Design Document

Pinecrest People Mover

Members:

Ricardo Martinez

Maurice Pruna

Mentor:

Gabriela Wilson

Instructor:

Masoud Sadjadi

October 10, 2014

Florida International University School of Computing and Information Sciences

CIS 4911 U01

© 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 six sections: Introduction, System Design, Glossary, the 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 design methodology used, and provides a list of definitions and acronyms used in the document.

The System Design section provides a high level explanation of the system with the use of diagrams, descriptions of the hardware, software, data storage and security as well. The Detailed Design gives the description of the structure for each subsystem using static models and dynamic models. Also there is reference to the Code Specification on the Appendix C.

Table of Contents

1. [Introduction 4](#_Toc401928471)

[1.1 Problem definition. 4](#_Toc401928472)

[1.2 Design methodology used 4](#_Toc401928473)

[1.3 Definitions, acronyms, and abbreviations. 4](#_Toc401928474)

[1.4 Overview of document 4](#_Toc401928475)

1. [System Design 6](#_Toc401928476)

[2.1 Overview 6](#_Toc401928477)

[2.2 Subsystem Decomposition 6](#_Toc401928478)

[2.3 Hardware and Software Mapping 7](#_Toc401928481)

[2.4 Persistent Data Management 7](#_Toc401928482)

[2.5 Security/Privacy 8](#_Toc401928483)

1. [Detailed Design 8](#_Toc401928484)

[3.1 Overview 8](#_Toc401928485)

[3.2 Static model 8](#_Toc401928486)

[3.3 Dynamic model 8](#_Toc401928487)

[3.4 Code Specification 8](#_Toc401928488)

1. [Glossary 9](#_Toc401928489)
2. [Appendix 10](#_Toc401928490)

[5.1 Appendix A 10](#_Toc401928491)

[5.2 Appendix B 11](#_Toc401928492)

[5.3 Appendix C 12](#_Toc401928493)

[5.4 Appendix D 13](#_Toc401928494)

1. [References 14](#_Toc401928495)

# 1. Introduction

This section gives a brief indication about what is the problem presented by the community of Pinecrest in the person of our mentor and client. The have been using their trolley system called People Mover (PPM)

## 1.1 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.

## 1.2 Design methodology used

The agile software development model is what is being used to develop the system. We specified the scope of the system, analyzed the use cases needed, and defined the functional and non-functional requirements in the Requirement Document (RD). In this Design Document, we will use the same analysis model from the Requirement Document to show the design objects, system architecture, and subsystem decomposition that will be part of the process. The agile system will guarantee that the documentation process is well designed but at the same time we are going to be involved in the developing process to get a fast delivery of the system to the client. The UML diagrams as part of the analysis model are required to aid in designing the system. The use case diagrams from the RD makes the creation of the class diagrams and sequence diagrams easier to define.

## 1.3 Definitions, acronyms, and abbreviations.

PPM: Pinecrest People Mover

SRS: Software Requirements Specification

UML: Unified Modeling Language

## 1.4 Overview of document

* Chapter 2 describes the System Design more focused in:
  + Overview
  + System Decomposition
  + Hardware and Software Mapping
  + Persistent Data Management
  + Security/Privacy
* Chapter 3 describes the Detailed Design of the system
  + Overview
  + Static model
  + Dynamic model
  + Code Specification
* Chapter 4 Glossary
* Chapter 5 is Appendix
* Chapter 6 is References

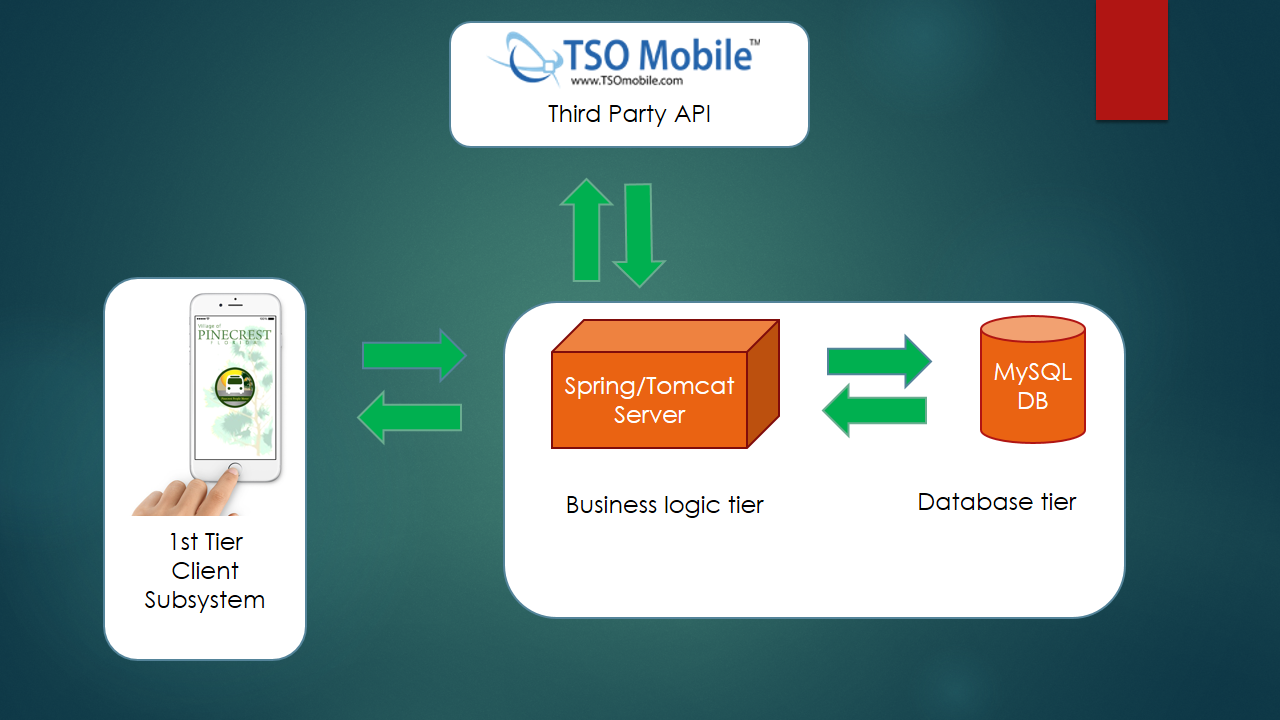
# 2. System Design

Along this chapter we will cover our system and subsystem design. We will present:

* High-level description of the system design.
* Detail description of each of the major subsystems.
* Maps of the interaction between the hardware and the subsystems.
* Structure of the data to be stored.
* Authentication processes and security in general.

