

Contents

1.	Introduction.....	3
1.1.	History.....	3
1.2.	Current Scenario	4
1.3.	Problem Statement	5
1.4.	Aims and Objectives	5
1.5.	Structure of the Report.....	6
2.	Background.....	7
2.1.	Project Elaboration.....	7
2.1.1.	Project Overview.....	7
2.1.2.	Project Scope and functions	8
2.2.	Analysis of Similar System.....	23
2.2.1.	CTA Bus Tracker.....	23
2.2.2.	NextBus.....	24
2.2.3.	Moovit.....	25
2.2.4.	Conclusion of Analysis of similar system.....	26
3.	Development.....	27
3.1.	Solution consideration	27
3.1.1.	Big Bang Model.....	27
3.1.2.	Agile Model	27
3.1.3.	Enhanced Waterfall Methodology	28
3.1.4.	Chosen Methodology	29
3.2.	Phases of Enhanced Waterfall Methodology	30
3.3.	Requirement Analysis Phase.....	33
3.3.1.	SRS Document.....	33
3.4.	System Design Phase	45
3.4.1.	User Case Design	45
3.4.2.	Architecture Design	76
3.4.3.	Process Designs	77
3.4.3.1.	Communication Diagram.....	77
3.4.3.2.	Sequence Diagram	88

3.4.4.3. Activity Diagram	100
3.4.5.4. Class Diagram.....	112
3.4.6.5. Database Design	113
3.5. Development Phase.....	124
4. Testing.....	134
4.1. Introduction.....	134
4.2. Scope.....	134
4.3. Out of Scope	135
4.4. Test Assumptions.....	135
5. Ethical, Legal and Social Issues.....	162
5.1. Ethical Issues	162
5.2. Legal Issues.....	163
5.3. Social Issues.....	164
6. Conclusion and Recommendations.....	165
7. Reference	170

1. Introduction

In present world, technology is ubiquitous and there are many innovations and invention which has changed our way of living. One of the field where technology is advancing is **Navigation**. Navigation is a process of controlling movement of craft or vehicle from one place to another. The art of Navigation was developed about 4000 years ago by Phoenicians sailors for routing their ships using primitive charts and observation of the sun and stars (Gautam, A., 2011). Today, there are various computerized navigation system such as TomTom, Magellan SmartGPS etc. A navigation system is a computer program that gives location information represented with a graphical map (WiseGeek, n.d.). In present, various private and public vehicle company use these computerized systems in order to provide navigation service to passengers.

1.1. History

Satellite Navigation System is one of the type of navigation system. It is a system of collection of satellites that provides geolocation information such as latitude, longitude etc. The first satellite navigation system called TRANSIT was built by the US Navy in 1959 primarily designed to locate submarines (Sullivan, M., 2012). Today, Global Navigation Satellite System (GNSS) is widely used satellite system. It is of two type: Global Positioning System (GPS) and Global Satellite Navigation System (GLONASS) and these are the only global navigation systems in existence today (Croslow, A., 2013).

GPS is a commonly used global navigation system. GPS system is composed of 24 synchronized atomic clock equipped satellites which is orbiting around earth and the atomic clock emit radio signal at speed of light as a result it locate position in earth with exact time and high level of precision (Schutz, Y. & Chambaz, A.,n.d.). Initially, GPS was designed for military and intelligence application purpose. The concept of GPS was invented by the idea during 1957 by MIT Scientists. In 1974, U.S. launched first GPS system called NAVSTAR which is composed of 24 satellites. In 1999, first commercial GPS mobile phone was launched and during 2001, GPS system is implemented in vehicle navigation such as Car navigation system and Tom-Tom. (Sullivan, M., 2012). Today, GPS is highly available in every commercial and non-commercial products. In addition, the use of GPS in phone and vehicles such as cars, bus etc. is getting increasingly popular.

Nowadays, every smartphone is embedded with a GPS navigation system (WiseGeek, n.d.). The popularity of smartphone has become so higher that almost everyone has a smartphone today. The evolution of smartphone started from 2007, when Apple launched its first iPhone with its own operating system iOS and in November 2007, Google announced a new mobile operating system called Android and the first Android smartphone called G1 was launched in 2008 (Arthur, C., 2012). At present, there are wide range of Android smartphone in the market and variety of navigation application can be found that run on these smartphone.

The google map is one of the leading and popular map service. According Chivers, T. (2013), google map has mapped 28 million miles of road which includes 194 countries since 2005. Initially, when the Google map service started in 2005, it provides limited services for navigation including satellite image. Today, google map provides various features such as finding best route, turn by turn navigation, and automatic re-route according traffic jam etc. (Google, n.d.).

1.2. Current Scenario

In current scenario, we can find various navigation application for smartphone and most popular of them are Waze, Navmi GPS world and Maps.me etc. (Favre, L., 2015). With growth of smartphone user, the use of these application is also growing. In addition, most of the public and private vehicle are equipped with vehicle tracking systems. For instance, we can find a car navigation system such as TomTom which provides turn-by-turn navigation to the user. In context of Nepal, the use of GPS navigation and tracking service is minimum while there are some company that provide GPS tracking and navigation service for private vehicles such as Telematics Tech, GPS Nepal etc.

On the other hand, most of the organization are starting to use online system for their business. Due to the development in the ICT, the online activities has become more reliable and convenient. In past, the computerized system was not popular and there was lack of computer infrastructure such as reliable internet etc. Nowadays, various organization are promoting their services and business online. Currently, Sajha Yatayat do not have an online system but they are willing to promote and provide their service online and the company only have a static website to provide various information about their vehicles.

The current public transportation service of Nepal is very poor. The common issues are irregular traffic situations, unpredictable arrival times and schedules etc. As a result, the current travelling experience of public passenger is very poor. To support this fact, I have conducted a survey to public people. According to the survey, it is found that most of the people are using public vehicle as their primary travelling option and most of them are facing various problems. Most of the people have problems in finding right information about vehicle such as route details fare price etc. Around 80% of passenger are facing problem due to uncertain time and schedule of public vehicle. On the other hand, most of the people use smartphone and almost all of them have internet service on their smartphone. I have found that currently only very few people uses smartphone application to improve their travelling experience but more than 80% of people are willing to have an android application where they can get various information such as bus location, fare, estimated arrival time etc. Also, more than 90% of people prefer to have an online payment system in order to pay their fare during their ride in bus.

1.3. Problem Statement

Currently, Sajha Yatayat has problem with tracking and monitoring their all route's vehicles in real time and managing vehicle information such as fare amounts, routes, schedules etc. In addition, the current public transportation service of Nepal is unreliable and unsatisfactory and passenger experience is very poor. Most of the time, the arrival time of vehicles are not accurate and unpredictable. Therefore, my client require an online system to track and manage all the operational vehicles on their current routes and provide a vehicle tracking service to public passengers via smartphone application.

1.4. Aims and Objectives

The aim of this project is to enhance and maximize the current vehicle monitoring and management facility of Sajha Yatayat by developing an online web portal system. In addition, this project is also aimed to solve current transportation related issues and problems that passengers are facing by developing an Android application which helps them to find required vehicle around them and provides various information.

Objectives:

- To identify current transportation related issues of Nepal by undertaking a survey to at least 25 people within April 20th 2015.
- To design database of the system using oracle designer and implement the database using oracle DBMS during the development phase.
- To develop a fully functional system (both web portal and android app) by Second week of December.
- To perform unit test and integration testing to the developed system using testing tool Junit by end of December.
- To deploy the completed system in the client's system after successful completion of this system at the end of January.

1.5. Structure of the Report

So far, the introduction section of this report has been accomplished where various information related to the project such as aims and objectives, problem statement etc. has been discussed. Therefore, the remaining structure of the report has been divided into following sections:

1.5.1. Background

In this section, project will be further elaborated. The overview of the system will be discussed and various system features will be discussed along with its wireframe. In addition, similar system will be analyzed and compared with the proposed system. Also, future enhancements are included.

1.5.2. Development

In this section, various methodologies will be comparatively discussed that could be used for the development of the new system and then all the tasks for each phases of the selected methodology will be described. Also, I will be describing current engaged phase of this project.

1.5.3. Testing

In this section, the system is tested. Various test plan has been produced and test cases are executed.

1.5.4. Ethical, Legal and Social Issues

In this section, various issues related to this project has been identified. In addition, solution to those issue has also been described.

1.5.5. Conclusion

In this section, the report is critically evaluated. In addition, various learning from this project has been identified and what went wrong and solutions to those problem has been identified.

2. Background

2.1. Project Elaboration

2.1.1. Project Overview

The main purpose of the system is to make an application where user can track and monitor public vehicle and their time. The proposed system will comprise of a centralized main system with web portal, an Android App and driver app. The system will have mainly 3 users; client's user and passenger and driver. The client's user will be using the web portal in order to interact with the system and perform various tasks such as adding new vehicle information, fare information etc. On the other hand, the passengers will be using the Android App in order to perform various tasks such as tracking required vehicle, getting estimate time, e-payment and other various tasks. Similarly, the driver will be using driver app in order to update vehicle location.

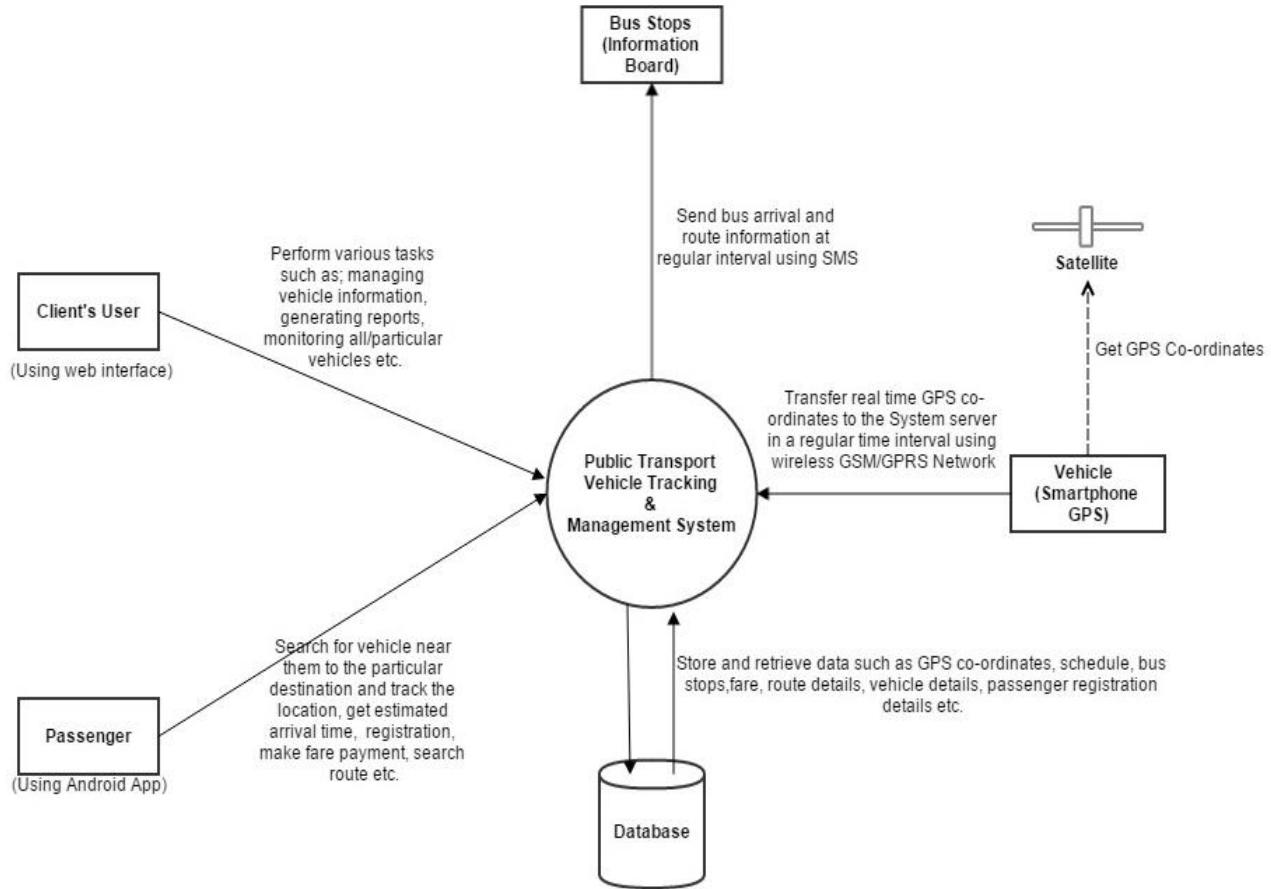


Figure: Overview representation of Proposed System

2.1.2. Project Scope and functions

2.1.2.1. Passengers (Using Android App):

2.1.2.1.1. Search for particular transport vehicles:

The passenger can search for the particular vehicle lists of required destination of their choice. The user can either search using their location or browse manually. The 'Near You' selection will enable to search vehicle around user's current location.



Figure: Home page (main search page)

2.1.2.1.2. Get estimate arrival time, fare amount and other information:

When user search for particular vehicles, then various information such as estimated arrival time, fare amount, departure time, distance, nearest bus stop for that vehicle, vehicle number etc.

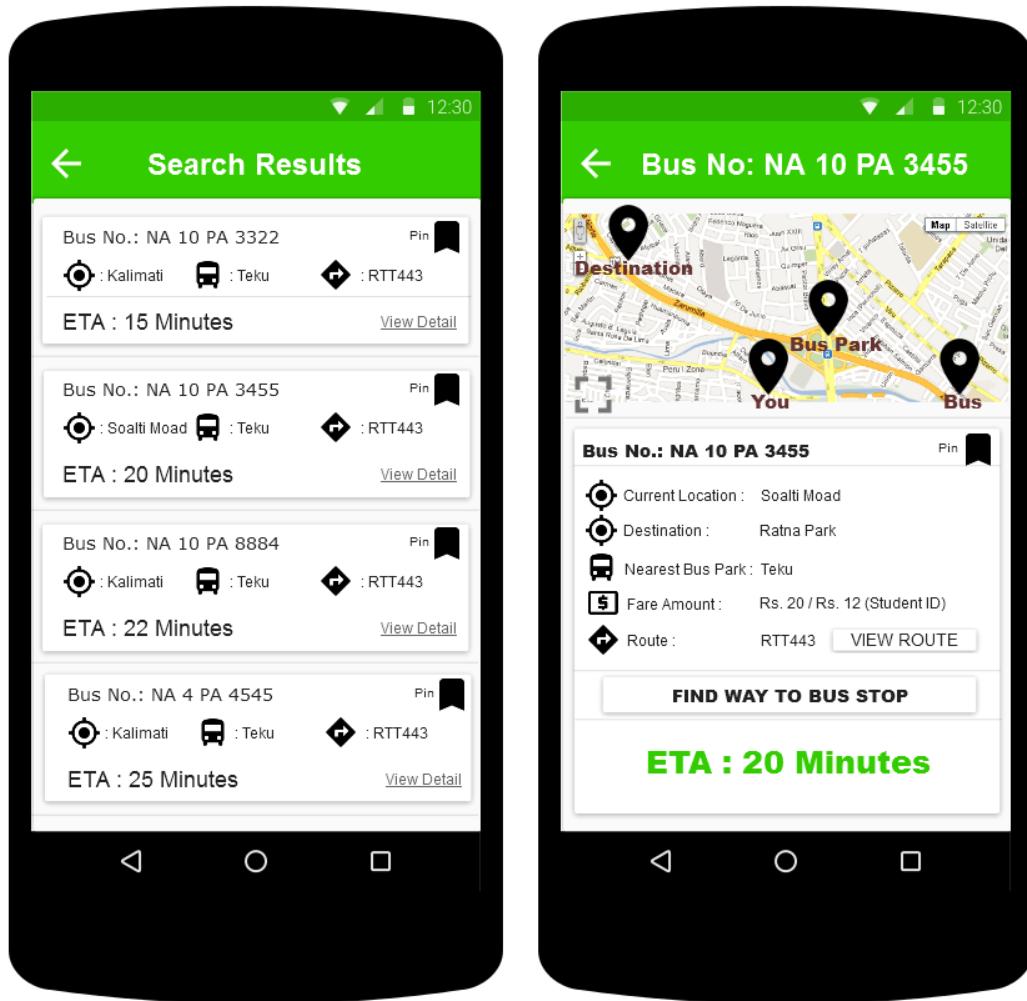


Figure: All available bus display

Figure: Detail of Selected Bus

a) Choose and track your required Vehicle:

After the list of all available vehicle is displayed, user can select their required vehicle in their active status. The user need to ‘Pin’ the required vehicle from the search result and then that vehicle will appear in ‘Status’ section and remains until user ‘unpin’ it.

