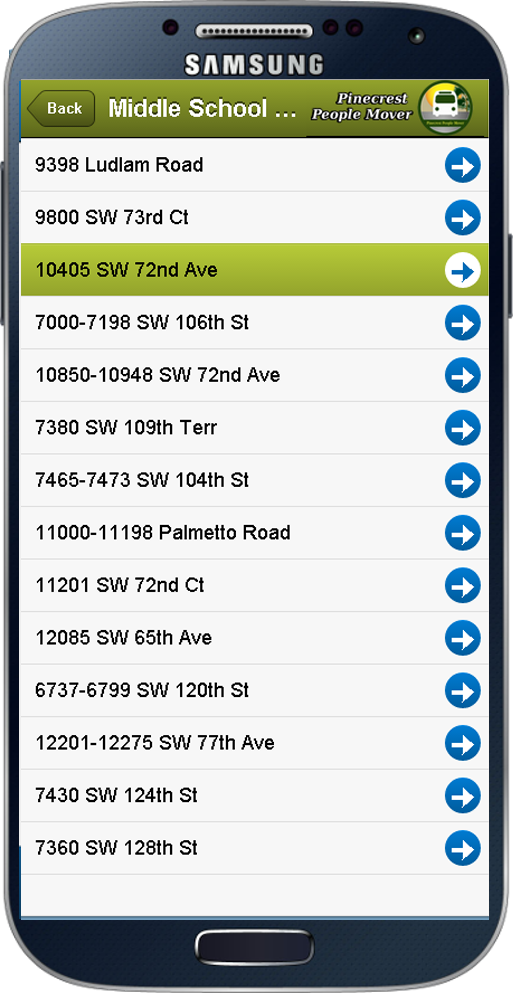


Figure E-4: Trolley Alerts

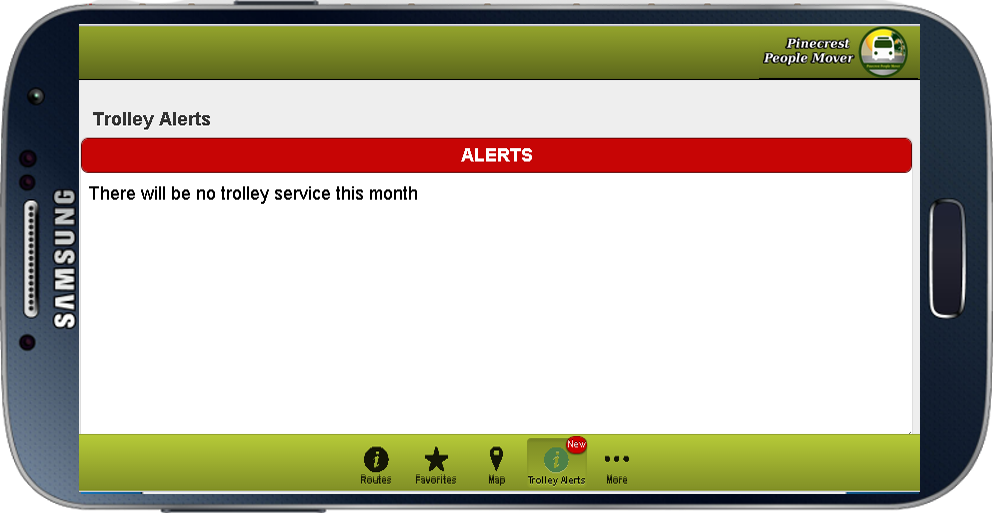
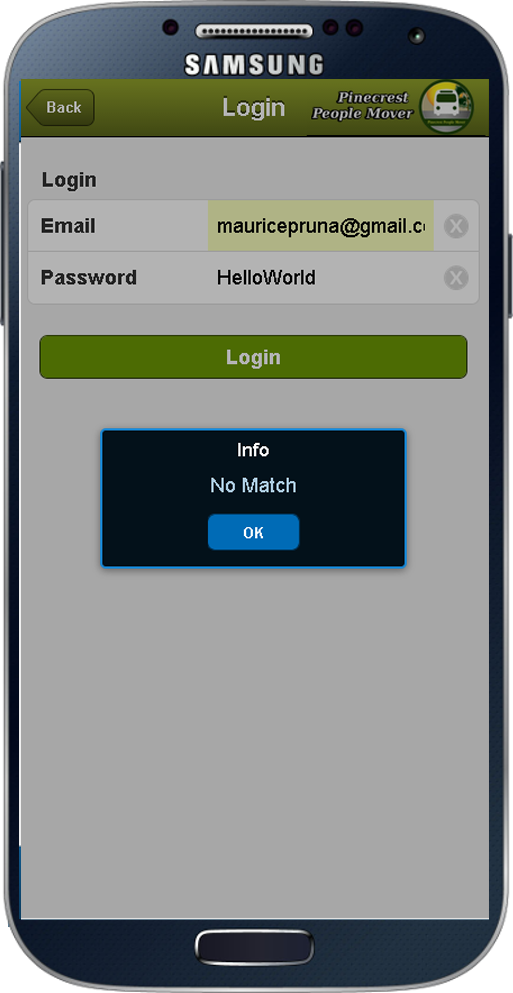