## 2.1 Overview

The design chosen to be used in our system is based in a three tier architecture. Given that our system has an important part mobile oriented, we thought that a light weight data processing would be beneficial, also this represents easier scalability and better handling of the security. Our system also implements a client server architecture given that our business logic and database reside in the same server and they exchange resources and services. The subsystem defined as client side is an application developed using Sencha Touch framework. The server subsystem is composed by a tomcat server with a resident web app using Spring Framework over Java and MySQL as database server The server subsystem takes care of the resources requested from the client subsystem, database processing and requests to a third party API that provide data of the trolleys as well.

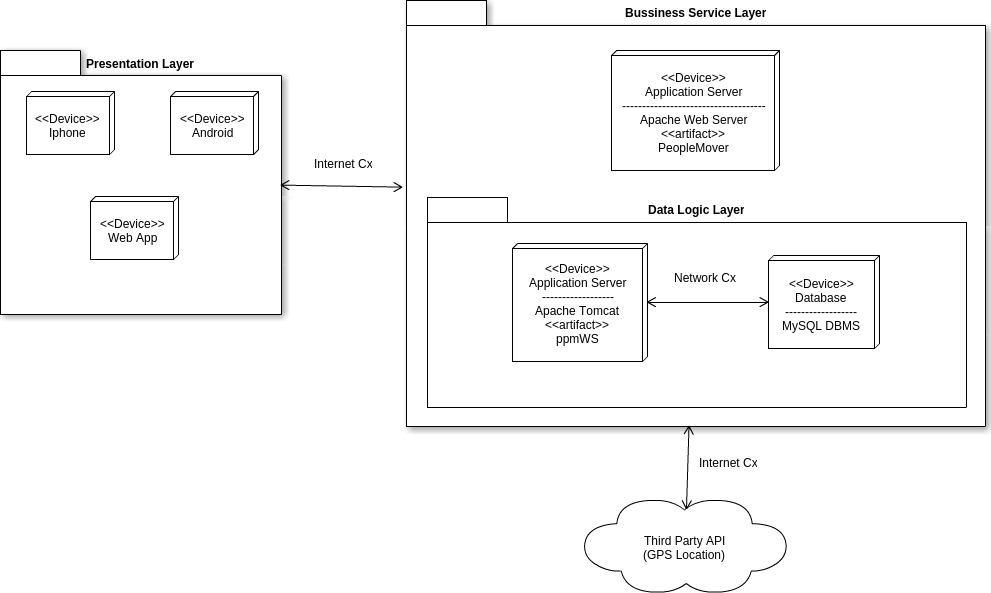


## 2.2 Subsystem Decomposition

This system is composed by 4 subsystems. The first one is the Client subsystem which will manage how the interface is delivered to the user. This subsystem will provide to regular and registered users with features to allow tracking of trolleys, find near stops and send message to the system. The Authentication subsystem will provide with the functionality to perform specific task to registered users (like define favorite stops, define alerts). This subsystem resides part in the client side and part in the business logic tier. The part in the business logic tier will perform most of the validation process and will manage the information request corresponding to registered users. The Web Service subsystem will manage the interaction with the third party API that will provide the system with basic trolley information, will store and retrieve information from the database, and will deliver information to the Client Subsystem in the form of web service. The Email subsystem will take care of process all the email transmission requested from the Client subsystem and will be used as method to manage the stops alerts.



## Hardware and Software Mapping

The hardware and software mapping is shown below:

## \\buffalo.cs.fiu.edu\homes\Downloads\ppm.png2.4 Persistent Data Management

## 2.5 Security/Privacy

The user’s passwords will be encrypted using a hashing function. This password will be encrypted before being sent to the Database. In case of an intruder get to the database it will not be able to retrieve the original password. The regular users will not be able to save any favorite stop and/or define an alert for a favorite stop. All the authentication process from the client side is validated through the People Mover API. This API is the one that connects to the database retrieves the necessary information and send it back to the client side.

# 3. Detailed Design

The detail design of the system is the step prior to the implementation process. It will a more detailed specification in a level that is easier enough to get into the implementation process in a straight way. All the modules present on the system are specified pursuing as much detail as possible. It will be used different modeling procedures that are part of the detail design. The sections are divided as follow:

Overview – This section briefly describe the behavior and structure of each subsystem.

Static model – This part provide detailed description of the structure for each subsystem.

Dynamic model – This section provides different diagrams for the main control in each subsystem.

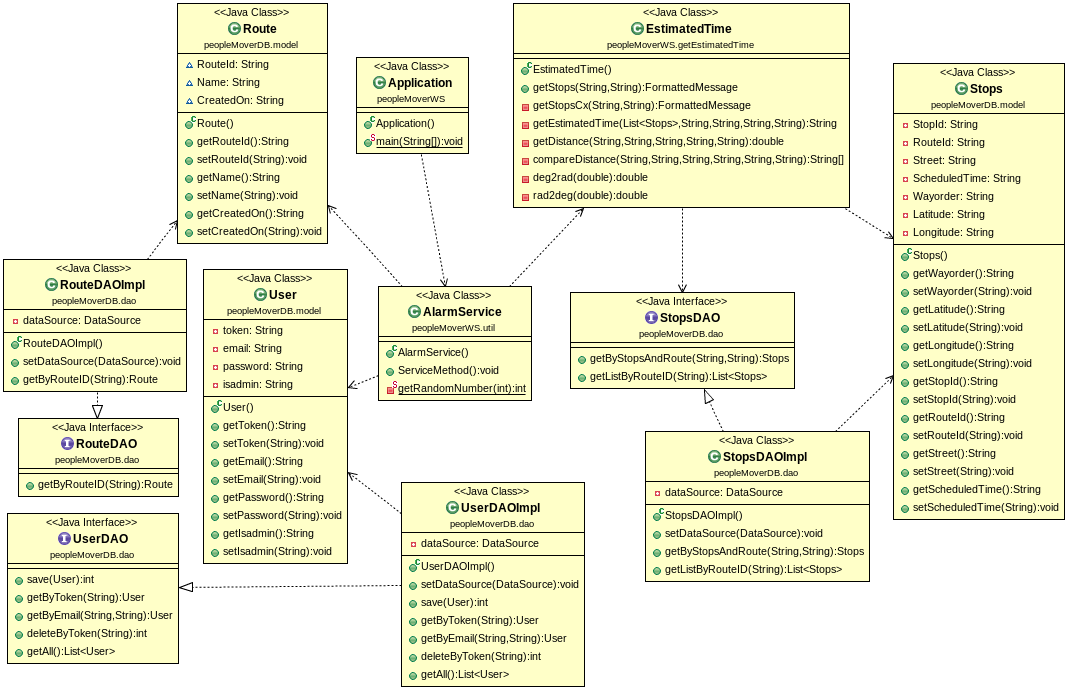
Code Specification –This section describes the class interfaces and constraints for the main control object in each system

## 3.1 Overview

The application we wish to create is composed of four subsystems. The first subsystem is the client subsystem. It has the overall forms and visual information needed on the client side. Users can check the trolley location on the map, also list stops and see the estimated time to arrival, check trolley alerts and send feedback, problems and requests to the system. The business layer contains the other three subsystems. The Web service subsystem is the one that process all the data communication between the client side and the business layer. It also contains classes to process the information retrieves from the TSO Api regarding the trolleys. All this information is stored and retrieves from a MySql database. . The Favorite subsystems allows users to create, remove favorite stops. In this subsystem users can also activate and deactivate alarms and define a time when they want to receive those. The Alarm subsystem is in charge of sending the emails at the time defined by the users that registered their favorite stops to receive alarms.