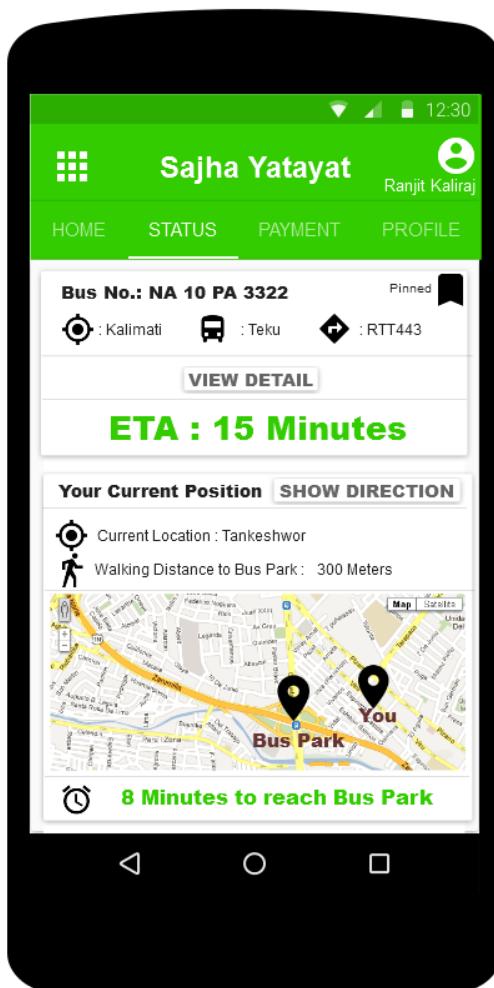


Figure: Pinned bus from the search result

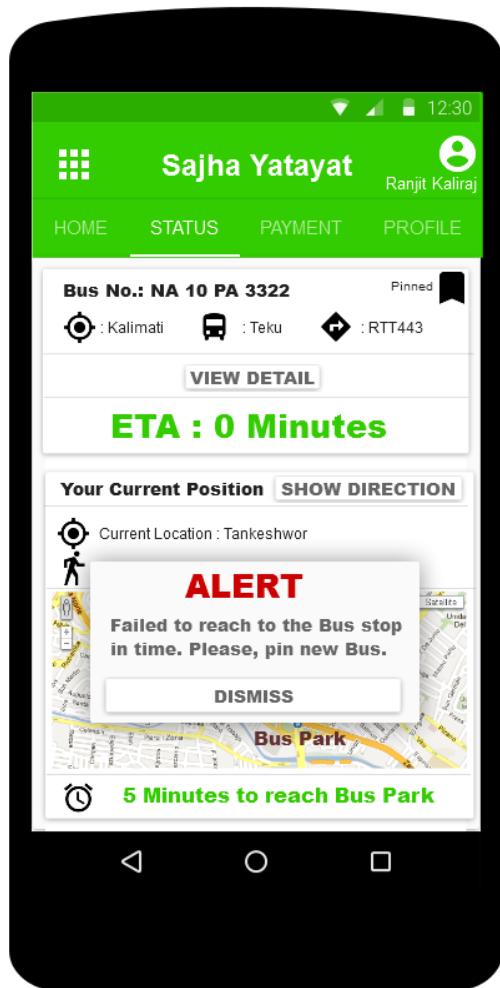


Figure: Time out: user could not reach in time

b) Get walking distance and direction to Bus stop:

After user select their required vehicle, this app will automatically provide walking distance and step by step direction to the nearest bus stop to the user. User can get step by step navigation to the bus park.

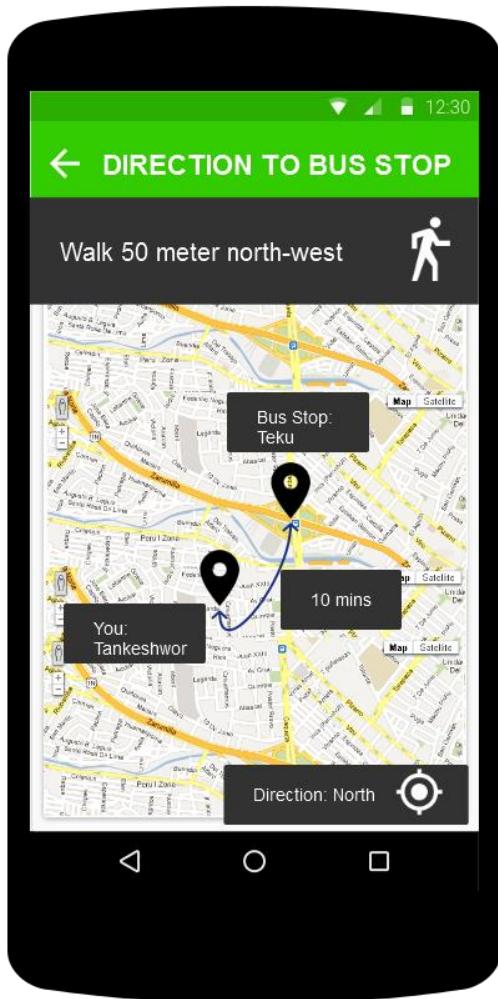


Figure: Getting walking direction from user location to bus stop

c) View vehicle location in Map Mode:

The passenger can also view the vehicle in a map view. The map will display the current location of the vehicle with its required information such as arrival time to the bus stop, passenger's current location, and location of the bus stop, walking way to the bus stop, destination location, and length from bus stop to the destination.

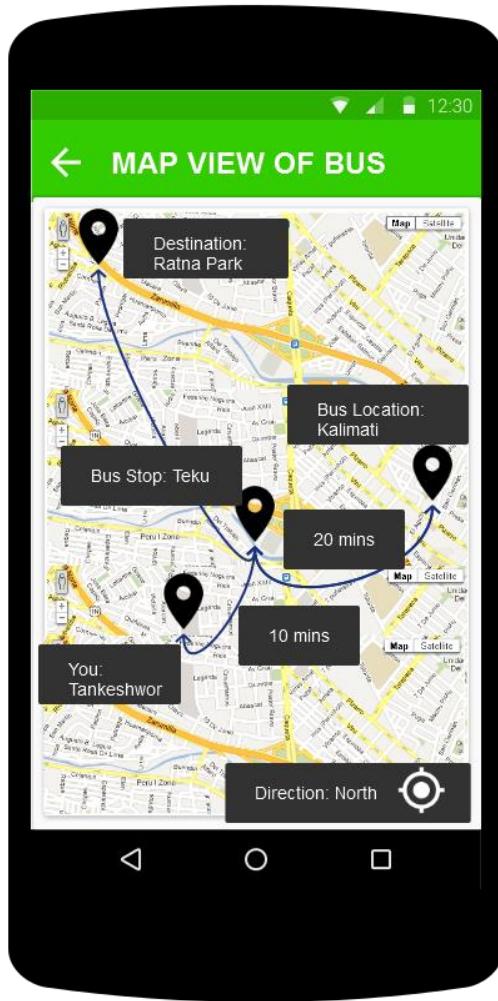


Figure: Map view of selected Bus

d) View route details:

Passenger can view the route detail of particular vehicle. When user select particular vehicle from the search result, then there is an option to view route of that selected vehicle and selecting that will display following interface.

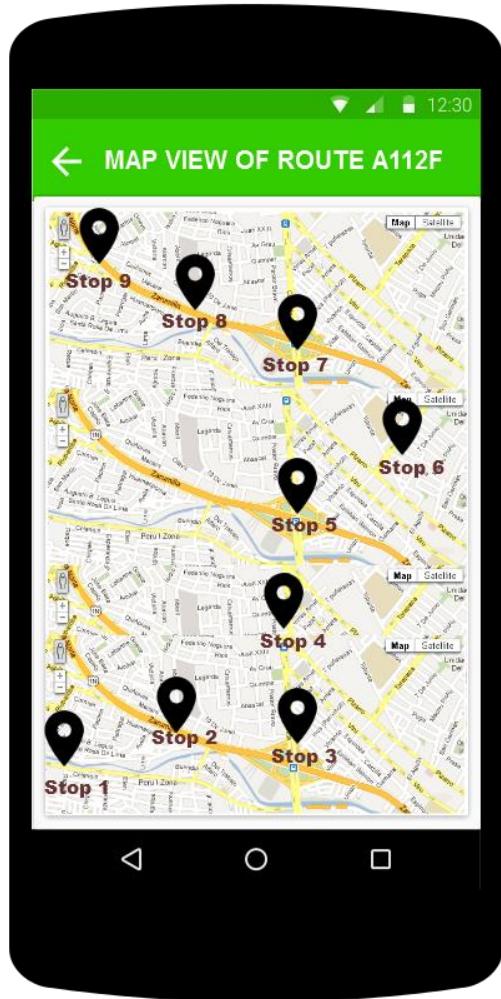


Figure: route view of selected Bus

e) Sign up for registration:

Passenger can register for the account using their email address. Registration is very important for various features such as online payment, share/track riding location etc.

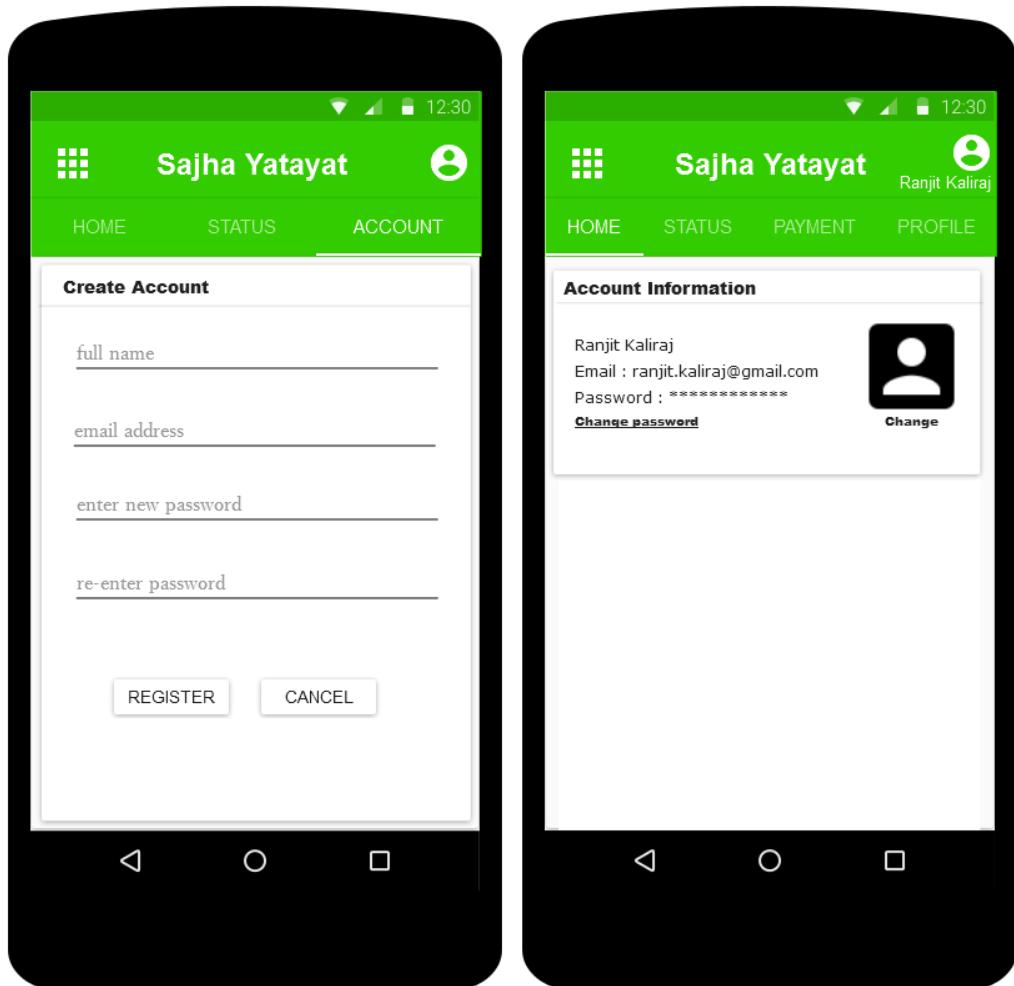


Figure: Registration form and profile overview

f) Online fare payment:

Online payment can be made using E-Sewa service from their smartphone while they are travelling in the bus in real time. When user pin the bus, user can pay fare for that bus. The payment option remain unavailable when no buses are pinned or selected.

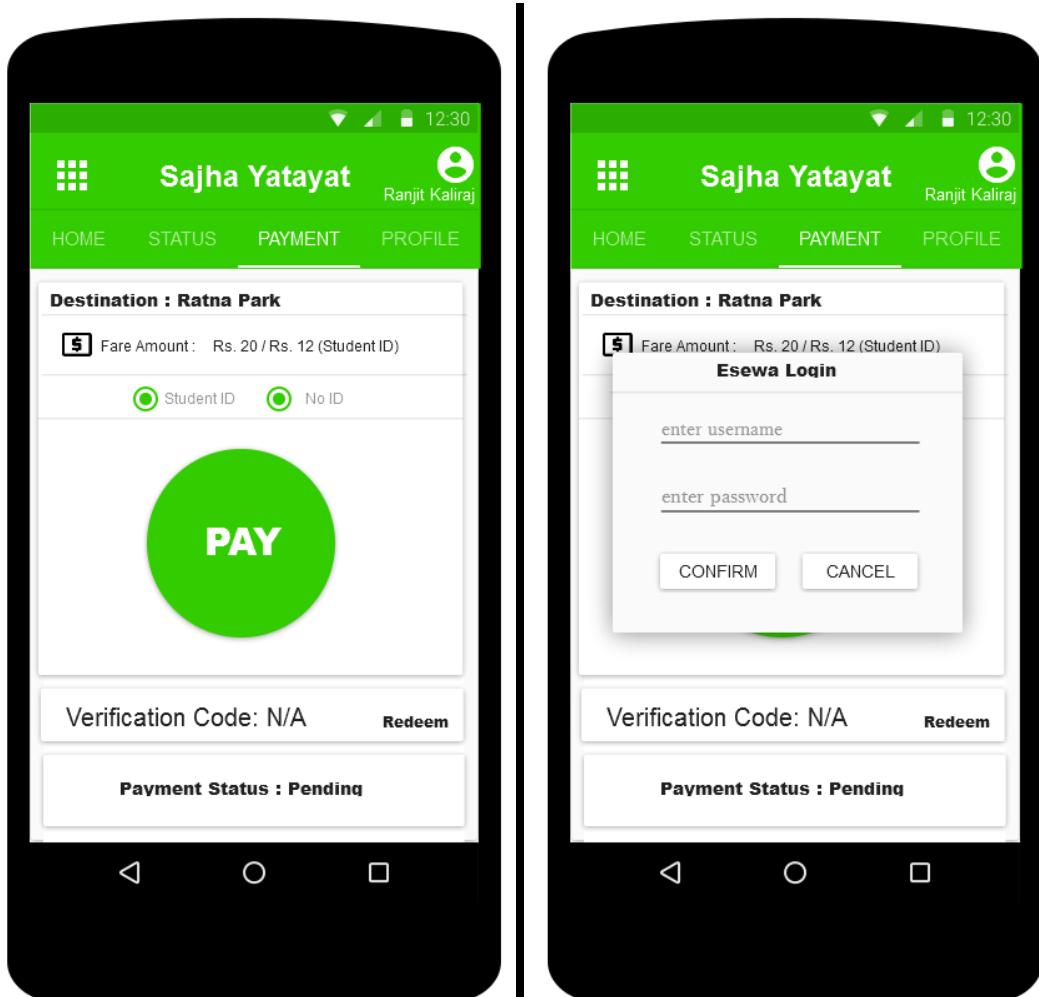


Figure: Payment Option

Figure: E-Sewa login page

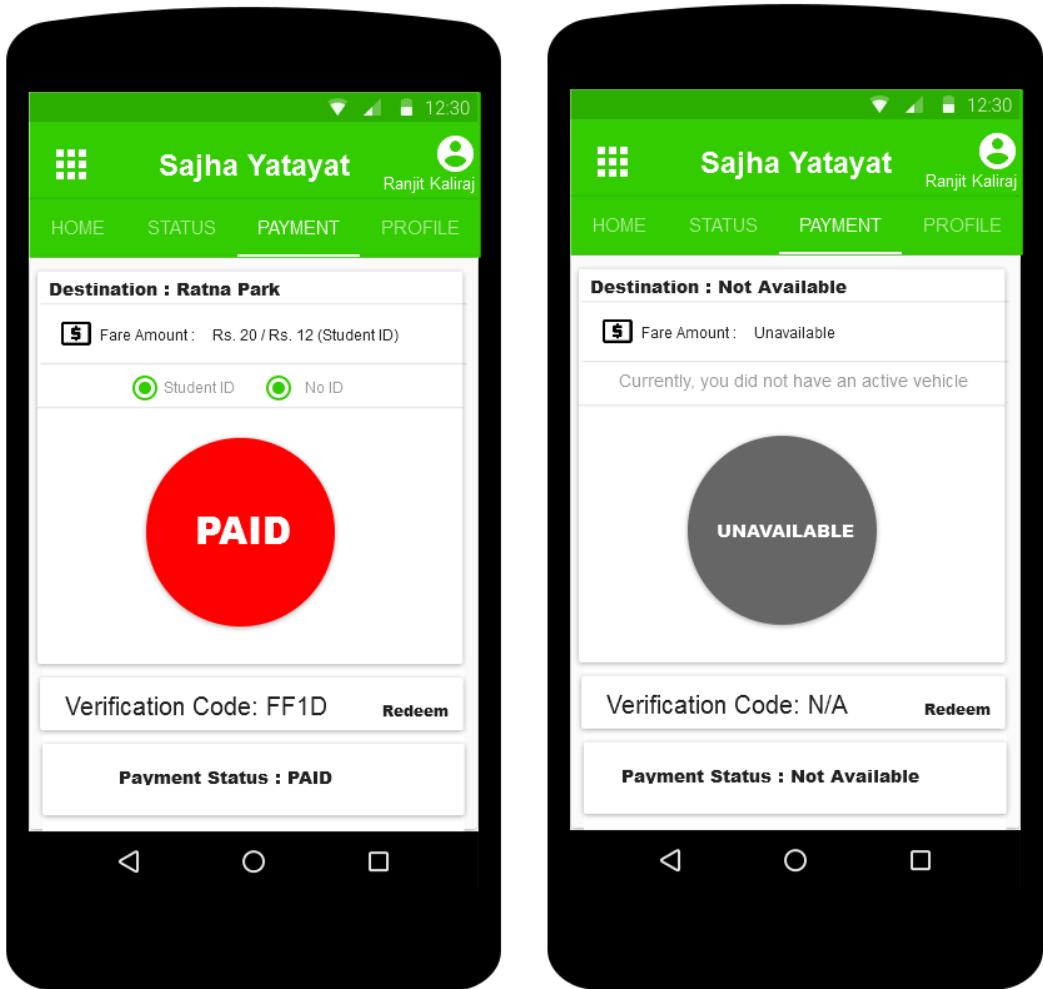


Figure: Payment paid status

Figure: Payment option not available

g) Queries:

Queries can be made for any complaints and question directly to the bus operator using this app. Queries can be made by member or non-member passenger.