Figure E-5: Login View



## 6.6 Appendix F - Diary of meeting and tasks.

Meeting 1

|  |  |
| --- | --- |
| Date: | Wednesday, September 3, 2014 |
| Location: | Google Hangouts |
| Start time: | 8:00 pm |
| End time: | 8:30 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Introduction to Teammates |
|  | 2. Discuss ideas and gather information of what may be needed for the project. (use Miami-dade transit website as reference) |
|  | 3. Go over information in regards to PineCrest Mover |
| Assigned Tasks: |  |
|  | 1. Meet September 4, 2014 at FIU |
|  |  |

Meeting 2

|  |  |
| --- | --- |
| Date: | Wednesday, September 4, 2014 |
| Location: | FIU, JCCL Lab |
| Start time: | 8:15 pm |
| End time: | 9:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Gather questions for tomorrow’s conference call with mentor |
|  | 2.Share bootstrap and Google maps API resources |
|  | 3. Go over what is needed in document and feasibility matrix |
| Assigned Tasks: |  |
|  | 1. Meet September 5, 2014 for conference call 12:30pm – 1:30 pm |
| Duration: | 45 minutes |

Meeting 3

|  |  |
| --- | --- |
| Date: | Friday, September 5, 2014 |
| Location: | Conference Call |
| Start time: | 12:30 am |
| End time: | 1:30 pm |
| In Attendance: | Gabriela Wilson, Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Problem discussion and identification of possible users   * Agreed to start a Web/Mobil app * Defined guest user for general interaction * Defined registered user for “favorite bus stop” * Extra tool requested to manage routes and buses. |
|  | 2. Defined status of the GPS devices on the movers.   * Identified Trolley Company as the source of GPS data. |
|  | 3. Identified similar applications from other local governments. |
|  |  |
| Assigned Tasks: | 1. Create mock ups design 2. Define use cases 3. Contact the trolley company to define the data retrieval |
| Duration: | 60 minutes |

Meeting 4

|  |  |
| --- | --- |
| Date: | Saturday, September 6, 2014 |
| Location: | FIU, JCCL Lab |
| Start time: | 10:30 pm |
| End time: | 4:30 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Begin Feasibility Study document |
|  | 2.Begin Project Plant document |
|  | 3.Discuss best approach for implementation of project |
| Assigned Tasks: |  |
|  | 1. Meet September 7, 2014 10:30 AM at FIU |
| Duration: | 6 hours |