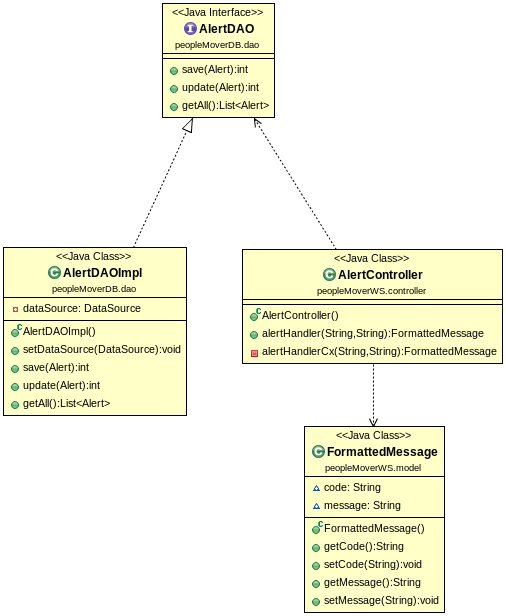
## 3.2 Static model

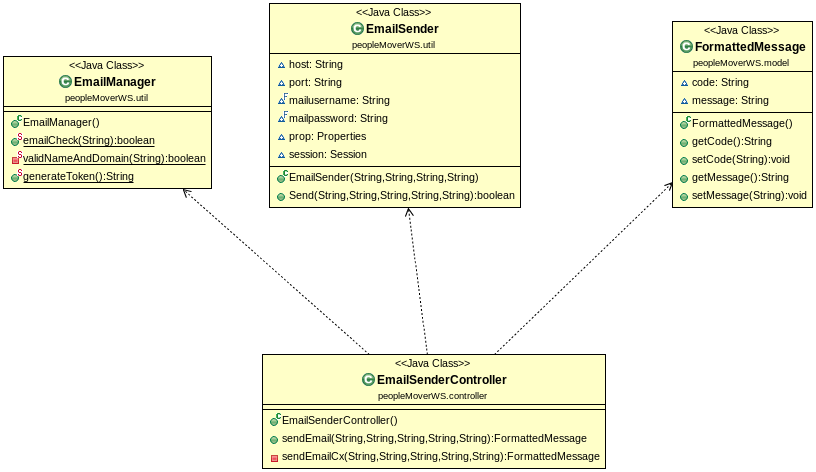
The following models are part of the business logic tier. The User Authentication class diagram is the subsystem that manages the user login, registration, and the information that will be provided to the Client subsystem. The Web Service subsystem is defined by the information requested to the external API, it process that information and is send it back to the client Subsystem.