Figure: Query submit form

Client (Using Web interface):**a) Manage Vehicle and other information:**

The client's user can manage various information such as add, edit and delete. Following are list of information that need to be managed by this system.

- Vehicle information
- Route information
- Fare information
- Schedule information

The wireframe shows a web interface for adding vehicle details. At the top left is a placeholder for a logo. To its right, the system title is displayed: "Sajha Yatayat Public Transport Vehicle Tracking & Management System". On the left side, there is a vertical menu bar with a grey background. It contains a "Menu" section with four items: "Manage Vehicle Information", "Manage Route Information", "Manage Fare", and "Manage Schedule". Below the menu is a placeholder box with dimensions "176 x 100" and a diagonal cross pattern. The main content area has a white background. It starts with a section titled "Add Vehicle Details" which includes five input fields: "Bus No.", "Route", "Driver", "Schedule", and "Destination", each with a "Text" label. Below these fields is a blue rectangular button labeled "Add Details". The entire wireframe is enclosed in a dark blue border.

Figure: Wireframe of adding new Vehicle Details.

b) Generate Reports:

The client's user can generate various reports. The reports required to generate by this system are following.

- Route with most passenger Report
- Route with most traffic Report
- Time where most passenger ride bus Report
- Vehicle, schedule, fare and route Details Report
- Passenger Details Report
- Bus with highest and lowest travelling time Report

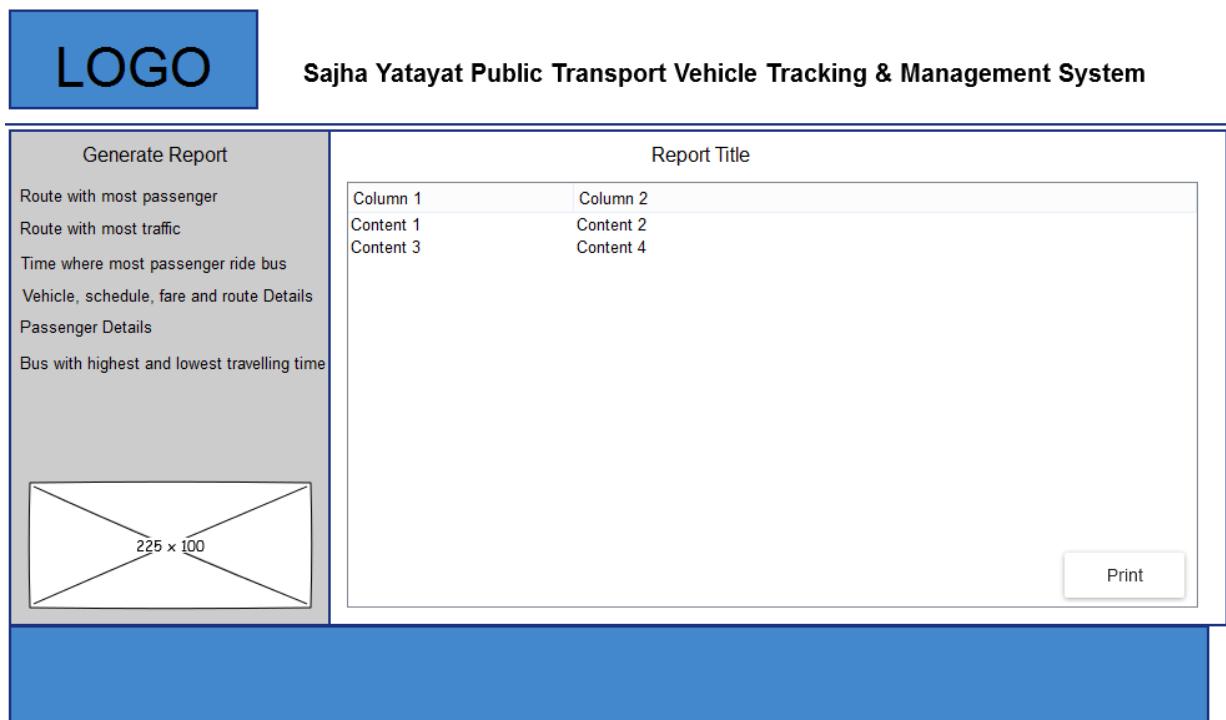


Figure: Wireframe of Report generation

c) Track and monitor vehicles:

The client's user can track and monitor particular or all vehicle of particular or all routes. User can view vehicle location in form of map view and list view. In addition, user can view traffic situation of particular vehicle.



Figure: Wireframe of Tracking and monitoring vehicle in map view

d) Flag particular vehicle:

The user can flag the particular vehicle and disable it. For example, in case of vehicle is inoperative then user can disable that vehicle so that it will not display in search result. In addition, the vehicle with traffic situation will be automatically flagged.

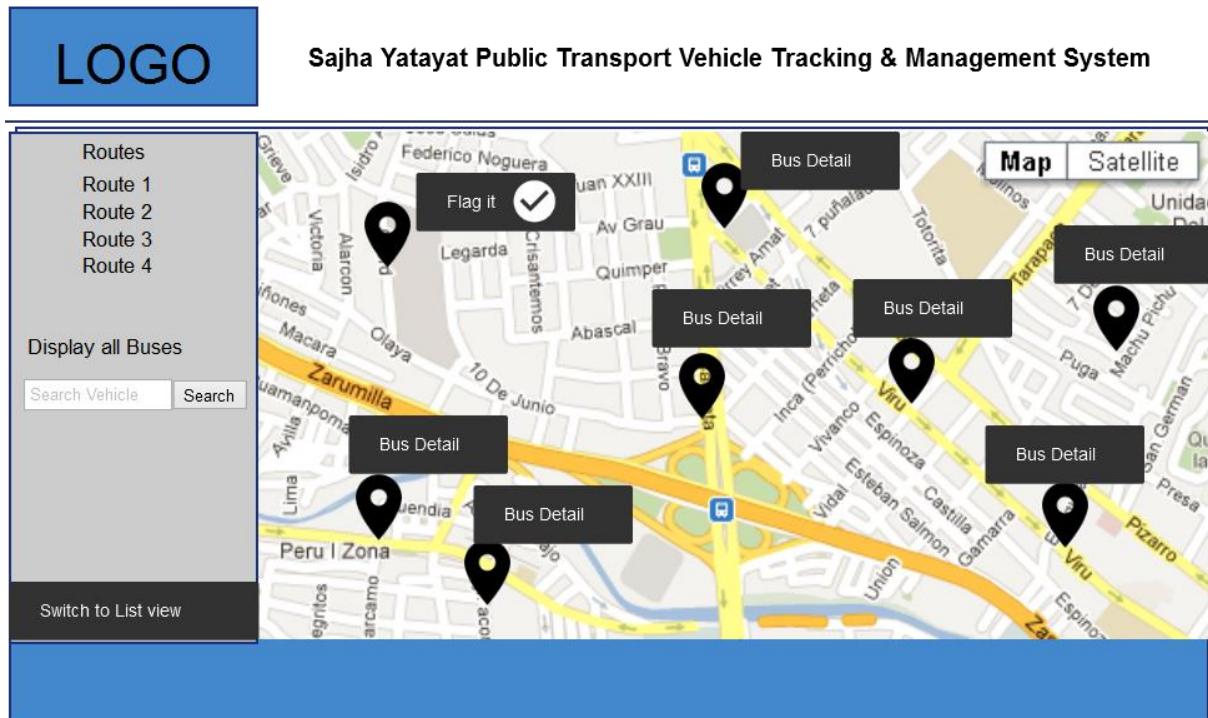


Figure: Wireframe of flagging particular Bus

2.2. Analysis of Similar System

During the research of this project, I have found several vehicle tracking system similar to the proposed system. Following are the list of the reference systems.

2.2.1. CTA Bus Tracker

CTA stands for Chicago Transit Authority. CTA Bus Tracker uses GPS devices to report bus location data (and more) back to our servers and as a result can show where buses are on a map and estimate when they will arrive at user's stop. This system provides various services such as bus estimate time, bus map, get arrival time as text message and subscribe where user can receive arrival and alerts by email or text message (CTA Bus Tracker, n.d).

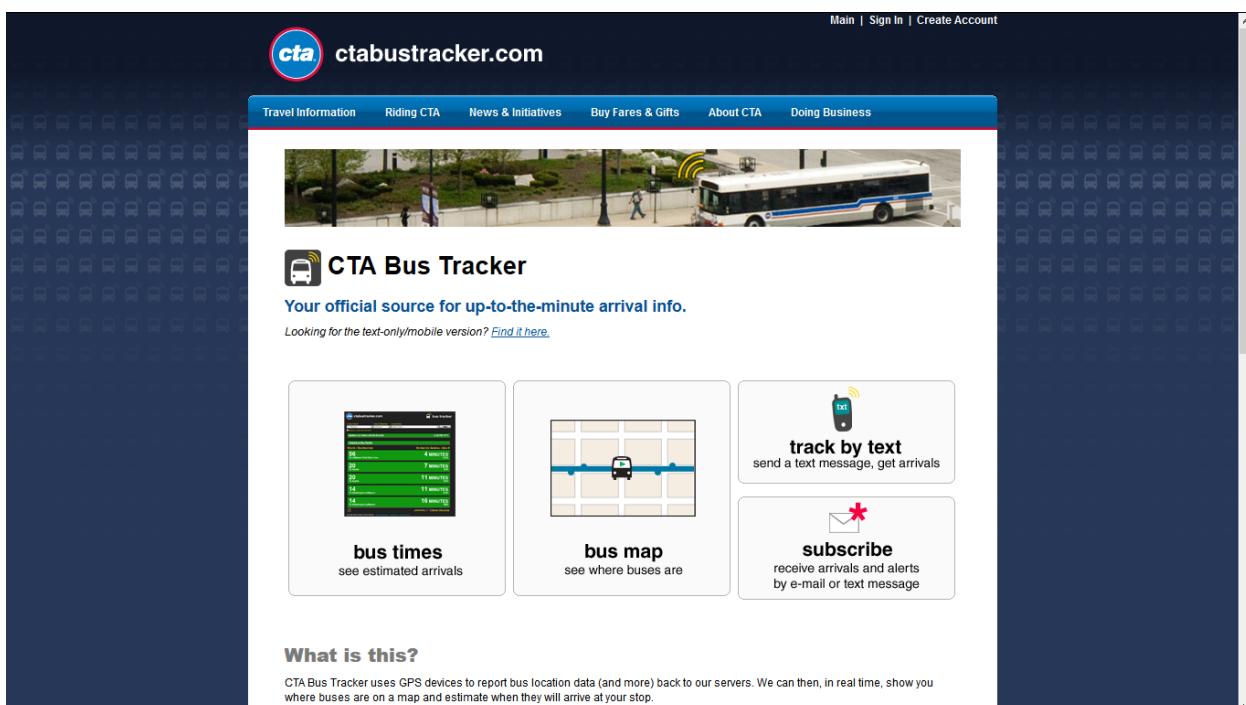


Figure: Screenshot of **ctabustracker.com** (CTA Bus Tracker, n.d)

2.2.2. NextBus

NextBus is a cloud-based solution that provides real-time passenger information systems to over 135 transit agencies and organizations. They serve more than 300 million riders each year and offer a host of options, including a GPS-enabled website for mobile devices that instantly recognizes the nearest stops in proximity order. They also offer an easy-to-use website that is compatible with a JAWS (Job Access With Speech) screen reader and meets ADA (Americans with Disabilities Act) guidelines. NextBus information is currently available in English, French, and Spanish. NextBus uses a proprietary algorithm that takes into account the vehicle's location and historical travel data to provide up-to-the-minute arrival time information (NextBus, n.d.).

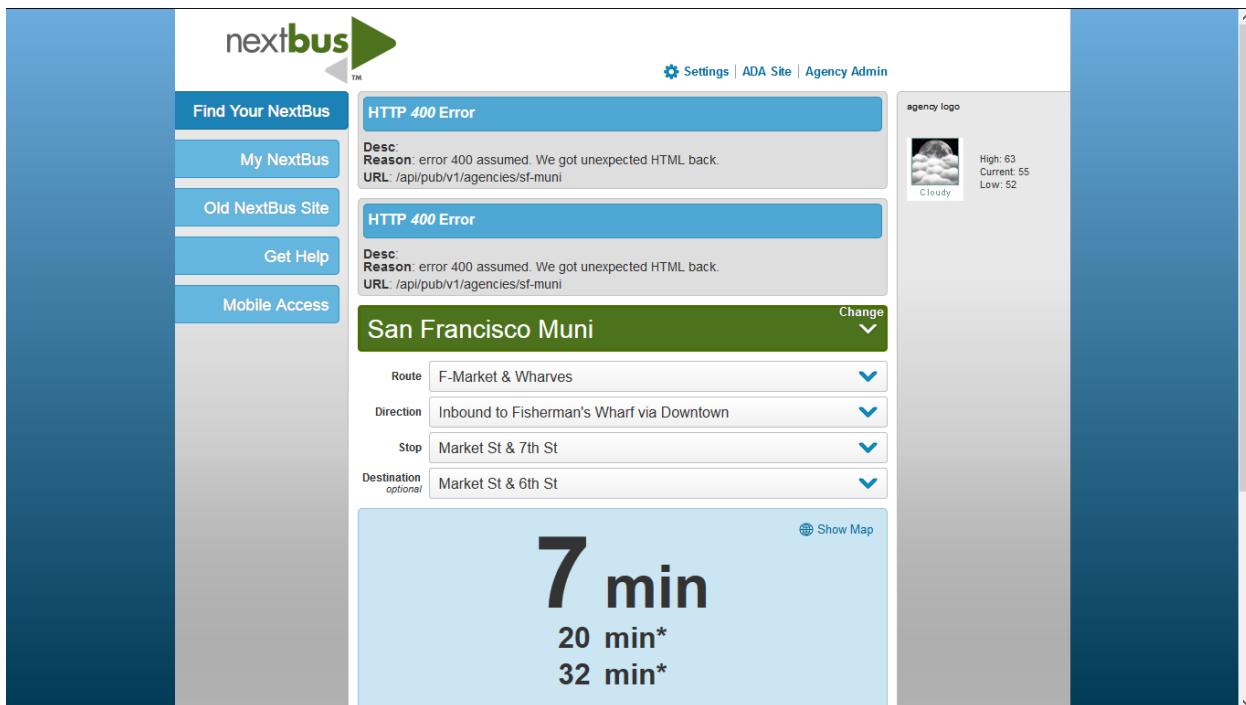


Figure: Screenshot of **nextbus.com** (NextBus, n.d.)

2.2.3. Moovit

Moovit is real time public transport vehicle tracking system with support of more than 500 cities around the world. The main feature of this system is that it provides vehicle with fastest and least crowded route. It provides various information such as schedules, maps, directions and step by step navigation. The search result is based on quickest route. Also, it provide features where user can get live reports and alerts such as transit change, favorite lines etc. from the other moovit users (Moovit, n.d.).

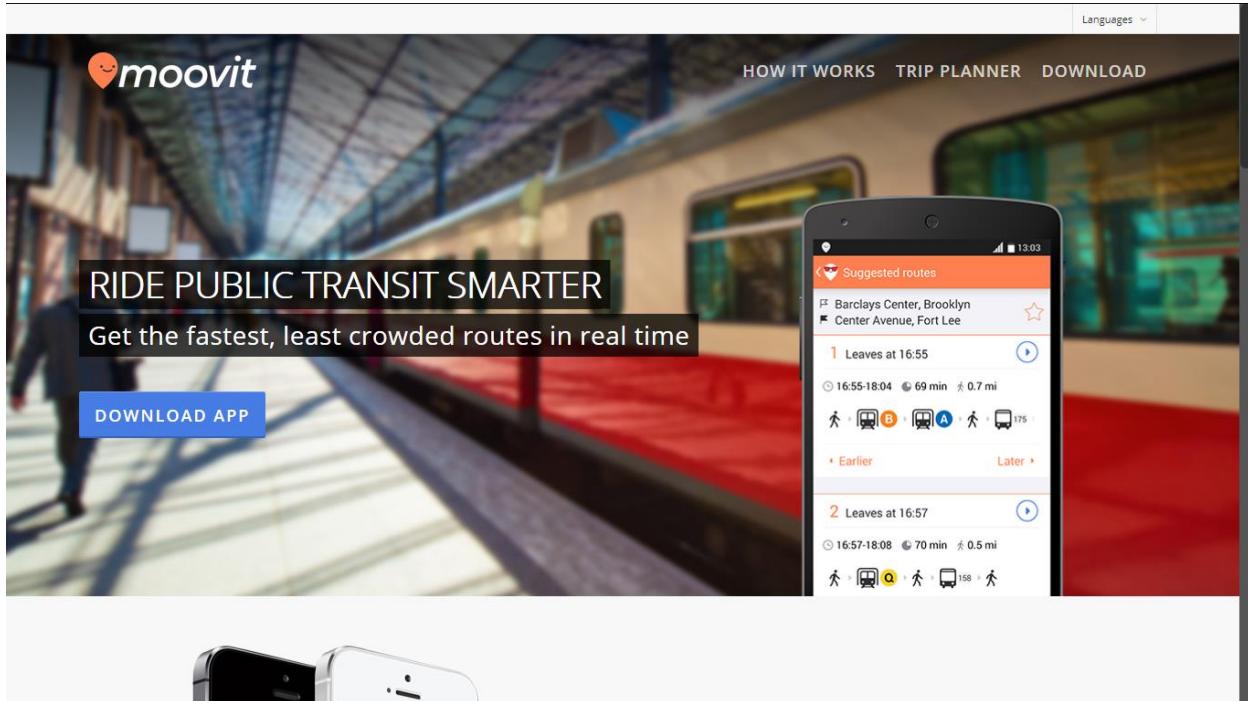


Figure: Screenshot of moovitapp.com (Moovit, n.d.)