Meeting 5

|  |  |
| --- | --- |
| Date: | Sunday, September 7, 2014 |
| Location: | FIU, JCCL Lab |
| Start time: | 11:00 pm |
| End time: | 4:45 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Think of specifications |
|  | 2.Begin Specifications doc |
|  | 3.Discuss best approach for implementation of project |
| Assigned Tasks: |  |
|  | Submit first revision documents |
| Duration: | 5 hours 45 minutes |

Meeting 6

|  |  |
| --- | --- |
| Date: | Tuesday, September 9, 2014 |
| Location: | FIU, JCCL Lab |
| Start time: | 6:30 pm |
| End time: | 8:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Check Requirements Feedback |
|  | 2. Work on the mockups design |
|  |  |
| Assigned Tasks: | Continue working on the mockups |
|  | Improve use cases model |
| Duration: | 90 min |

Meeting 7

|  |  |
| --- | --- |
| Date: | Wednesday, September 10, 2014 |
| Location: | Conference Call |
| Start time: | 12:00 am |
| End time: | 12:30 am |
| In Attendance: | Ricardo Martinez, Maurice Pruna, Gabriela Wilson |
| Agenda: | 1. Check Requirements |
|  | 2. Check mockups design updates |
|  | 3. Check business workflow |
|  | 4. Status of the TSO api and GPS devices |
| Assigned Tasks: | Continue working on the mockups |
|  | Improve use cases model |
| Duration: | 30 min |

Meeting 8

|  |  |
| --- | --- |
| Date: | Thursday, September 11, 2014 |
| Location: | Starbucks |
| Start time: | 7:30 pm |
| End time: | 9:30 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Read documentation for Sencha |
|  | 2. Did some research for MVC architecture |
|  |  |
| Assigned Tasks: | Continue working on the mockups |
|  | Begin with interfaces |
| Duration: | 2 hours (120 minu) |
|  |  |

Meeting 9

|  |  |
| --- | --- |
| Date: | Sunday, September 14, 2014 |
| Location: | FIU JCCL Labs |
| Start time: | 3:30 pm |
| End time: | 7:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Continue with Interface implementation |
|  | 2. Provide feedback and assign roles |
| Assigned Tasks: | Continue with Interface Implementation |
|  | Improve use cases model |
| Duration: | 3hours 30 min |
|  |  |
|  |  |

Meeting 10

|  |  |
| --- | --- |
| Date: | Wednesday, September 17, 2014 |
| Location: | Conference Call |
| Start time: | 12:00 pm |
| End time: | 1:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna, Gabriela Wilson, David Gonzalez (TSO) |
| Agenda: | 1. Ask questions in regards to usage of API |
|  | 2. Database integration |
| Assigned Tasks: | Look over API Documenation |
|  |  |
| Duration: | 60 min |
|  |  |
|  |  |

Meeting 11

|  |  |
| --- | --- |
| Date: | Sunday, September 21, 2014 |
| Location: | FIU JCCL Lab |
| Start time: | 3:30 pm |
| End time: | 7:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Go over API Documentation |
|  | 2. Continue with interface implementation |
| Assigned Tasks: |  |
|  |  |
| Duration: | 3hour 30 min |
|  |  |
|  |  |

Meeting 13

|  |  |
| --- | --- |
| Date: | Thursday, September 25, 2014 |
| Location: | FIU JCCL Lab |
| Start time: | 7:30 pm |
| End time: | 9:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. API Connection |
|  | 2. Continue with interface implementation |
| Assigned Tasks: |  |
|  |  |
| Duration: | 1hour 30 min |
|  |  |
|  |  |

Meeting 14

|  |  |
| --- | --- |
| Date: | Saturday, September 27, 2014 |
| Location: | Starbucks |
| Start time: | 3:00 pm |
| End time: | 6:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. API Connection |
|  | 2. Continue with interface implementation |
| Assigned Tasks: |  |
|  |  |
| Duration: | 3 hours |
|  |  |
|  |  |