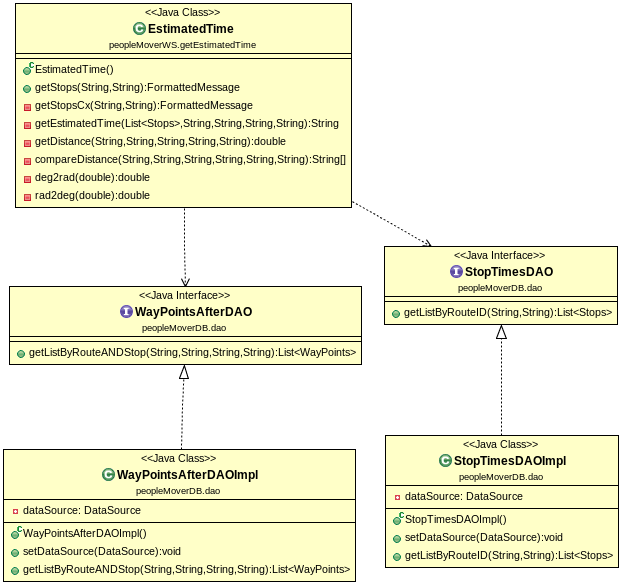
Alarm Service

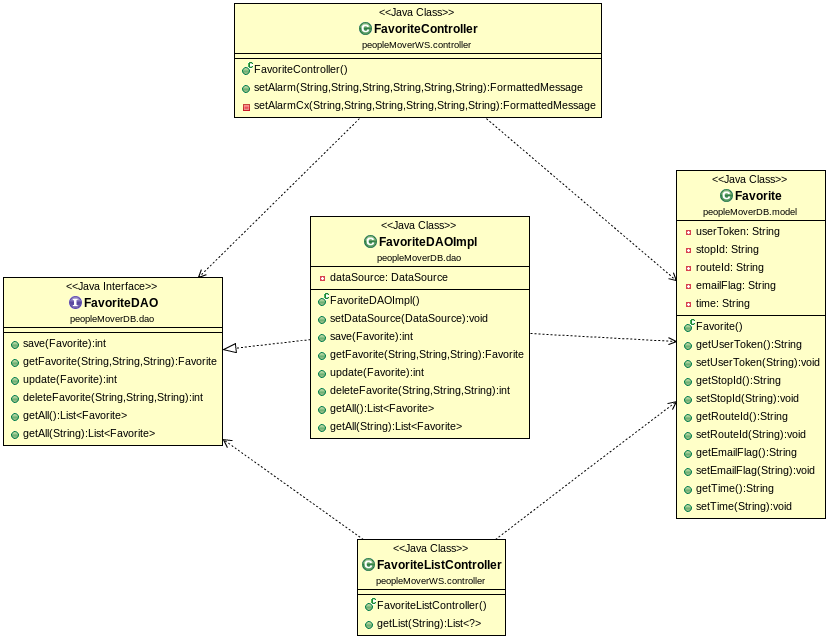
Alert Subsytem

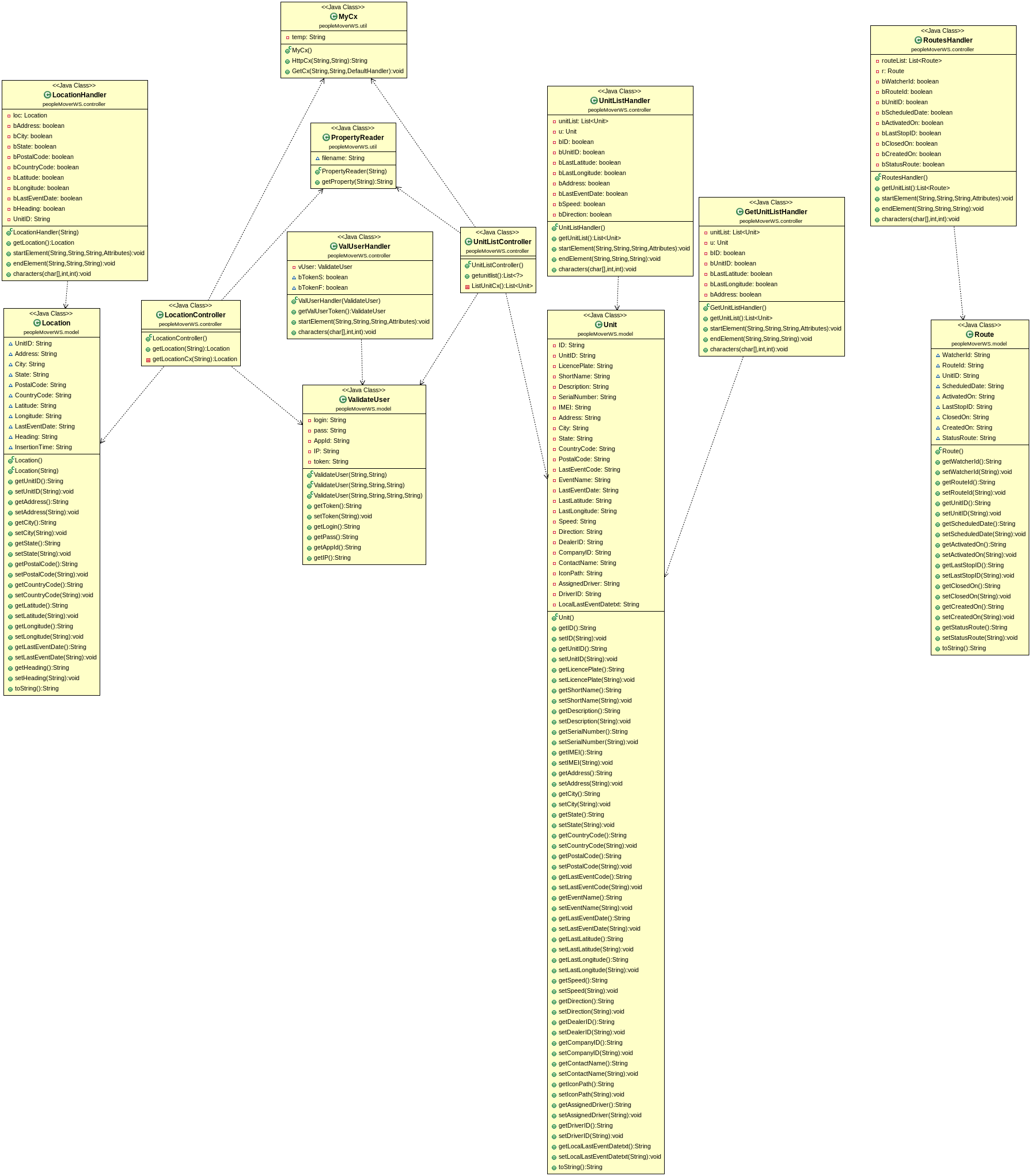


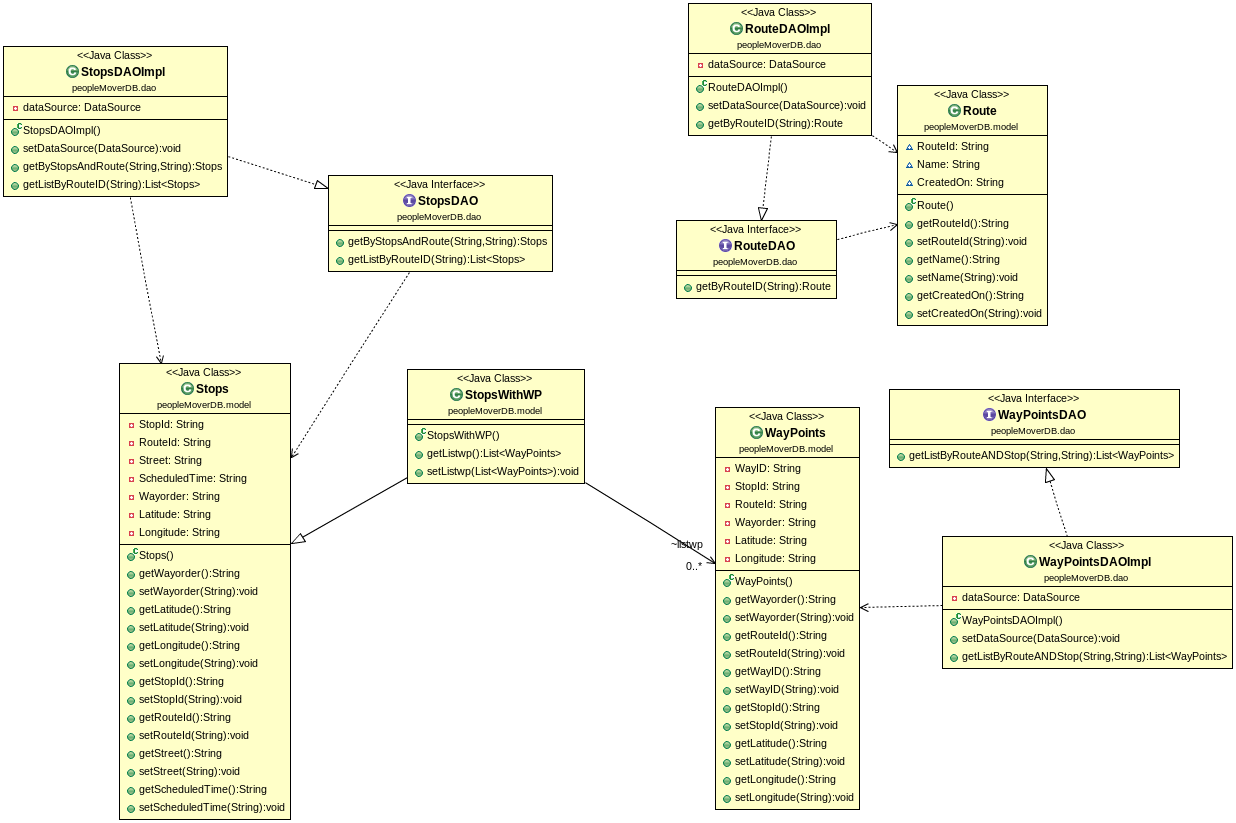
Email Subsystem

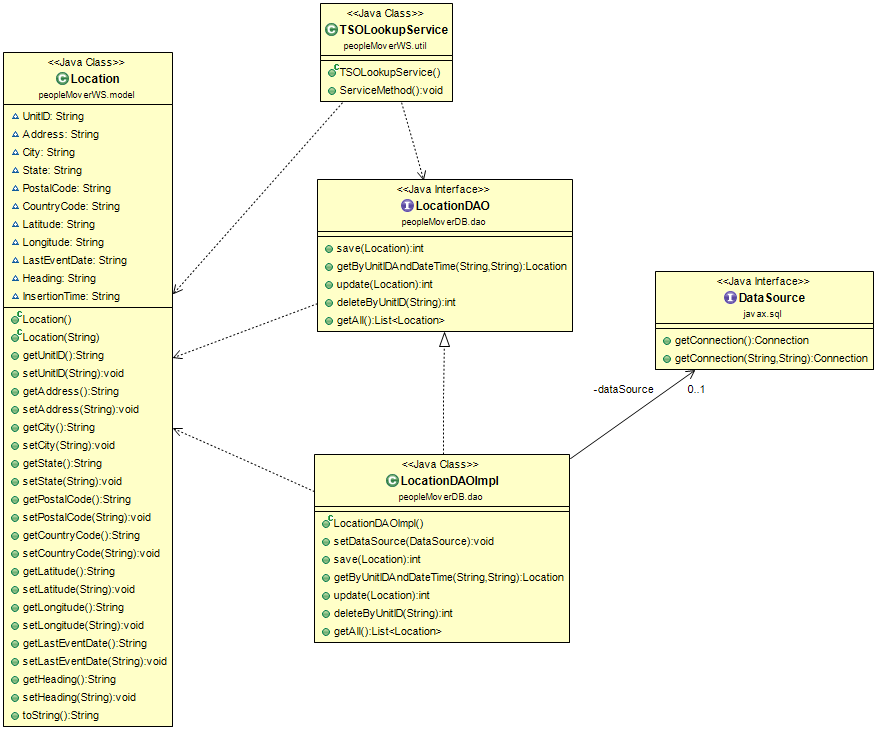