2.2.4. Conclusion of Analysis of similar system

In the above systems, all of them provides location tracking and estimate arrival time service for the public buses. Although, the primary focus of all systems is to provide real time vehicle location service to the users, all these system provides various features along with vehicle tracking. Following is a comparison table of these system with the proposed system.

Features	CTA Bus Tracker	NextBus	Moovit	My System
Search for particular transport vehicles	Yes	Yes	Yes	Yes
Get estimate arrival time	Yes	Yes	Yes	Yes
Get fare amount information	No	No	No	Yes
View Bus location in Map	Yes	No	No	Yes
Online fare payment	No	No	No	Yes
Subscription to receive arrival and alerts by emails or text message	Yes	No	No	No
Job Access With Speech (JAWS) compatible website for disable people	No	Yes	No	No
Direction and Step by step navigation and map view	No	No	Yes	No
Get fastest and low crowded route suggestions	No	No	Yes	Yes
View live reports and alerts from another user	No	No	Yes	No
Get walking direction to bus stop	No	No	Yes	Yes
Search Traffic situation	No	No	No	Yes
Provide vehicle information in the Bus stop via digital display.	No	No	No	Yes
Share location during bus riding	No	No	No	yes

3. Development

3.1. Solution consideration

A software development methodology is a framework in which organization develop software product. This methodology contain series of activities and phases through the software is developed (Zeil, S., n.d.). Following are the list of methodologies considered for this project.

3.1.1. Big Bang Model

Big Bang Model is very simplistic methodology. This methodology includes little planning and scheduling. In fact, there are no specific process to follow. In this methodology, after initial project requirement, we can directly start the development phase and coding. In addition, there is no testing phase and it can be performed during the deployment of the product (Lameguy, 2013).

Advantages of Big Bang Model

- This is a very simple model and is easy to manage.
- This methodology do not have planning phase and planning is not required or is less.
- The resource required are few.
- This methodology is very flexible for the developer.
- It is good for practice for new developer and student for academic purpose.

3.1.2. Agile Model

Agile is one of the popular methodology used in software development. There are various types of Agile methodology and one of the them are Scrum and Extreme Programming. In agile, Test Driven Approach (TDD) is used in order to avoid over coding since using this technique we first write test first and then code it (Rastogi, V., 2015). In this method, the project will be divided into the small iteration called Sprint and system is delivered in each successful completion of sprint (CA Technologies, 2011). The Sprint development phase includes: Requirement analysis, Product Design, Coding and Testing. Agile is also called ‘Moving quickly’ and this method have an adaptive team which is very beneficial for changing requirement (Balaji, S., 2012).

Advantages of Agile Model

- Useful when client is not sure about the requirements.
- Product is developed in modules and therefore system is delivered in small time. Therefore, project is focused in delivering system in small part instead of delivering whole system.
- Risks are mitigated by continuous involvement of clients.
- Good project documentation
- Testing is integrated and performed throughout the project life cycle.

3.1.3. Enhanced Waterfall Methodology

Waterfall is one of the widely used software development methodology and Enhance waterfall is an advanced (modified) version of traditional waterfall methodology. This methodology is more flexible and phases can be reversible and overlapped when necessary (Munassar, N. & Govardhan, A., 2010). In Waterfall, development is proceed in sequence of phases from requirement to maintenance and each phase comprise a feedback loop where we can review previous phase work (Techrepublic.com, 2006). The Enhance Waterfall model include all features and in addition to these, we can jump to any phases in case of we required to go back to previous phases. In this model, project is divided into small parts or iteration which can be called as mini-waterfall process and which allow team to demonstrate result earlier and get feedback (Maheshwari, S. & Jain, D., 2012).

Advantages of Enhanced Waterfall Method

- Easy to identify milestones and progress since each phase is started after completing previous phase and easy to separate one stage from another.
- Documentation is excellent.
- Requirement is stable and definite.
- It is easy to identify and correct flaws in design phase than in testing stage.
- It is easy to go back to previous phases in case of any additional requirement or issues.
- Component reusability is excellent and strong project management

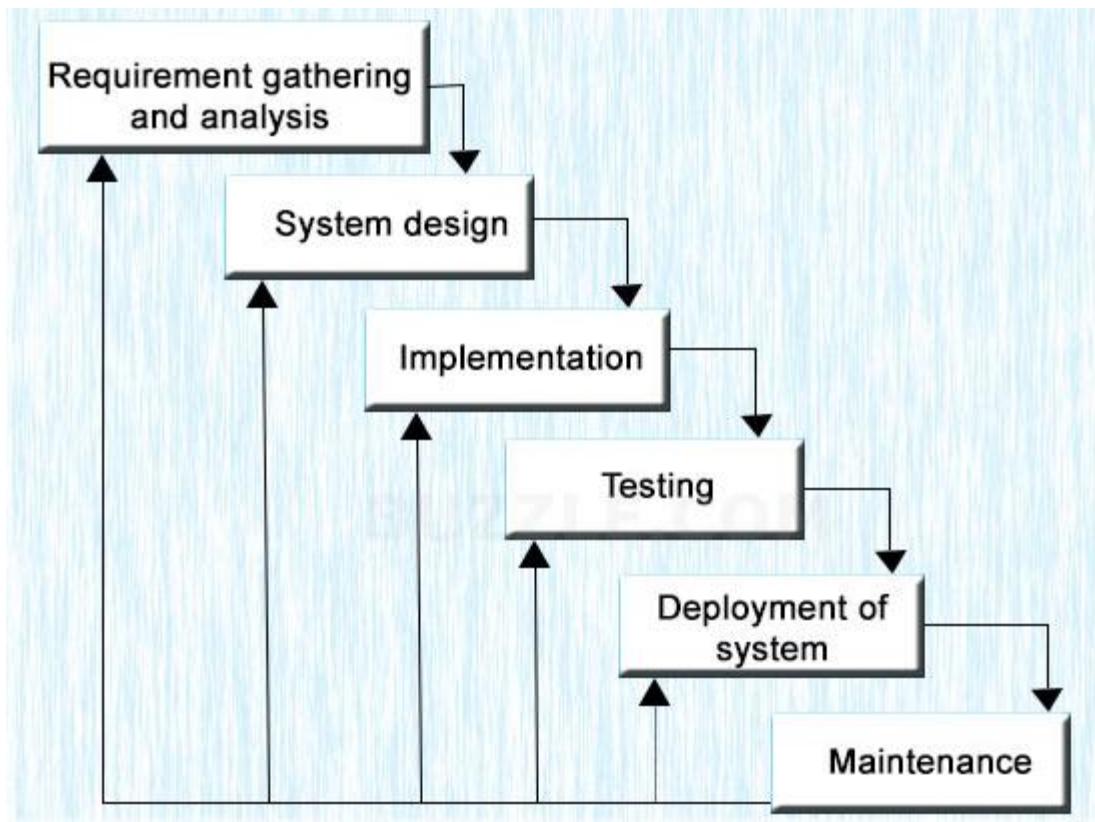
3.1.4. Chosen Methodology

All of the above methodologies are best option to choose and the project can be delivered successfully using any one of these methodology. However, considering certain factors such as quality of documentation, client involvement etc. **Enhance Waterfall Method** is chosen.

Using Big Bang, it is good for academic project and I can start developing the system as soon as possible and deliver the product in less time. However, there is no design and documentation part so this method is not suitable in term of documentation. In addition, using Agile method, the proposed system can be developed as modules and completed in less time. Also, this method is useful for single developer (Johnson, P., 2012). However, the requirement are well defined and according to the situation, meeting with the stakeholders on regular basis is not possible. Therefore, this methodology is not very useful in terms of effectiveness. Finally, the Enhance Waterfall methodology seems to be the most suitable one for my system. First reason is, the documentation is very strong and requirements are definite. In addition, milestones are easy to identify i.e. progress is measurable since development phases are separated. Unlike Agile, after requirement gathering, we are not required to interact with the stakeholder regularly. Also, the client require the complete system to deliver instead of portion of system in single time.

3.2. Phases of Enhanced Waterfall Methodology

In general, there are 6 phases in Enhanced waterfall methodology. Although, there are certain variation in standard phases of this methodology. For instance, according to Zeil S. (n.d.), the deployment (or installation) and maintenance phases are considered as single phase. Following are the common phases of Enhanced Waterfall Methodology.



Overview of Enhanced Waterfall Methodology (Parlekh, N., 2013)

3.2.1. Requirement Analysis Phase:

This phase is very important since correct requirement gathering is very important in order to develop right system. In this phase, all the possible requirement for the system will be identified. In addition, I will be studying client's business and problem domain and various challenges and risks of the system will be identified. Also, the scope of the project will be defined. I will be gathering requirements using various techniques such as interview, survey, and use cases etc. In addition, the project plan will be derived. The project plan will be represented with gantt chart. Also, the functional and non-functional requirement of system will be identified and System Requirement Specification document is developed.

3.2.2. Design Phase:

In this phase, database and various system component will be designed. The SRS document will be used as reference in order to design the system. The system will be designed into three segment: Architectural Design, High level Design and Low level Design. I will be using 3 tier-architecture design; presentation, Business and Data access layer for this system. Also, for high level design, I will use UML designing technique. In addition, the user interfaces will be designed using Pencil sketch tool. For the physical design of this system, I will be using Eclipse IDE, Android Studio. Also, I will be developing this system using Java programming language. In addition, for DBMS, oracle database will be used. For hardware, a smartphone to regularly update the GPS co-ordinate will be required (not to be confused with user's smartphone) and active internet connection.

3.2.3. Implementation Phase:

This is the coding phase where system is developed. The system will be developed module wise and each module will be developed integrated with unit testing. First, I will be developing the physical database on oracle platform. Then, I will develop core system i.e. web service and develop APIs. After this, I will develop an Android application and using the generated APIs and finally develop web system. In addition, for version controlling GIT will be used. After completing all

modules, they all will be integrated into the single system. Also, in each step of development, I will be performing unit testing.

3.2.4. Testing Phase:

This is an important phase. In this phase, I will be performing various functional and non-functional tests to the developed system in order to assure the system is working correctly without any issues and bugs. There are 3 major testing types: unit testing, integration testing and acceptance testing. The unit testing is performed in implementation phase. The integration testing is performed to integrated system and acceptance testing is performed in order to identify that correct functions and requirements are met. If issues found in the system, it will be solved.

3.2.5. Deployment Phase:

After successful testing of the system, the system will be finally delivered to client and installed to the client's environment and the client can start using the system. In addition, I will be developing various system manuals for the end users.

3.2.6. Maintenance Phase:

This is the phase after installing the developed system to the client's environment. No system are perfect and bugs free and there will be situations where client face various system related issues. In this phase, all those client's issues will be handled. Most of the time these issues will be solved in form of system patches and software updates. In addition, various modification will be made according to client's requirements.

3.3. Requirement Analysis Phase

3.3.1. SRS Document

3.3.1.1. Introduction

System Requirement Specification (SRS) provides an overall overview of the project purpose, scope, definition etc. The main aim of this document is to gather and have a good insight of this system by knowing the problem in details. The detail requirements for this system are provided in this document.

3.3.1.2. Purpose

The purpose of this document is to make to identify and list all the requirements of the client. In addition, this document provides high level overview of the system. Furthermore, this document defines the intended audience and project scope. Also, this report helps to identify what will be the user interface of the system.

3.3.1.3. Intended Audience

This document is intended for any individual user, team, developer, project manager etc. that needs to understand the system architecture and its specifications. The main uses for the reader are as follows:

Developer: The developer who wants to read, change, modify or add new requirements into the existing program can consult this document.

User: The end user, sajha yatayat staffs.

3.3.1.4. Project Scope

The scope this project is as follows:

- Manage various route information including busstops and route ways.
- Manage various fare, vehicle and schedule information including add, edit and delete.
- Track all the available vehicles and monitor them. In addition, flag and unflag vehicles as required.
- Generate various reports, such as traffic, passenger's details etc.
- An android application for end user, from which user can find vehicle around them and track them, get estimated time of arrival, make fare payment.

3.3.1.5. Overall Description

This section will give an overview of the complete system.

3.3.1.5.1. Product Perspective

The system has 3 sub system. Web system – from which the sajha yatayat staff can manage and track various vehicle, and generate various reports. In addition, the system contain an android application from which end user – passenger can find and track vehicle around them. In addition, the system contain a driver application from which each vehicle location is updated regularly to the main system.

3.3.1.5.2. User Classes and Characteristics

Basic knowledge of using the computer is recommended to use this system and android smartphone is required.

3.3.1.5.3. Operating Environment

The web system will operate any of the following operating environment which have browser enabled. In addition, the passenger and driver app will only operate on android OS.

- Microsoft Windows 7/8/10
- Ubuntu and (other Linux OS)
- Apple Mac OS
- Android smartphone

3.3.1.5.4. Design and Implementation Constraints

The system is carried out using Hypertext Preprocessor (PHP) and uses MySQL for database server. So minimum PC having at least 1GB RAM and CPU over 2.4 MHz is required to run the system at a good speed. The internet speed of minimum of 256 Kbps is required for fast update of the live score to the server.

For language support only English language is available. For the internet connection any internet can be connected to run the system in multiple desktop or laptops.

3.3.1.5.5. Assumptions and Dependencies

The system does not have any dependencies.

3.3.1.5.6. Fact Finding Techniques

Interview: conducted with the Sajha yatayats' staff. The main purpose of this interview is to identify what is the current situation of vehicle information management and what are their current problems.

Survey: A survey is conducted in order to identify current problems of passenger in regards to vehicle and time and to find what their preference to avoid their current problem are and which can be implemented on this system.

3.3.1.5.7. Functional Requirement

1. Find/Search Bus

Description: A passenger need to get Bus for his/her particular destination and then he/she will search for particular Bus near to his/her current location. First, user provide preferred destination to go with appropriate search option selected and then system provide appropriate proximity Bus lists along with Estimate Arrival time to the user nearest next stop and other information.

S.N.	Requirements	Priority
1.	Passenger shall be able to find particular vehicle near their destination.	Must have
2.	Passenger shall be able to view details of vehicle including ETA, bus stop etc.	Must have

2. Track Bus

Description: A passenger get the list of Bus for his/her required destination and then passenger select their preferred vehicle from the list and can track that particular Bus by adding it to pinned list by executing ‘Pin Bus’ use case. In addition, user can view the current location of the bus in map by executing ‘View Bus in Map’ use case.

S.N.	Requirements	Priority
1.	Passenger shall be able to book mark a vehicle and then start tracking the location of that vehicle.	Must have
2.	Passenger shall be able to view vehicle location on map.	Must have
3	Passenger shall able to view walking route from user location to the bus stop.	Must have

3. Payment

Description: Passenger is in the Bus and wish to make online fare payment. The passenger first select the fare amount to pay then provides the online payment gateway credentials, confirms the payment.

S.N.	Requirements	Priority
1.	Passenger shall be able to make fare payment.	Must have

4. View Route

Description: Passenger wish to view all the available routes. He/she select to view all the available routes and gets the list of all available routes. In addition, passenger further select to view particular route in interactive map mode.

S.N.	Requirements	Priority
1.	Passenger shall be able to view all the available routes.	Must have
2.	Passenger shall be able to view detail of selected route on map.	Must have

5. View Traffic

Description: Passenger wish to know the current situation of traffic in various places in available routes. He/she first provides particular place and then receives the traffic situation information for that place.

S.N.	Requirements	Priority
1.	Passenger shall be able to view all the available routes.	Must have
2.	Passenger shall be able to view traffic affected roads of selected route on map.	Must have

6. Authenticate User

Description: A Passenger login to the system by providing login credentials. Access will be granted if the login credentials are correct otherwise access to the system will be denied.

S.N.	Requirements	Priority
1.	Passenger shall be able to login to the system.	Must have
2.	Passenger shall be able to reset their password.	Must have
	Passenger shall be able to logout of the system.	Must have

7. Manage Profile

Description: Passenger wish to create a new account. First, passenger complete registration form and then passenger submit the filled form.

S.N.	Requirements	Priority
1.	Passenger shall be able to create new account.	Must have
2.	Passenger shall be able to view and update their account details.	Must have

8. Query and Feedback

Description: Passenger wish to contact with the Sajha Yatayat for various issues or questions. He/she first fill the query/feedback form by providing all the required information including his/her issue or question and finally submit the query/feedback.

S.N.	Requirements	Priority
1.	Passenger shall be able to post their query	Must have

9. Manage Schedule

Description: A staff add schedule information for various Bus and for different route. In addition, a staff make changes to schedule details and update the changed schedule. Similarly, staff delete existing schedule information of particular or all bus.

S.N.	Requirements	Priority
1.	Passenger shall be able to view schedules	Must have
2.	Passenger shall be able to add new schedules	Must have
	Passenger shall be able to update schedules	Must have
	Passenger shall be able to delete schedules	Must have

10. Manage Fare

Description: A staff add fare information for various distance and route. In addition, staff make changes to fare amounts and update the new fare amount for various distance. Similarly, staff delete existing fare amounts detail from the system for particular distance or all distance.

S.N.	Requirements	Priority
1.	Passenger shall be able to view fares.	Must have
2.	Passenger shall be able to add new fare.	Must have
	Passenger shall be able to update fares.	Must have
	Passenger shall be able to delete fare.	Must have

11. Manage Vehicle Information

Description: A staff add vehicle information to the system and that vehicle will be available to track by the passenger and staffs. In addition, staff update or delete the required vehicle information from the existing list of vehicles.