Meeting 15

|  |  |
| --- | --- |
| Date: | Sunday, September 28, 2014 |
| Location: | Starbucks |
| Start time: | 3:00 pm |
| End time: | 7:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Prepare Documentations |
|  |  |
| Assigned Tasks: | Ready to present for tomorrow’s class presentation |
|  |  |
| Duration: | 4 hours |
|  |  |
|  |  |

Meeting 16

|  |  |
| --- | --- |
| Date: | Thursday, October 2, 2014 |
| Location: | Conference Call |
| Start time: | 2:00 pm |
| End time: | 3:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. API Connection lag time |
|  | 2. Presentation of interface implementation |
| Assigned Tasks: |  |
|  |  |
| Duration: | 1 hour |
|  |  |
|  |  |

Meeting 17

|  |  |
| --- | --- |
| Date: | Thursday, October 8, 2014 |
| Location: | Starbucks |
| Start time: | 7:30 pm |
| End time: | 10:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | 1. Show Markers on Map |
|  | 2. Show routes on Map |
| Assigned Tasks: |  |
|  |  |
| Duration: | 2 hours 30 minutes |
|  |  |
|  |  |

Meeting 18

|  |  |
| --- | --- |
| Date: | Thursday, October 8, 2014 |
| Location: | Starbucks |
| Start time: | 12:00 pm |
| End time: | 1:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna, Gabriela Wilson, David (TSO developer) |
| Agenda: | 1. Figure out lag time for trolley |
|  | 2. Update Gabriela with progress |
| Assigned Tasks: |  |
|  |  |
| Duration: | 1 hour |
|  |  |
|  |  |

Meeting 19

|  |  |
| --- | --- |
| Date: | Monday, October 13 2014 |
| Location: | Starbucks |
| Start time: | 7:00 pm |
| End time: | 10:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | Analysis of the DB design |
|  | Work on the automatic insertion on the historical table |
| Assigned Tasks: | Look information to create the native build  Interact with Google Maps Api to draw an asynchrony marker |
|  |  |
| Duration: | 3 hours |
|  |  |
|  |  |

Meeting 20

|  |  |
| --- | --- |
| Date: | Saturday, October 18 2014 |
| Location: | Starbucks |
| Start time: | 5:00 pm |
| End time: | 10:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | DB adjustments  Work on the connection between Sencha and the PPM API |
|  | Redesign of the States diagrams |
| Assigned Tasks: | Create the Model and Stores needed to retrieve information from the PPM API |
|  |  |
| Duration: | 5 hours |
|  |  |
|  |  |

Meeting 21

|  |  |
| --- | --- |
| Date: | Tuesday, October 21 2014 |
| Location: | Starbucks |
| Start time: | 7:00 pm |
| End time: | 10:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | Analysis of the DB design changes requested by Gabriela |
|  | Work on the synchronic calls to for the Trolley position within Sencha |
| Assigned Tasks: | using Google Maps Api to draw an asynchrony marker |
|  | Start working on the Login and Registration modules |
| Duration: | 3 hours |
|  |  |
|  |  |

Meeting 22

|  |  |
| --- | --- |
| Date: | Tuesday, October 23 2014 |
| Location: | FIU CS Lab |
| Start time: | 7:00 pm |
| End time: | 10:00 pm |
| In Attendance: | Ricardo Martinez, Maurice Pruna |
| Agenda: | Adjust classes on a MVC fashion on Sencha |
|  | Adjust the synchronic calls to for the Trolley position within Sencha |
| Assigned Tasks: | Testing of the drawing of asynchrony marker |
|  | Testing on the Login and Registration modules |
| Duration: | 3 hours |
|  |  |
|  |  |

# 7. References

<http://www.sencha.com/>

<http://www.sencha.com/forum/>

<http://spring.io/>

<http://www.sublimetext.com/>

<https://cacoo.com/diagrams/>

<https://developers.google.com/maps/documentation/javascript/tutorial>

<http://phonegap.com/blog/2013/11/20/SenchaPhoneGap/>