Estimated Time



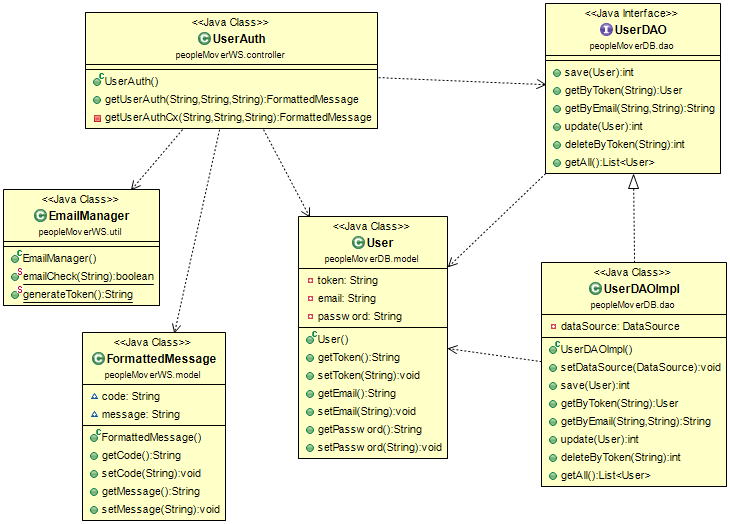
Favorite Definition

Location, Unit and Route

Stops, Route, and Waypoints

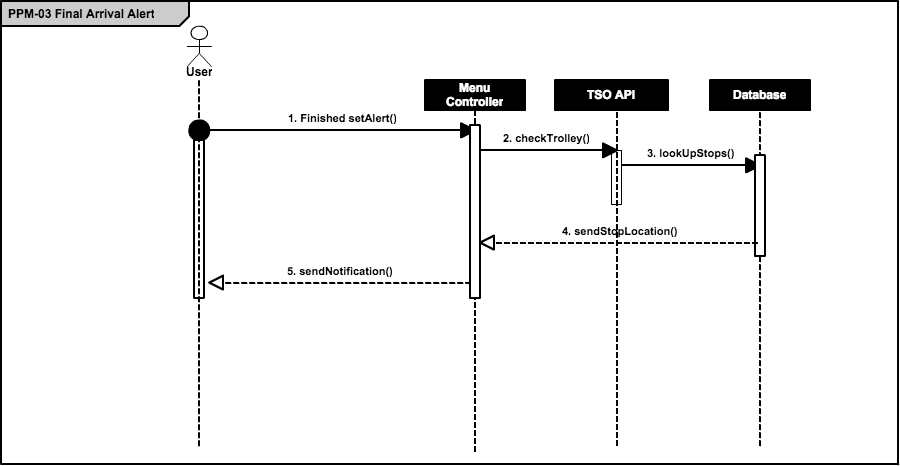
TSOLookUpService (Historical Location)