S.N.	Requirements	Priority
1.	Passenger shall be able to view vehicle detail.	Must have
2.	Passenger shall be able to add new vehicle detail.	Must have

	Passenger shall be able to update vehicle detail.	Must have
	Passenger shall be able to delete vehicle detail.	Must have

12. Manage Route Information

Description: A staff add new route information including bus stops, distance between bus stops, longitude/latitude of bus stops etc. Similarly, staff update or delete any existing route and its related data.

S.N.	Requirements	Priority
1.	Passenger shall be able to view route, bus stop and route way data.	Must have
2.	Passenger shall be able to add new route, bus stop, and route way information.	Must have
	Passenger shall be able to update route, bus stop, and route way information.	Must have
	Passenger shall be able to delete route, bus stop, and route way information.	Must have

13. Track Bus

Description: A staff track the location of all the Bus for all routes. He/she first choose the particular route and get all the Bus location for that route in an interactive map mode. In addition, staff flag a particular vehicle by executing ‘Flag Bus’ use case.

S.N.	Requirements	Priority
1.	Passenger shall be able find all the vehicle for particular route and track/view their location on map.	Must have
2.	Passenger shall be able view vehicle location detail on table format.	Must have
	Passenger shall be able to flag vehicle.	Must have
	Passenger shall be able to unflag vehicle	Must have

14. Authenticate Staff

Description: A Staff login to the system by providing login credentials. Access will be granted if the login credentials are correct otherwise access to the system will be denied.

S.N.	Requirements	Priority
1.	Passenger shall be able to login to the system.	Must have
2.	Passenger shall be able to logout from the system.	Must have

15. Generate Reports

Description: A staff generate various reports based on the requirements.

S.N.	Requirements	Priority
1.	Passenger shall be able to generate various reports.	Must have

16. Update Vehicle Location

Description: Driver update the vehicle location when their route started.

S.N.	Requirements	Priority
1.	Driver shall be able to view their route and vehicle name and then update location.	Must have

3.3.1.5.8. Non-Functional Requirements

- **Usability**

Description: The user should be able to add, edit, remove, and make attendance of a staff within a 15 sec.

Rational: It should be faster than a manual paper based attendance

Dependency: R2, R3, R4, R5

Source: User of the system.

- **Consistency**

Description: The background color, font-family & size etc. are consistent on each page.

Rational: Inconsistency may affect user preferences and may confuse them.

Dependency: Not dependent

Source: Client web portal user, Android app user

- **Reliability**

Description: The system must be reliable such as user must not get wrong or inaccurate information from this system. For instance, if user get wrong location information from this system then this application will not useful.

Rational: System should be reliable and provide accurate information to the system.

Dependency: Not dependent.

Source: Client web portal user, Android app user

- **Privacy**

Description: The user provided information such as password must be encrypted and must not be compromised.

Rational: The user password is one of the major concern and encrypting password will protect user's data.

Dependency: Not dependent.

Source: Client web portal user, Android app user

- **Response time:**

Description: The response time of system must be minimum. When user request for information then information should be provided in less time. If the system takes long time to display information then also user will get undesired experience and avoid to use the system. Also, this system regularly update location information and time and slow response will provide wrong information to the user.

Rational: The user must be provided data in less time so that user can comfortably use this system services.

Dependency: Not dependent.

Source: Client web portal user, Android app user

- **Stability:**

Description: The system must be stable and must not crash unexpectedly.

Rational: Unnecessary system crash will make user to avoid using this system.

Dependency: Not dependent.

Source: Client web portal user, Android app user

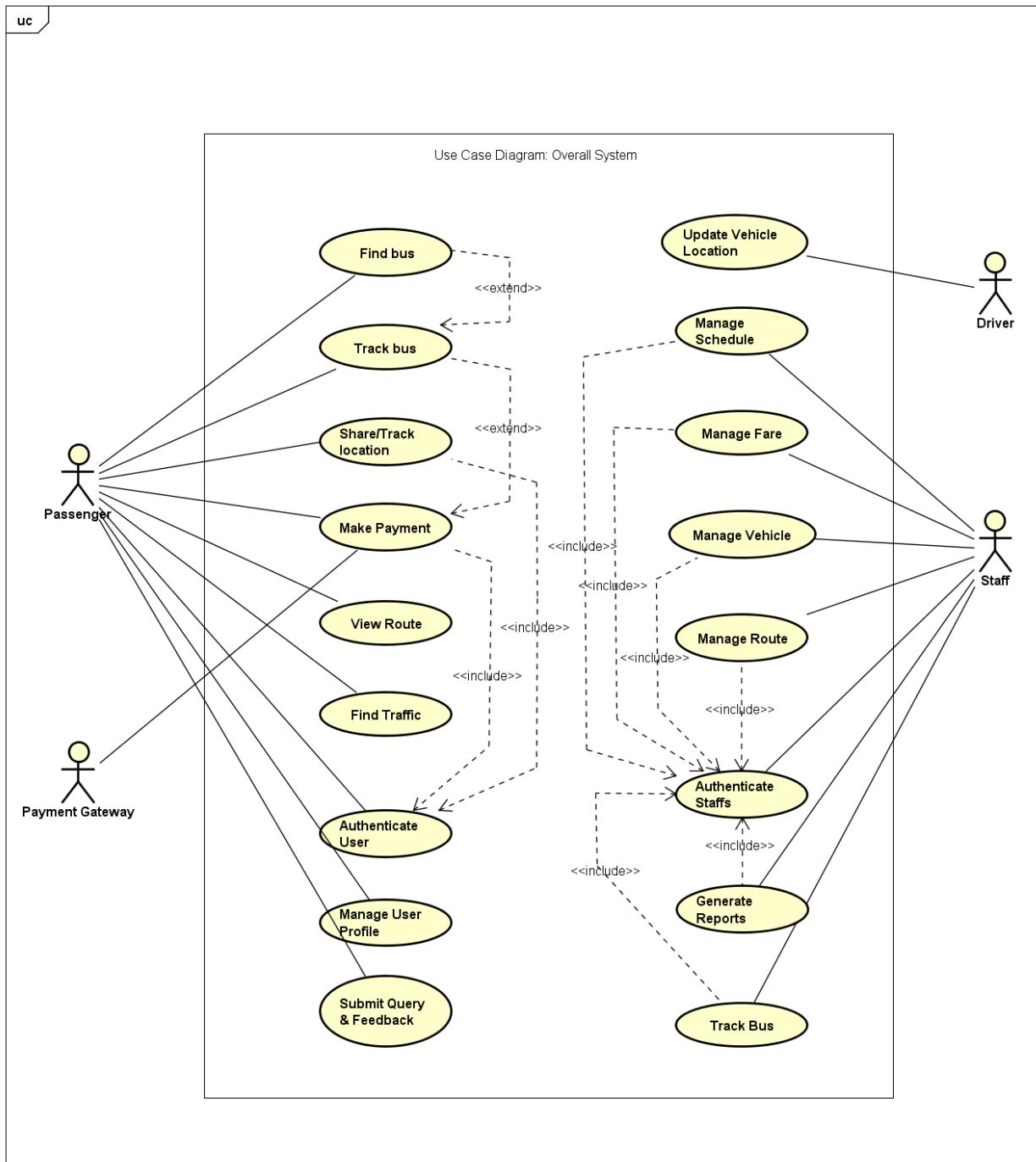


Figure: Overview of use case diagram

3.4. System Design Phase

3.4.1. User Case Design

3.4.1.1. Find/Search Bus

Actors: Passenger (Initiator)

Purpose: Find a nearest Bus around passenger's current location.

Overview:

A passenger need to get Bus for his/her particular destination and then he/she will search for particular Bus near to his/her current location. First, user provide preferred destination to go with appropriate search option selected and get Bus lists result along with Estimate Arrival time to the user nearest next stop and other information.

Type: Primary (Initiator)

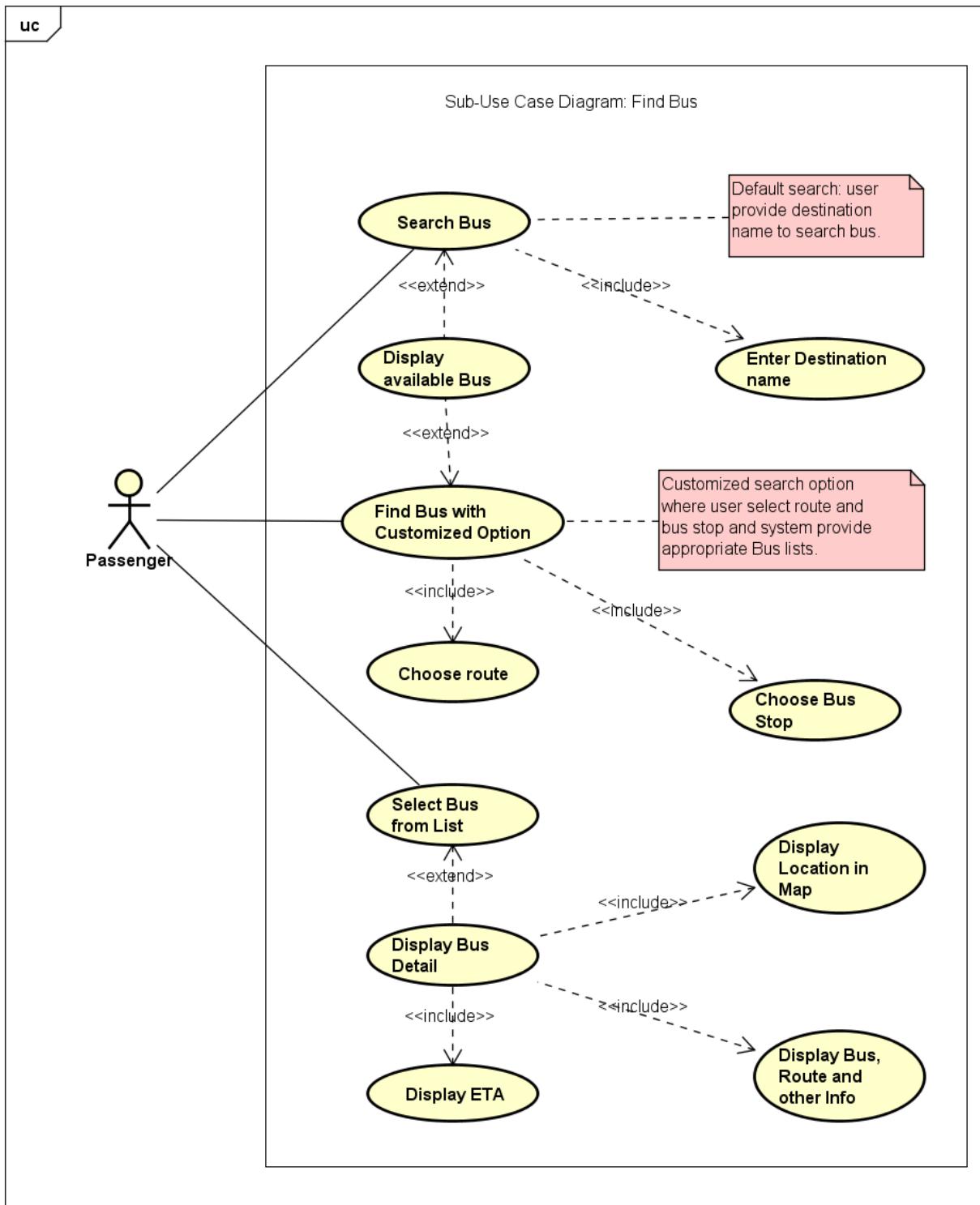
Dialog:

Actor	System
1. Passenger select the required search option and search for the all the available Bus.	2. Displays list of available Bus near to passenger with additional information such as next stop, ETA to next stop, Bus number etc.

Alternative Course of action:

Line 2:

If system could not find any Bus around passenger's current location then the system will display no-result found message.



3.4.1.2. Track Bus

Actors: Passenger (Initiator)

Purpose: Passenger selects the required bus for his/her destination and start tracking.

Overview:

A passenger get the list of Bus for his/her required destination and then passenger select their preferred vehicle from the list and can track that particular Bus by adding it to pinned list by executing ‘Pin Bus’ use case. In addition, user can view the current location of the bus in map by executing ‘View Bus in Map’ use case.

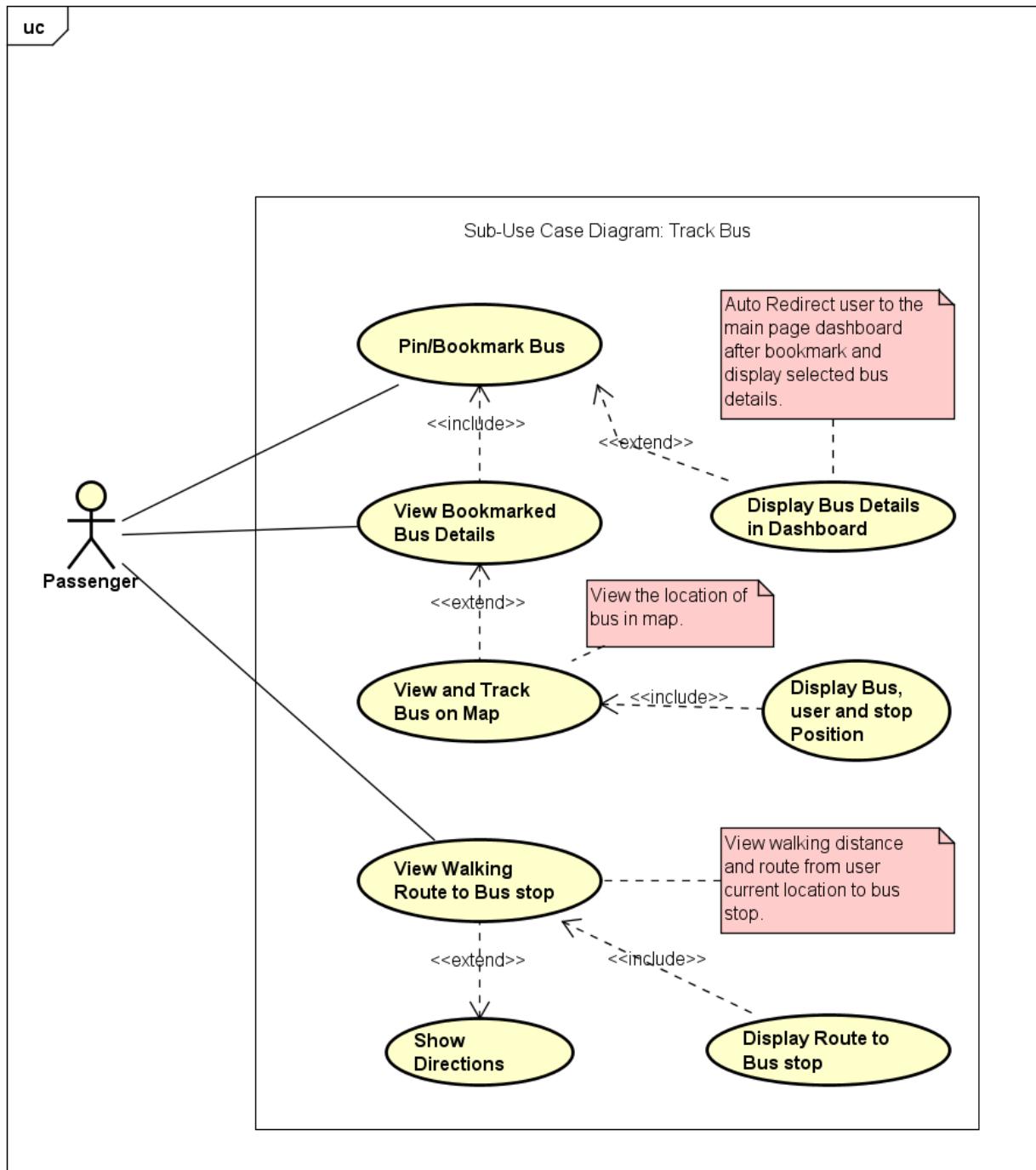
Type: Primary (Initiator)

Dialog:

Actor	System
<p>1. Passenger select a particular Bus from the result list of Bus and pin it for tracking.</p> <p>3. Passenger selects to view Bus location in the map.</p> <p>5. Passenger select to get a walking route to the bus stop.</p>	<p>2. System add the selected Bus details to the main dashboard and continuously update the Bus information.</p> <p>4. Display the Bus location in the map.</p> <p>6. Calculates shortest distance way to the bus stop from user location.</p> <p>7. Display the route to the bus stop.</p>

Alternative Course of action:

Line 6: If the system is unable to find/calculate the shortest distance/route to the bus stop then user will be provided an error message.



3.4.1.3. Payment

Actors: Passenger (Initiator)

Purpose: Passenger make a fare payment.

Overview:

Passenger is in the Bus and wish to make online fare payment. The passenger first select the fare amount to pay then provides the online payment gateway credentials, confirms the payment.