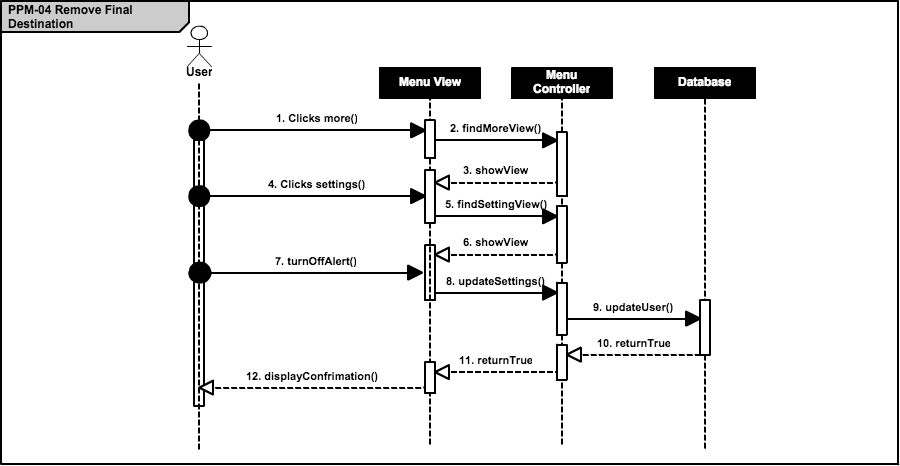
Authentication Service

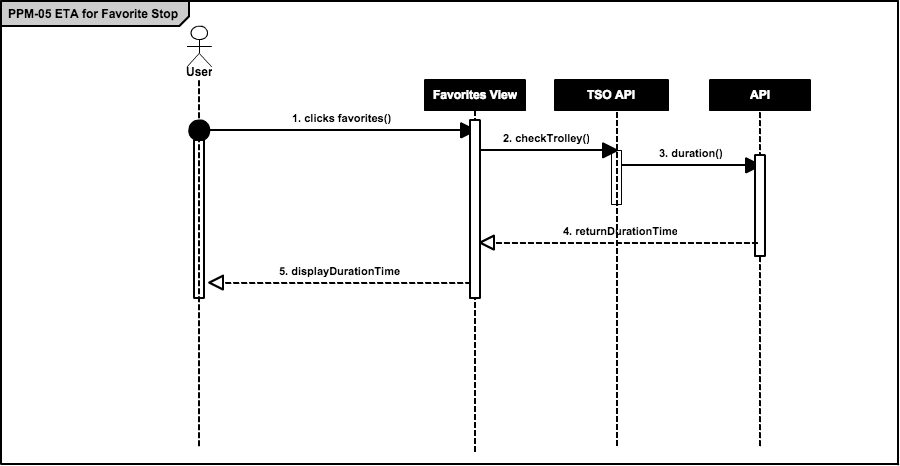


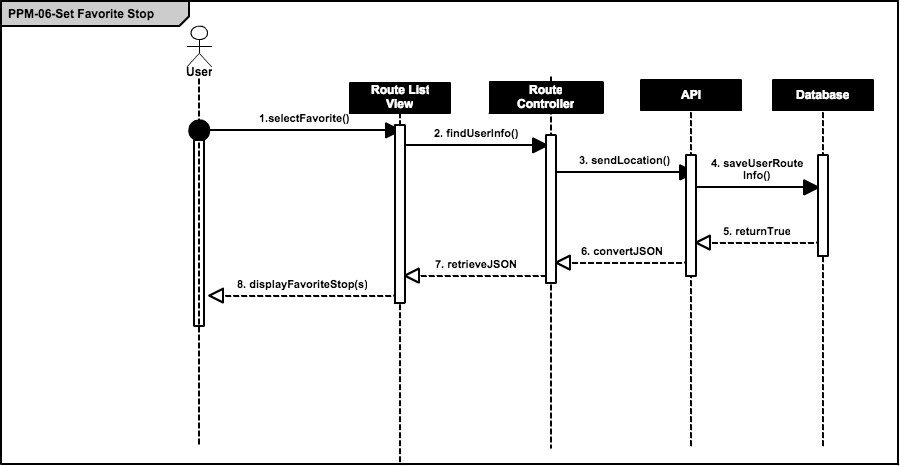
## 3.3 Dynamic model

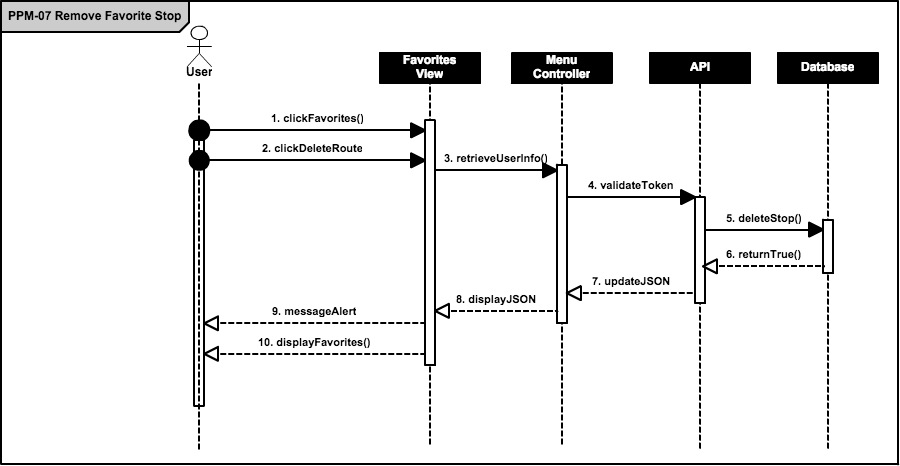


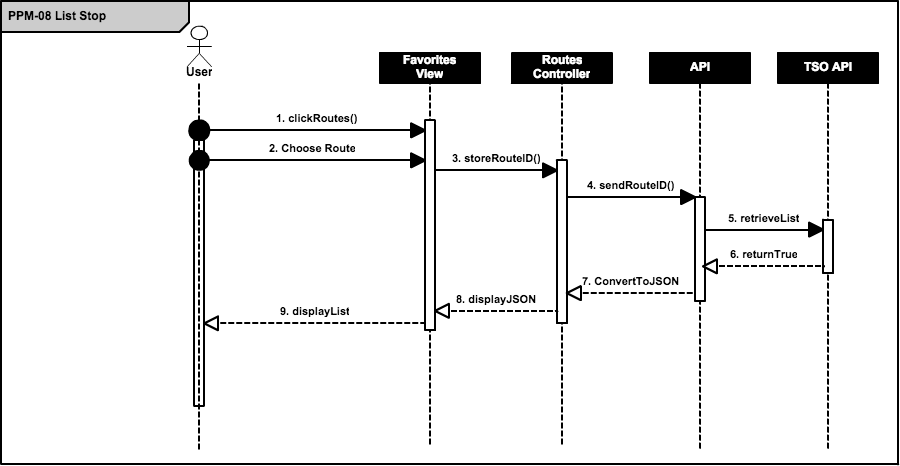


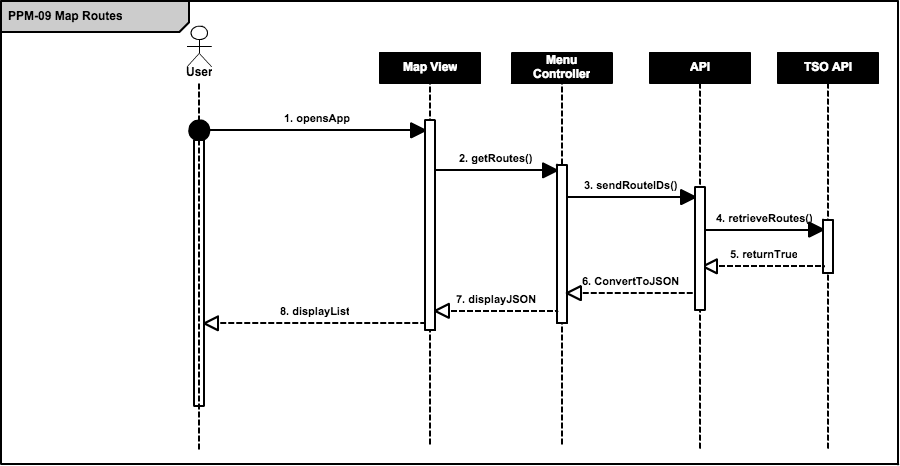


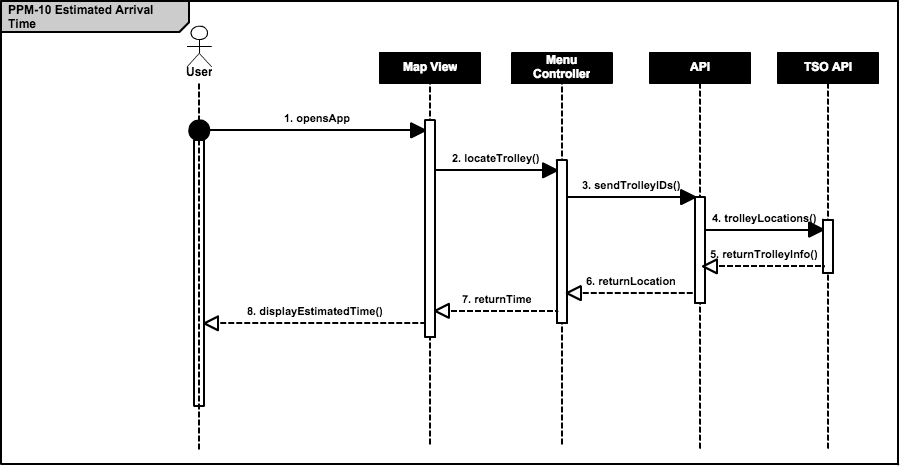


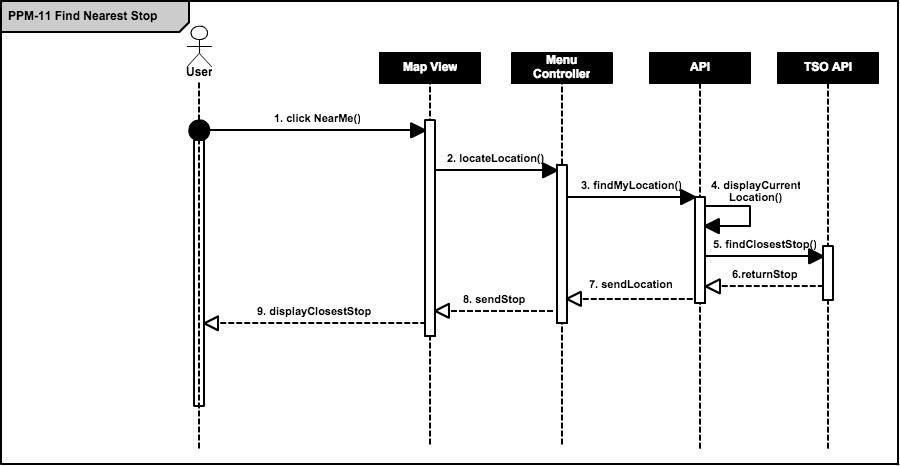


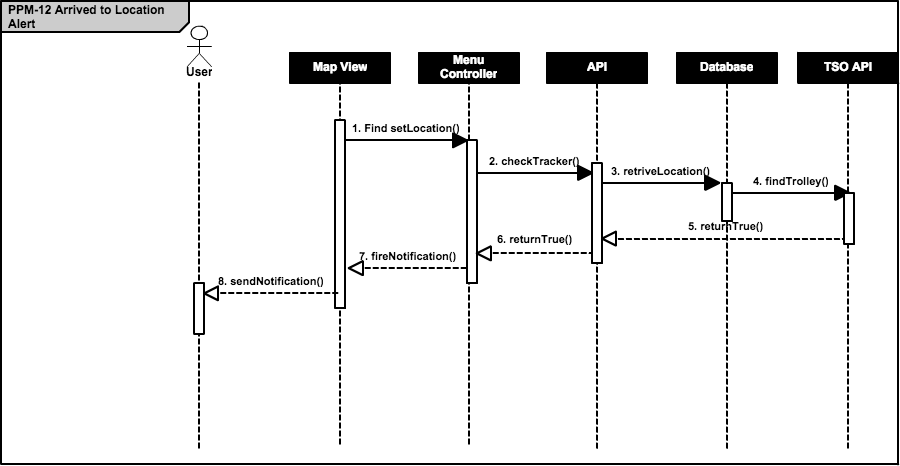


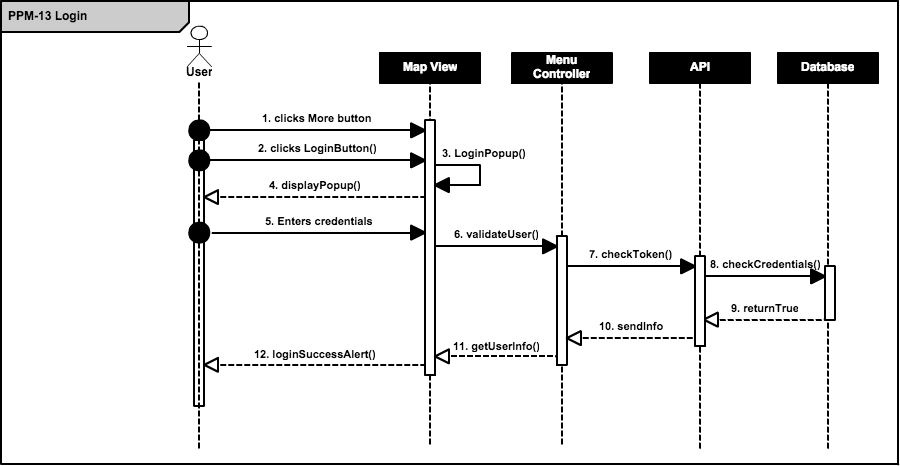


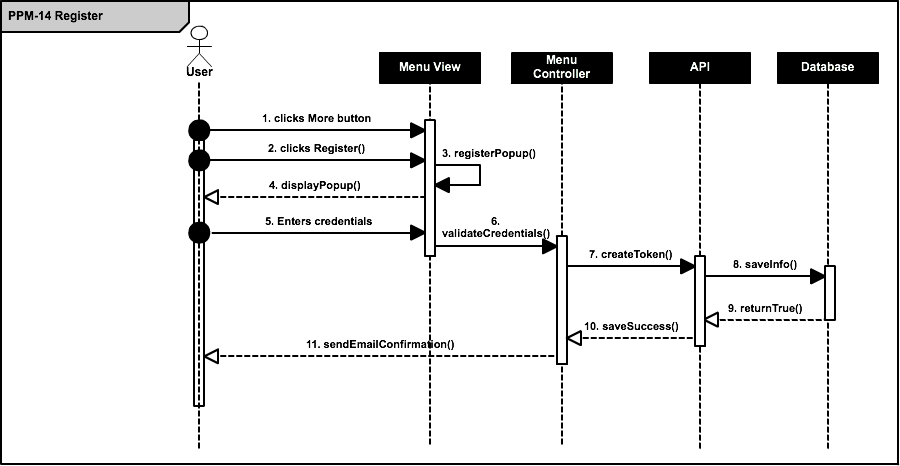


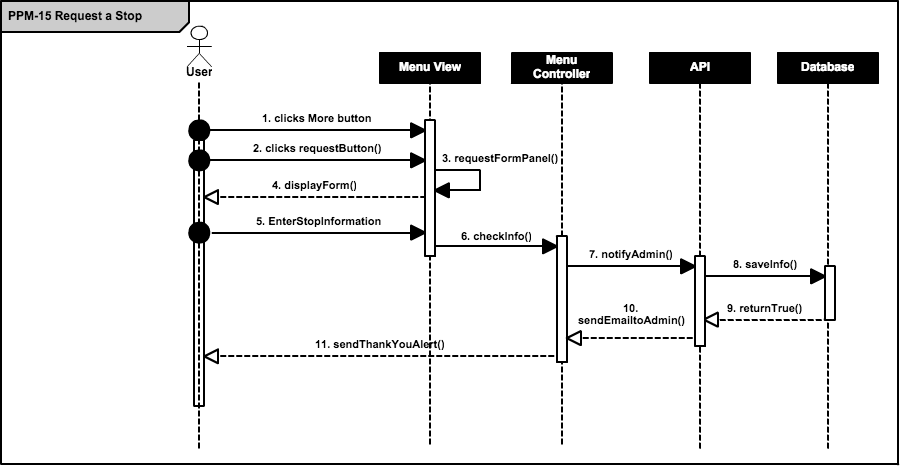












## 3.4 Code Specification

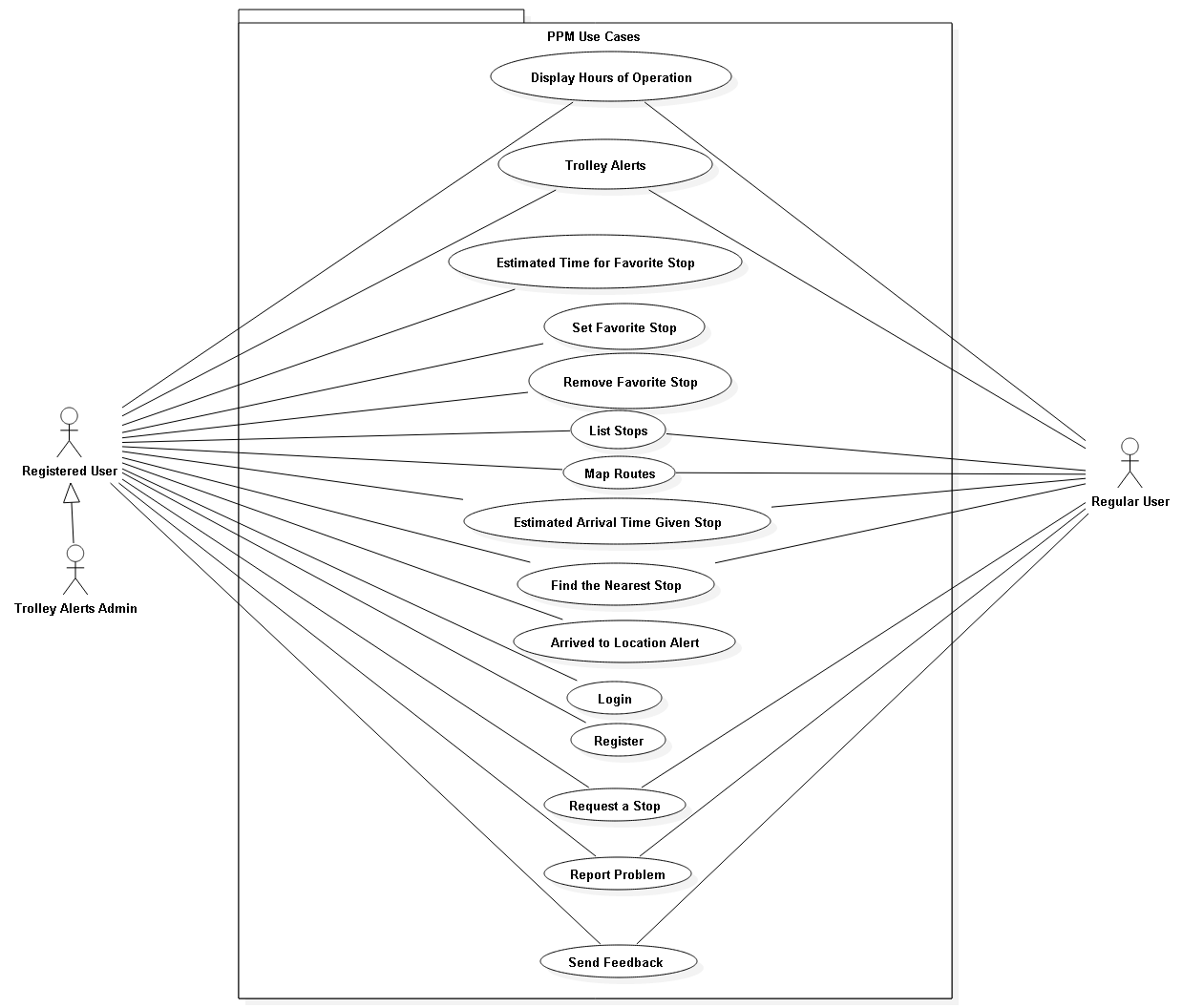
The email subsystem main functionality is set on the filter. As the users interacts with the filters by enabling checkboxes and selecting items in the lists, flags are enabled for each interaction. At the time of displaying the recipients list the setJudgeFilters() and setStudentFilters() functions build another function that is then passed to the store for judges and students