Type: Primary (Initiator)

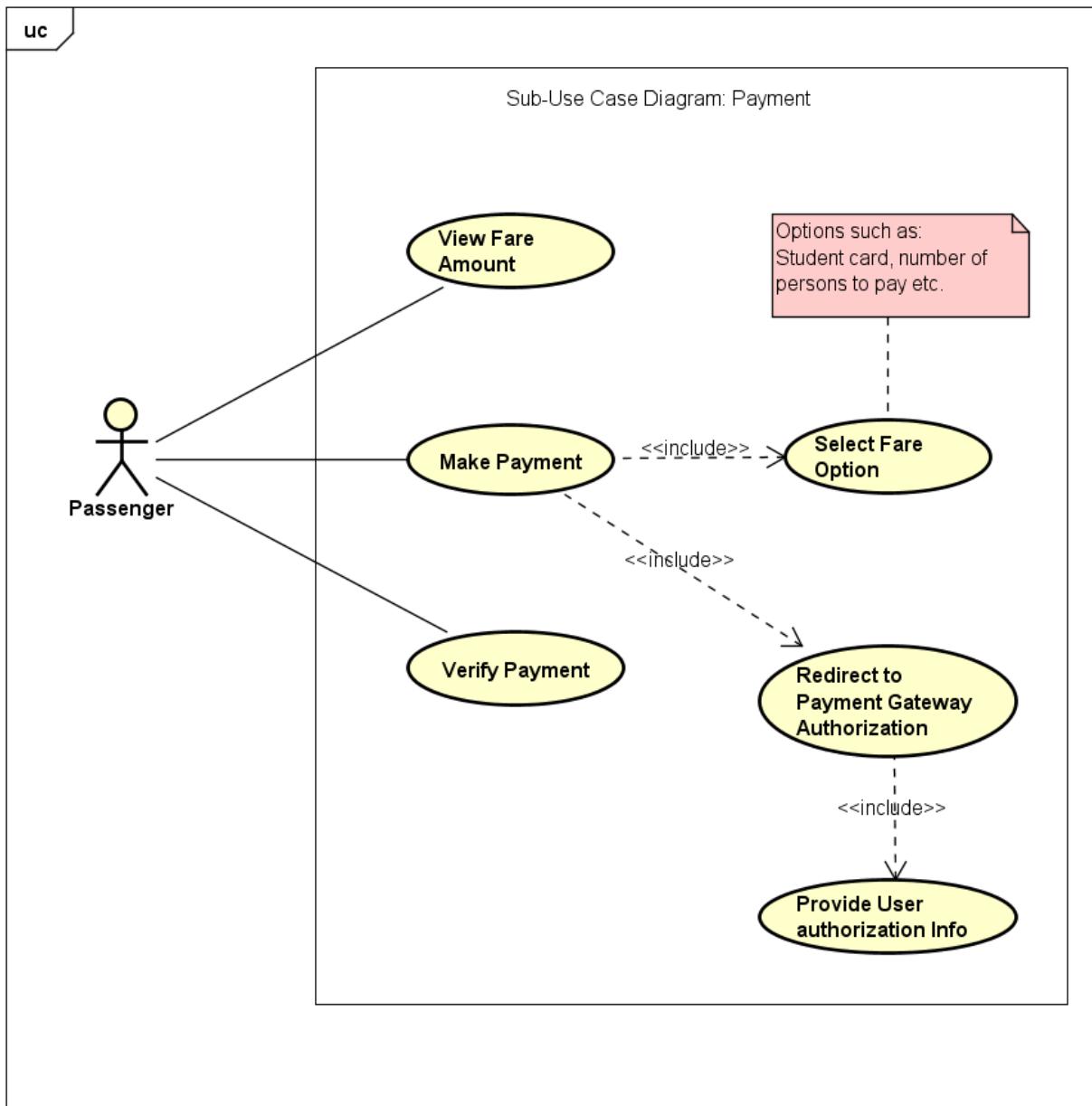
Dialog:

Actor	System
1. Passenger login to the system using the use case 'Login'. 4. Passenger fill the payment criteria and make a payment. 6. Passenger enters payment gateway login credential and confirm the payment.	2. Checks if the provided credentials are correct or not. 3. Provide access to the system if credential are correct else denied access to system and display error message. 5. Provide redirection to the payment gateway login page. 7. Request and checks the payment status from the payment gateway. 8. Verify the payment and send verified message to passenger.

Alternative Course of action:

Line 5: If the payment gateway login page is not redirected or displayed. Display passenger error message.

Line 8: If the payment verification failed user is provided verification failed message.



3.4.1.4. View Route

Actors: Passenger (Initiator)

Purpose: Passenger view the all the available route of sajha yatayat.

Overview:

Passenger wish to view all the available routes. He/she select to view all the available routes and gets the list of all available routes. In addition, passenger further select to view particular route in interactive map mode.

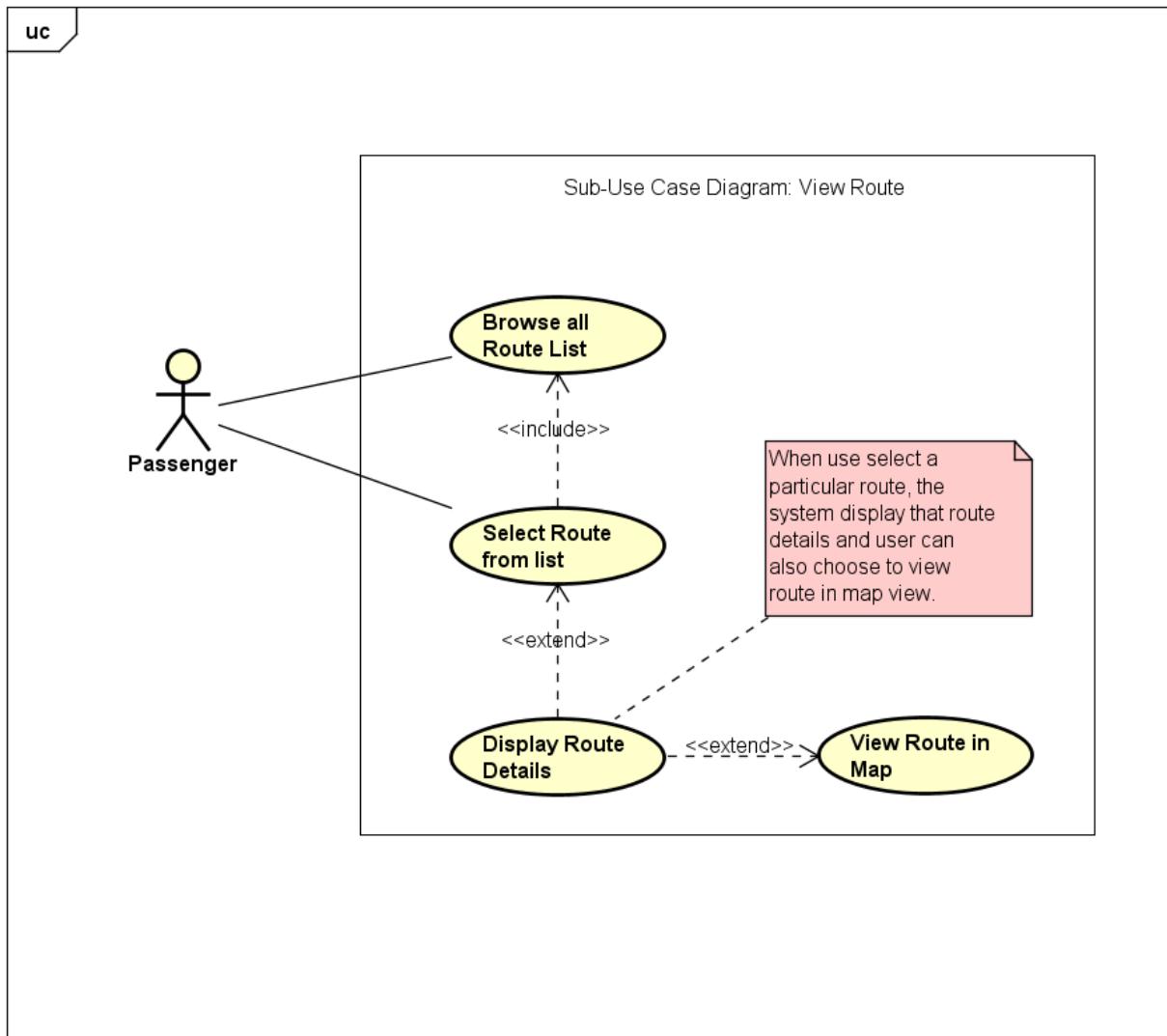
Type: Primary (Initiator)

Dialog:

Actor	System
1. Passenger selects to view all the available routes. 4. Passenger select particular route from the list.	2. Gets the available routes. 3. Displays the available routes. 5. Fetch the route location data from the data store. 6. Display the route in interactive map mode.

Alternative Course of action:

N/A



3.4.1.5. View Traffic

Actors: Passenger (Initiator)

Purpose: Passenger view the traffic situation on particular place.

Overview:

Passenger wish to know the current situation of traffic in various places in available routes. He/she first provides particular place and then receives the traffic situation information for that place.

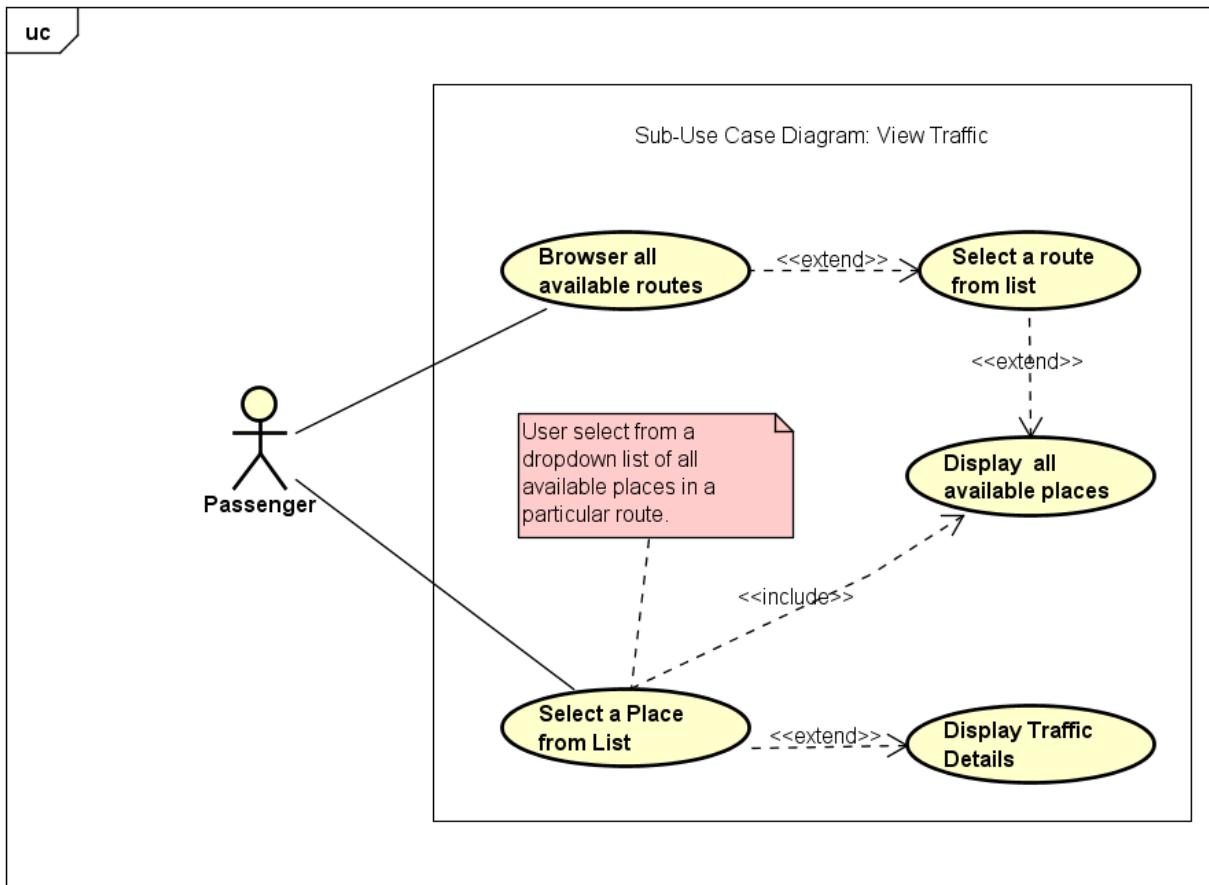
Type: Primary (Initiator)

Dialog:

Actor	System
1. Passenger selects to view the traffic finder page. 3. Passenger select any particular route from the list and view traffic detail for that route.	2. Get all the available place name from the data store and display to passenger. 4. Fetch the traffic data of passenger selected route. 5. Display traffic information.

Alternative Course of action:

Line 4: If no traffic data found, display error message to the passenger.



3.4.1.6. Authenticate User

Actors: Passenger (Initiator)

Purpose: User login to the system in order to access to the system.

Overview:

A Passenger login to the system by providing login credentials. Access will be granted if the login credentials are correct otherwise access to the system will be denied.

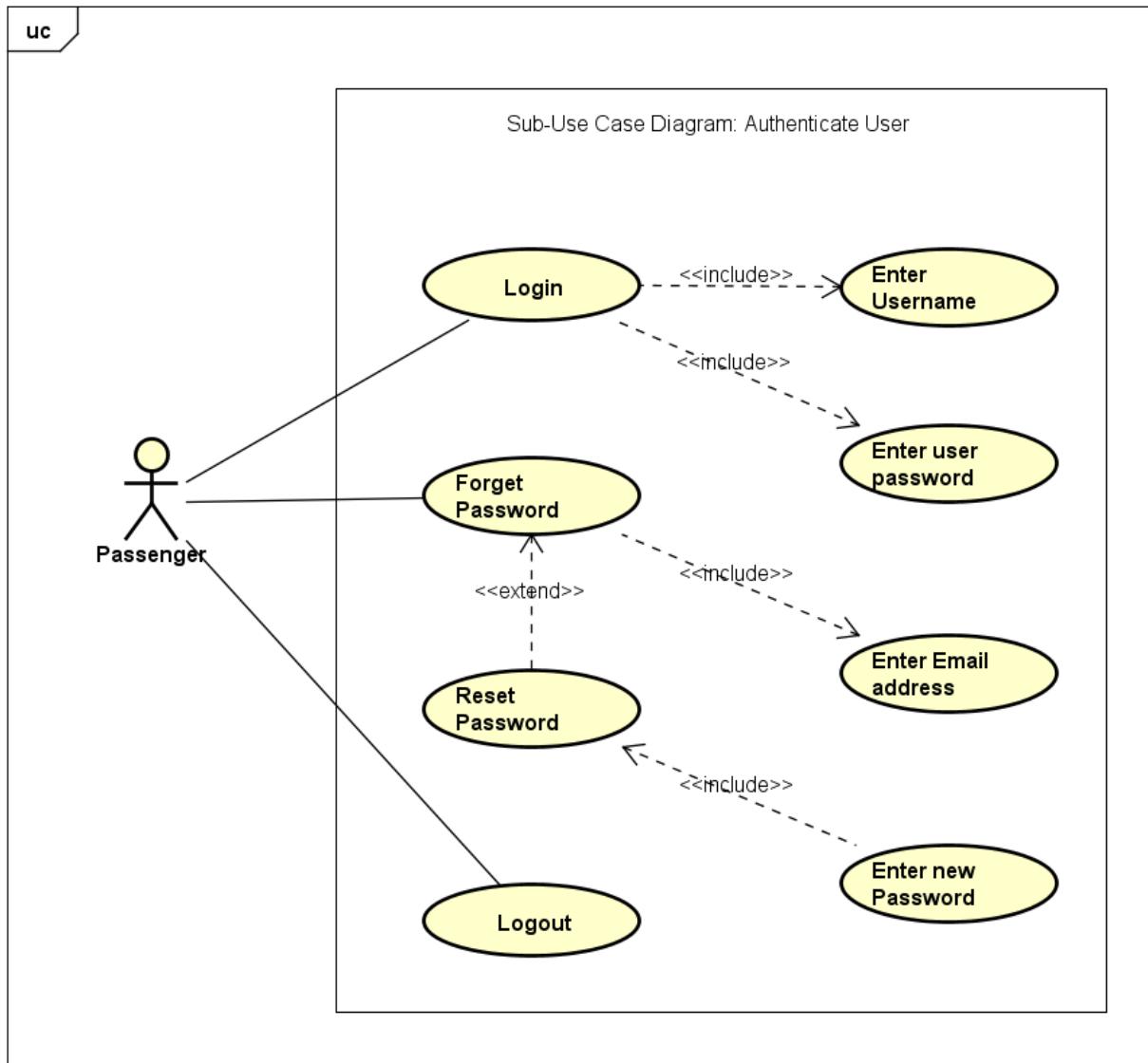
Type: Primary (Initiator)

Dialog:

Actor	System
1. Passenger provide usename and password 4. Passenger forget the password and click reset password button. 6. Passenger enter the email and submit. 9. Passenger enter the code and submit. 12. Passenger provides new password and submit. 14. User click logout button.	2. Validate the username and password the passenger. 3. If authentication success, grant passenger access to the system and if authentication failed, deny the access to the system. 5. System provides email input page. 7. System check the emails and if correct then system sends the reset code to passenger's email address. 8. Redirect to code enter page. 10. validates the code and if correct ask user to enter new password. 13. System change the new password. 15. Logout the user.

Alternative Course of action:

Line 3: If authentication failed, display invalid username/password error message.



3.4.1.7. Manage Profile

Actors: Passenger (Initiator)

Purpose: Passenger creates a new account.

Overview:

Passenger wish to create a new account. First, passenger complete registration form and then passenger submit the filled form.