respectively. This function will be using the flags previously enabled and state each record in the store to determine if that record has to be included in the filter. When the user deselects either student or judge from the list in the recipient lists step, the sent flag for that record is disabled. Finally on the send step the subsystem fetches the filtered data for students and judges and builds the email bodies by replacing the fields on the email template with the record values. Also any record with the sent flag disabled is ignored. A simple loop iterates over the built emails and calls the PHP API to send the email.

The Live Stats subsystem makes extensive use of the Sencha Touch’s built in data binding features to populate the list, rendered graph, and apply the corresponding grouping. The popup that appear when a user selects an element in the list or graph call on the the itemtap() function. This function will map the element chosen with the corresponding record from the store and process the record for the adequate information to display in the popup.

# 4. Glossary

# 5. Appendix

## 5.1 Appendix A



## 

## 5.2 Appendix B

|  |  |
| --- | --- |
| 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 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 Problem |
| Description | Allow user to send messages to the system. In this case to request a 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 request 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 |

## 5.3 Appendix C– Documented Class Interfaces

## 

## 5.4 Appendix D

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 |
|  |  |
|  |  |

# 6. References

1. Glossary - define terms used in document, especially domain specific terms.
2. Appendix
   1. Appendix A - Use case diagram for use cases being implemented.
   2. Appendix B - Use cases being implemented (from the RD).
   3. Appendix C – Documented class interfaces (code) for the subsystem(s) you will implement and the constraints.
   4. Appendix D - Diary of meeting and tasks.