Type: Primary (Initiator)

Dialog:

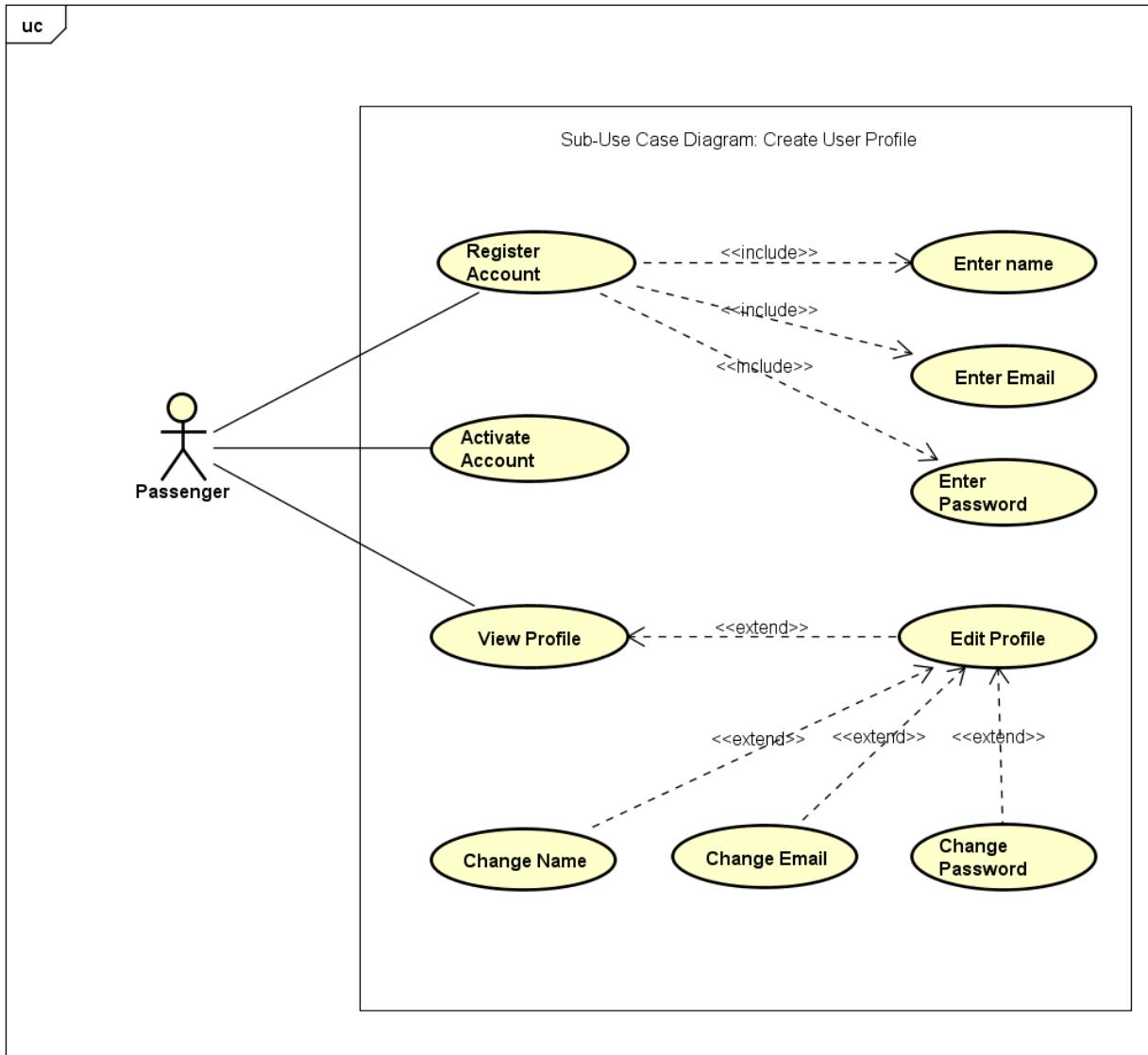
Actor	System
1. Passenger fill the registration form. 5. Get the verification code and submit the verification code. 7. Passenger view profile	2. Validates the form information and submit the registration form 3. Save the data to the data store. 4. Send verification code to user email address and display verify page to passenger. 6. Validates the verification code and redirect to home page. 8. System display user profile.

Alternative Course of action:

Line 2: If validation failed, then error message will be displayed to the passenger.

Line 5: If verification code is not received, request to get new verification code.

Line 6: If verification code validation failed, display error message.



3.4.1.8. Query and Feedback

Actors: Passenger (Initiator)

Purpose: Passenger submit query/feedback.

Overview:

Passenger wish to contact with the Sajha Yatayat for various issues or questions. He/she first fill the query/feedback form by providing all the required information including his/her issue or question and finally submit the query/feedback.

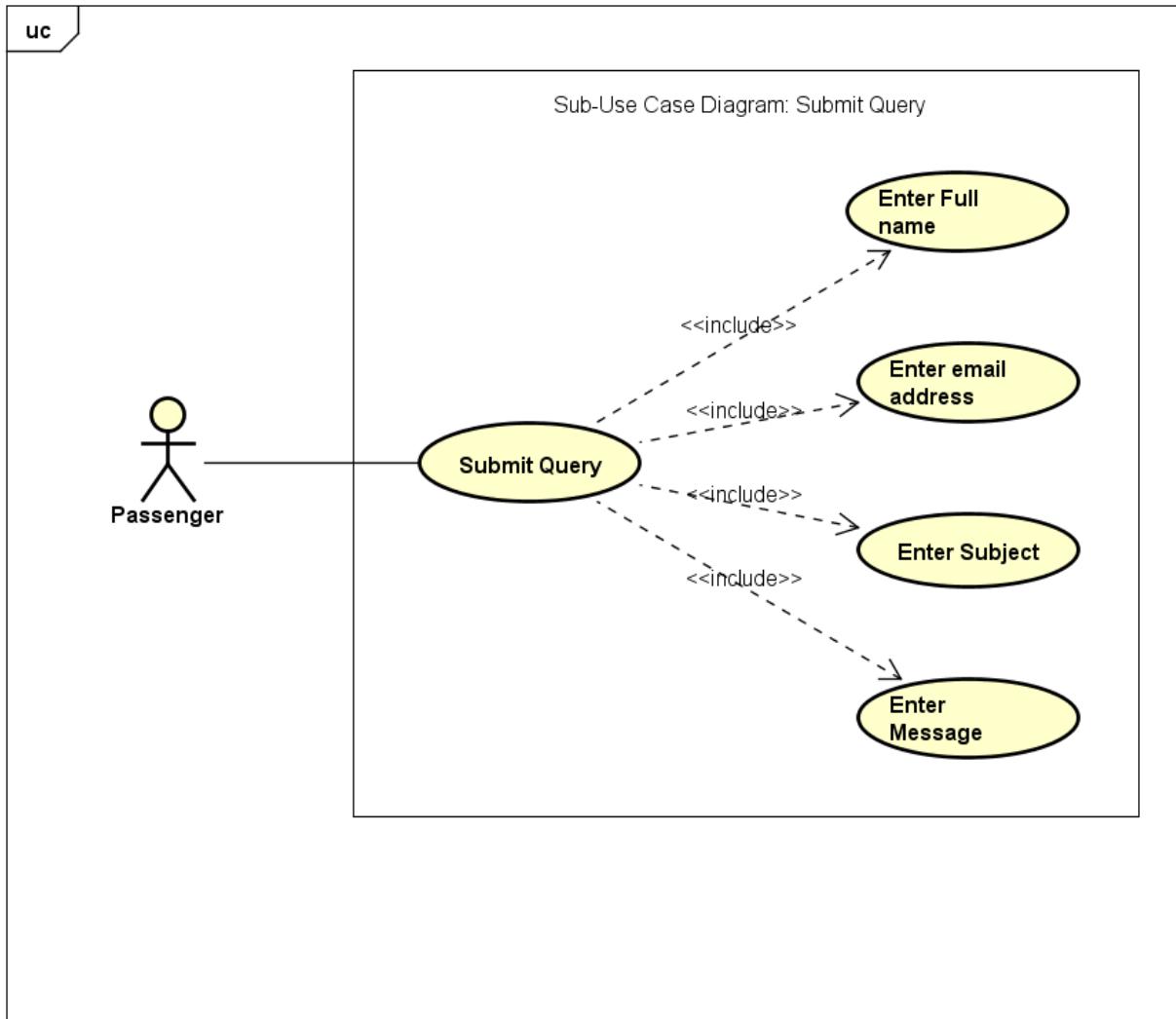
Type: Primary (Initiator)

Dialog:

Actor	System
1. Passenger select to submit a query and fill all the information and submit the query.	2. Validates the query form and send the query information to the admin.

Alternative Course of action:

Line 2: If validation failed, display error message and ask user to check submission form and submit again.



3.4.1.9. Manage Schedule

Actors: Staff (Initiator)

Purpose: Staff add, update and delete the scheduling information of Bus.

Overview:

A staff add schedule information for various Bus and for different route. In addition, a staff make changes to schedule details and update the changed schedule. Similarly, staff delete existing schedule information of particular or all bus.

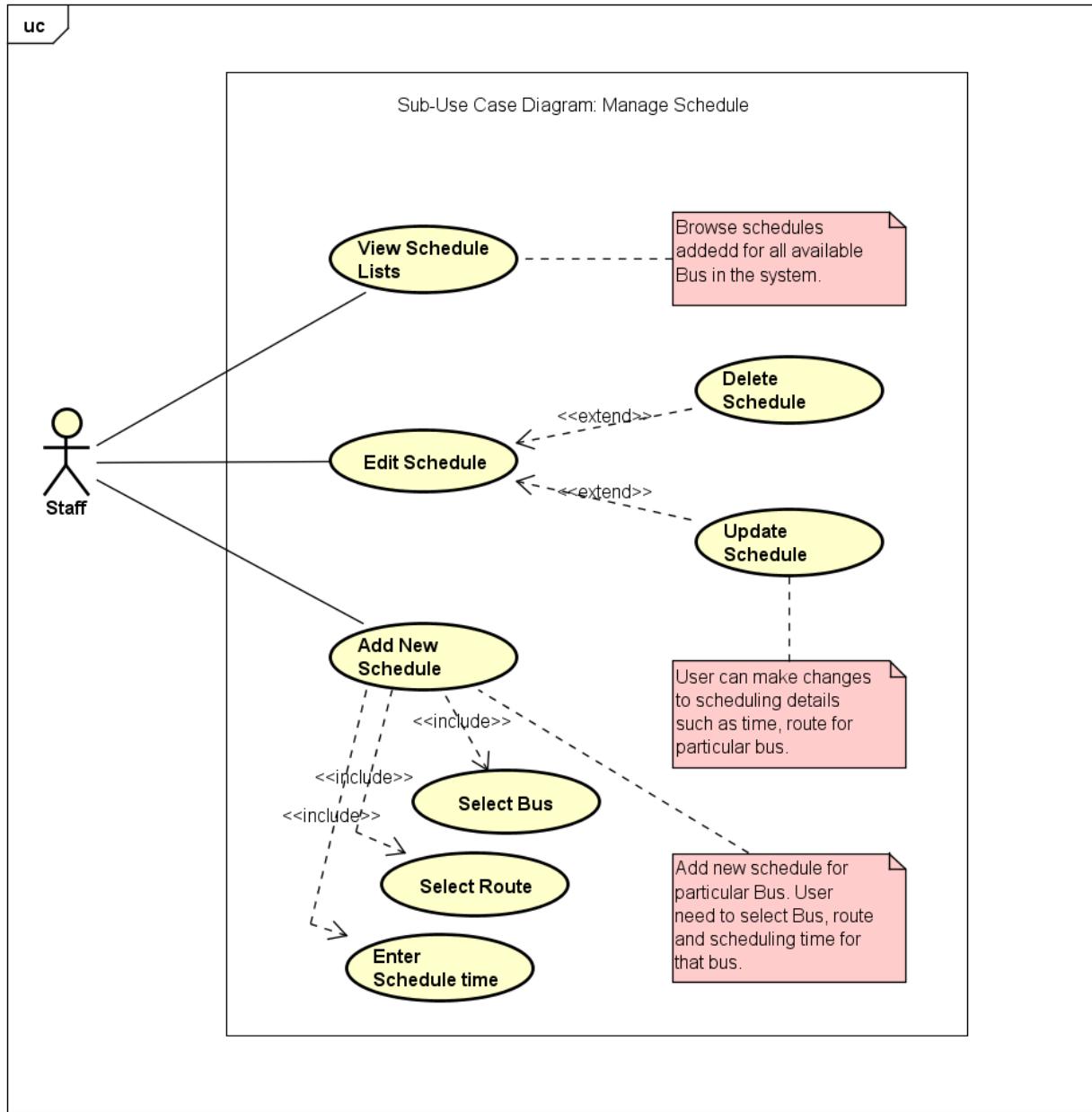
Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff add new schedule information. 3. Staff update the existing schedule information. 5. Delete particular or all schedule information.	2. Add the new schedule information in the data store. 4. Update the current schedule details to the changed values. 6.

Alternative Course of action:

Line 2, 4: Display error message if action is not complete.



3.4.1.10. Manage Fare

Actors: Staff (Initiator)

Purpose:

Overview:

A staff add fare information for various distance and route. In addition, staff make changes to fare amounts and update the new fare amount for various distance. Similarly, staff delete existing fare amounts detail from the system for particular distance or all distance.

Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff selects to view all fare details. 3. Staff enter new fare details and submit. 5. Staff make changes to the current value of fare and submit. 7. Staff selects particular fare details and click delete button.	2. Display fare list. 4. Validates the new fare details and record to the system. 6. Update the fare details. 8. Delete the selected fare details from the database.

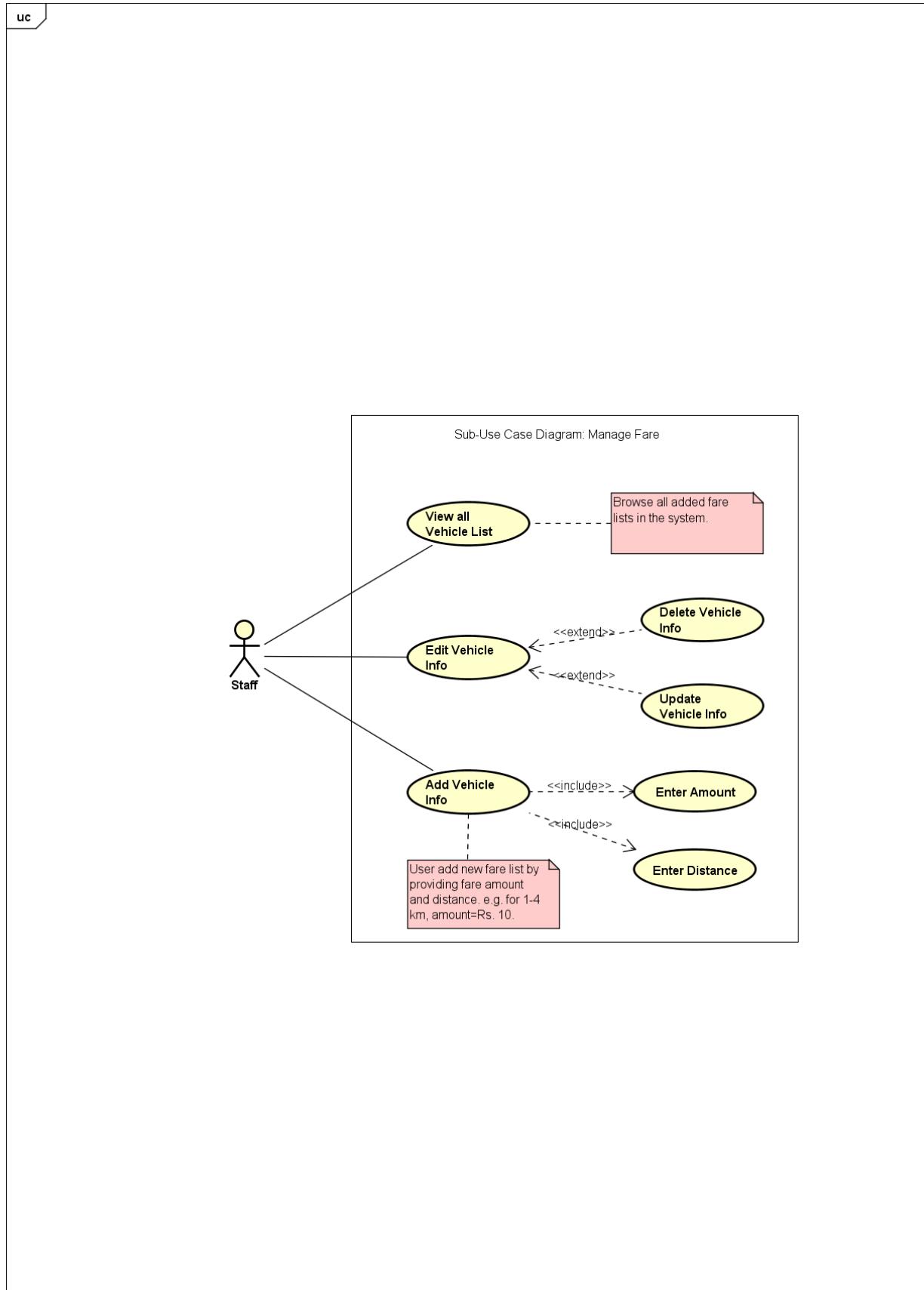
Alternative Course of action:

Line 2: If no fare details are found then system will display no details found message.

Line 4: If the entered details is invalid then system will throw error message.

Line 6: If system cannot make changes to the current selected fare details error message to try later will be displayed.

Line 8: If a fare record cannot be deleted then system will display error message to try later.



3.4.1.11. Manage Vehicle Information

Actors: Staff (Initiator)

Purpose:

Overview:

A staff add vehicle information to the system and that vehicle will be available to track by the passenger and staffs. In addition, staff update or delete the required vehicle information from the existing list of vehicles.

Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff selects to view all Vehicle details. 3. Staff enter new Vehicle details and submit. 5. Staff make changes to the current value of Vehicle and submit. 7. Staff selects particular Vehicle details and click delete button.	2. Display Vehicle list. 4. Validates the new Vehicle details and record to the system. 6. Update the Vehicle details. 8. Delete the selected Vehicle details from the database.

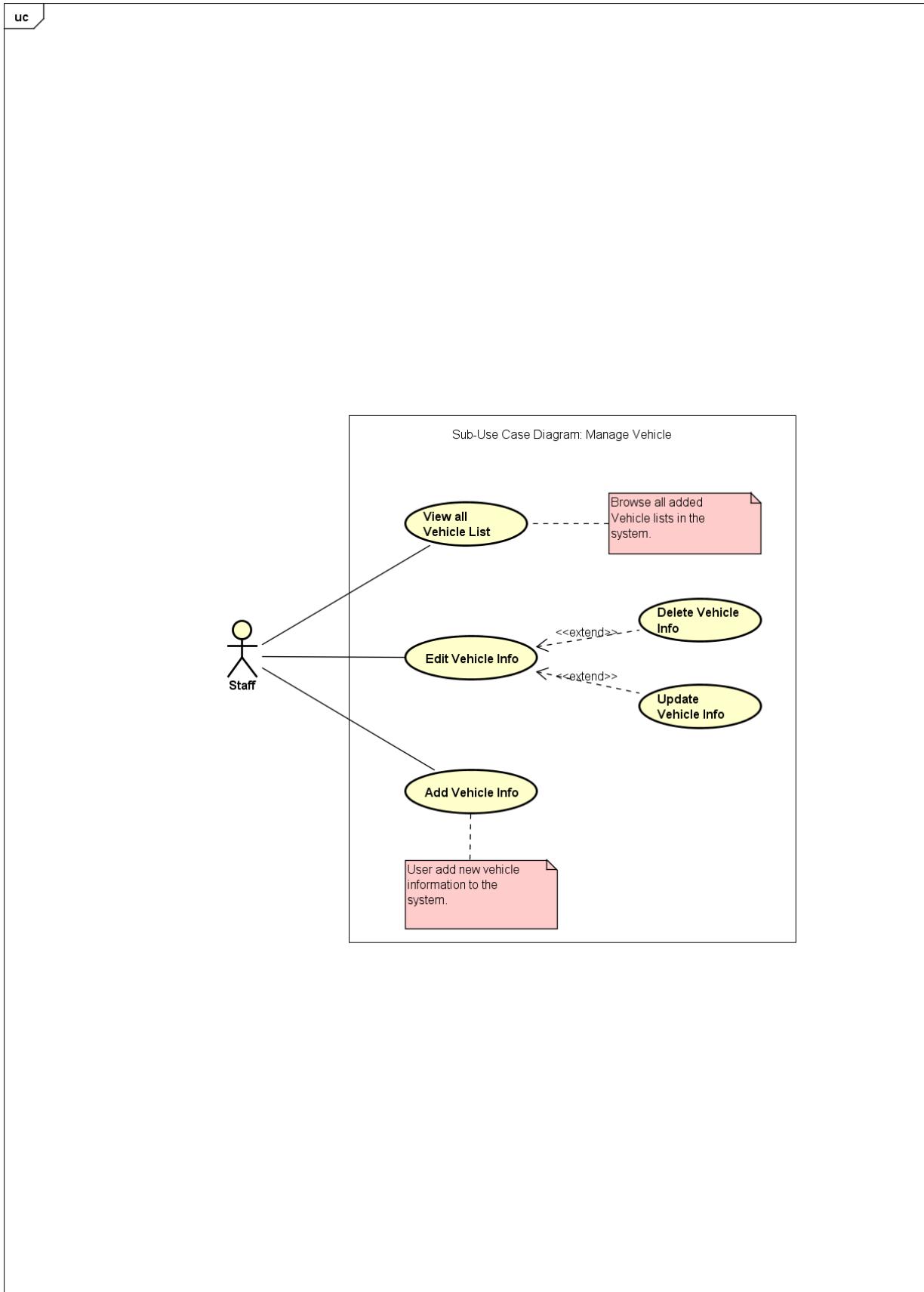
Alternative Course of action:

Line 2: If no Vehicle details are found then system will display no details found message.

Line 4: If the entered details is invalid then system will throw error message.

Line 6: If system cannot make changes to the current selected Vehicle details error message to try later will be displayed.

Line 8: If a Vehicle record cannot be deleted then system will display error message to try later.



3.4.1.12. Manage Route Information

Actors: Staff (Initiator)

Purpose:

Overview:

A staff add new route information including bus stops, distance between bus stops, longitude/latitude of bus stops etc. Similarly, staff update or delete any existing route and its related data.

Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff selects to view all route details. 3. Staff enter new route details and submit. 5. Staff make changes to the current value of route and submit. 7. Staff selects particular route details and click delete button.	2. Display route list. 4. Validates the new route details and record to the system. 6. Update the route details. 8. Delete the selected route details from the database.

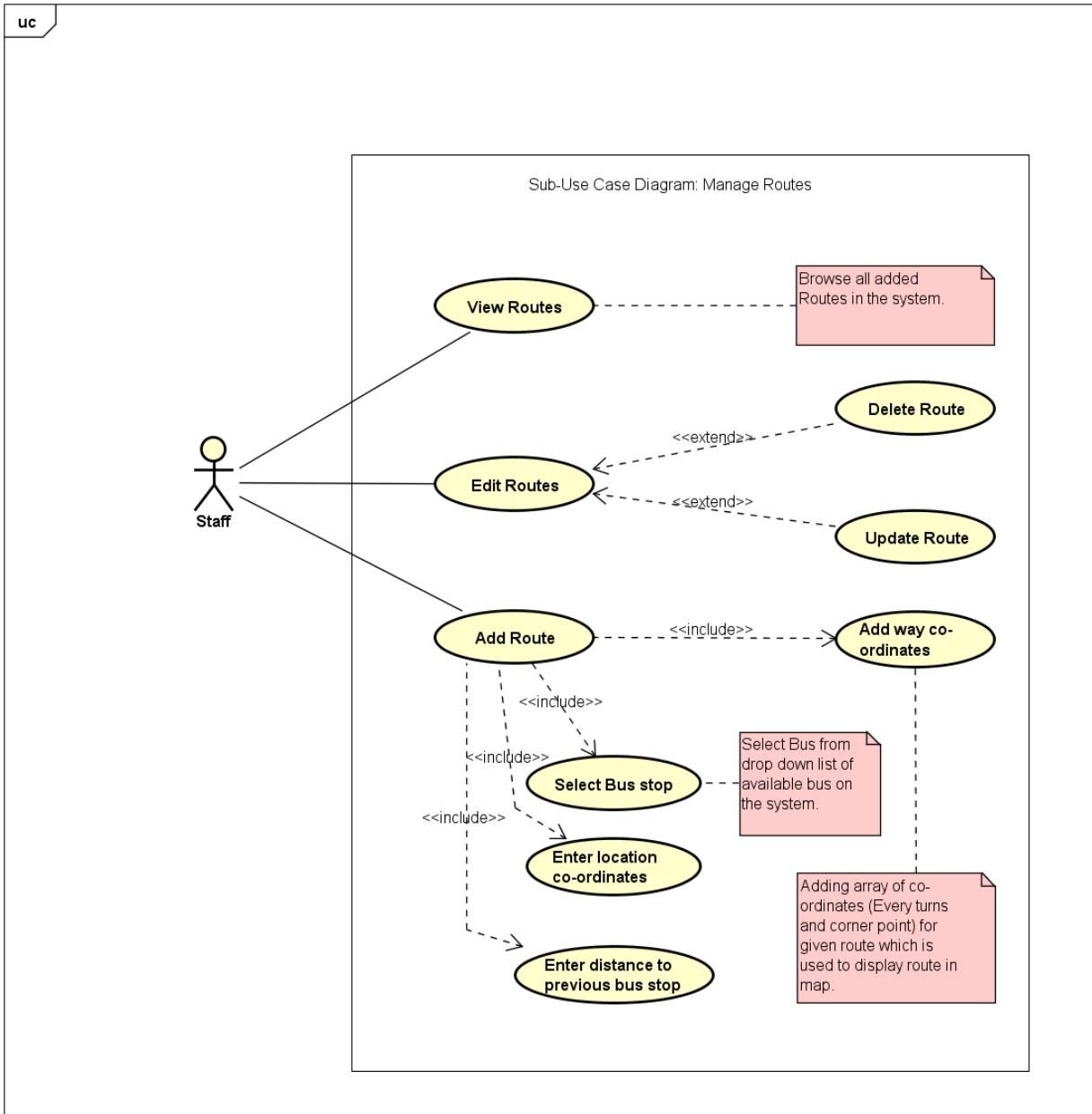
Alternative Course of action:

Line 2: If no route details are found then system will display no details found message.

Line 4: If the entered details is invalid then system will throw error message.

Line 6: If system cannot make changes to the current selected route details error message to try later will be displayed.

Line 8: If a route record cannot be deleted then system will display error message to try later.



3.4.1.13. Track Bus

Actors: Staff (Initiator)

Purpose:

Overview:

A staff track the location of all the Bus for all routes. He/she first choose the particular route and get all the Bus location for that route in an interactive map mode. In addition, staff flag a particular vehicle by executing ‘Flag Bus’ use case.

Type: Primary (Initiator)

Dialog:

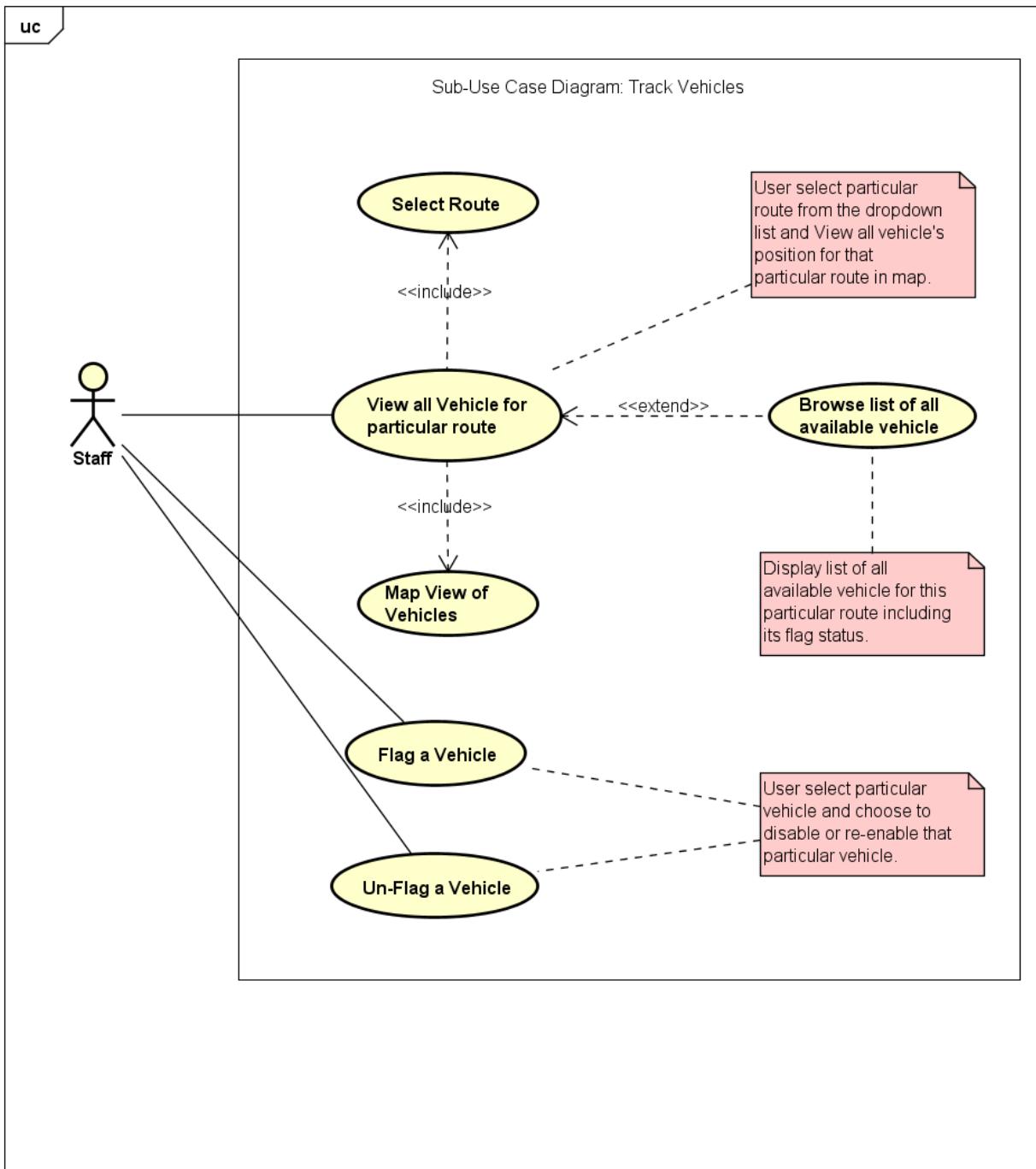
Actor	System
1. Staff select to view and browse the route list. 3. Staff select a route from the list. 5. Staff views all the vehicle position in map or in list view. 6. Staff select to flag vehicle from the list. 7. Staff also selects to re-enable any disabled vehicle.	2. Display list of available routes. 4. Get list of available active vehicles and display in map. 7. Disable the selected vehicle from the system. 8. Enable the selected vehicle and make it active.

Alternative Course of action:

Line 2: If any routes are not available then system displays not available message.

Line 7: If the vehicle cannot be flagged, system displays error message.

Line 8: If the vehicle cannot be un-flagged, then system displays error message.



3.4.1.14. Authenticate Staff

Actors: Staff (Initiator)

Purpose:

Overview:

A Staff login to the system by providing login credentials. Access will be granted if the login credentials are correct otherwise access to the system will be denied.

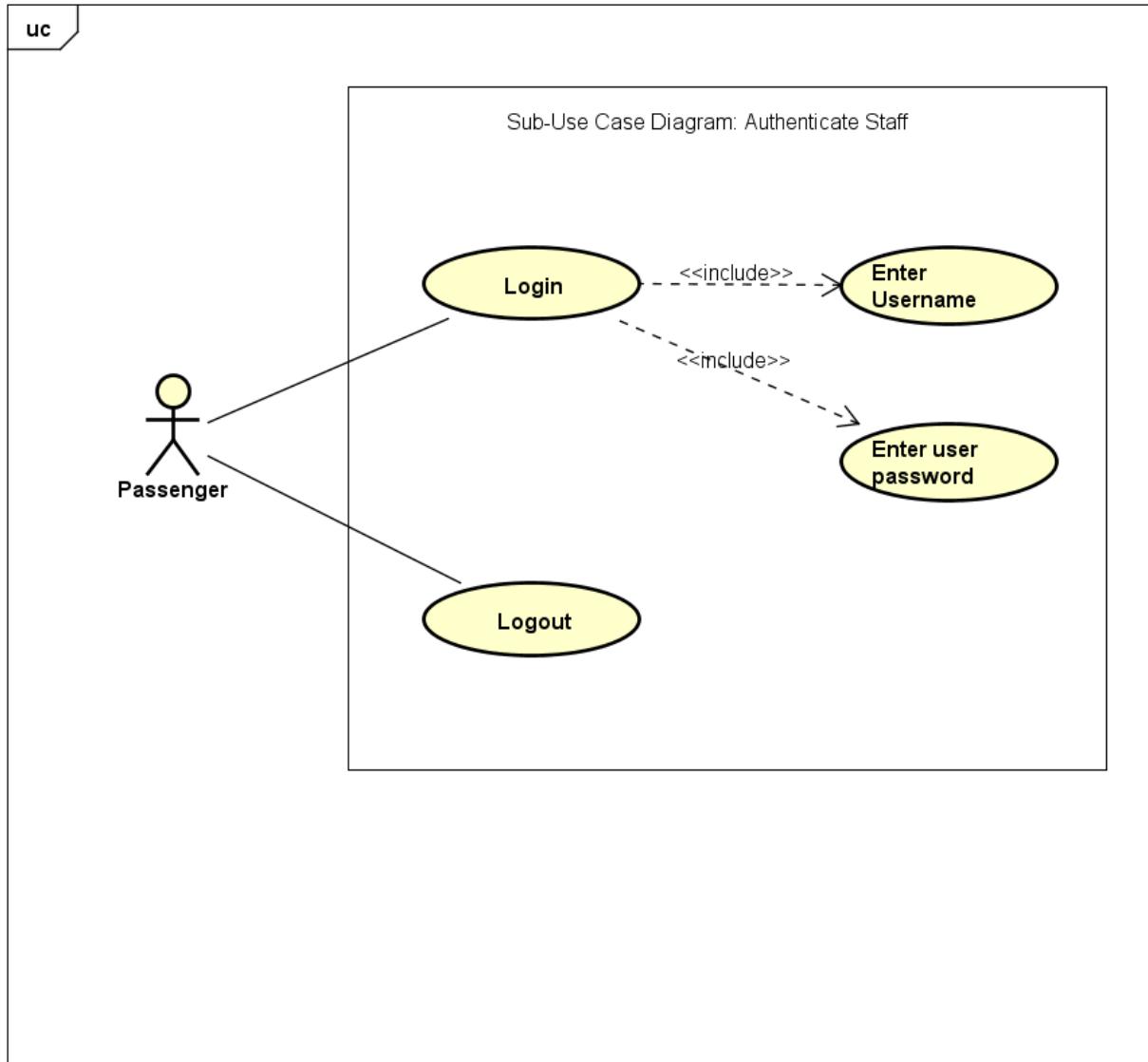
Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff navigate to the login page. 3. Staff enter user name and password. 6. Staff view the home page.	2. System displays the login page. 4. System checks the username and password. 5. If username and password is correct navigate to the home page.

Alternative Course of action:

Line 4: If the provided username and password is not correct, system re-direct to the login page displaying username/password error message.



3.4.1.15. Generate Reports

Actors: Staff (Initiator)

Purpose:

Overview:

A staff generate various reports based on the requirements.

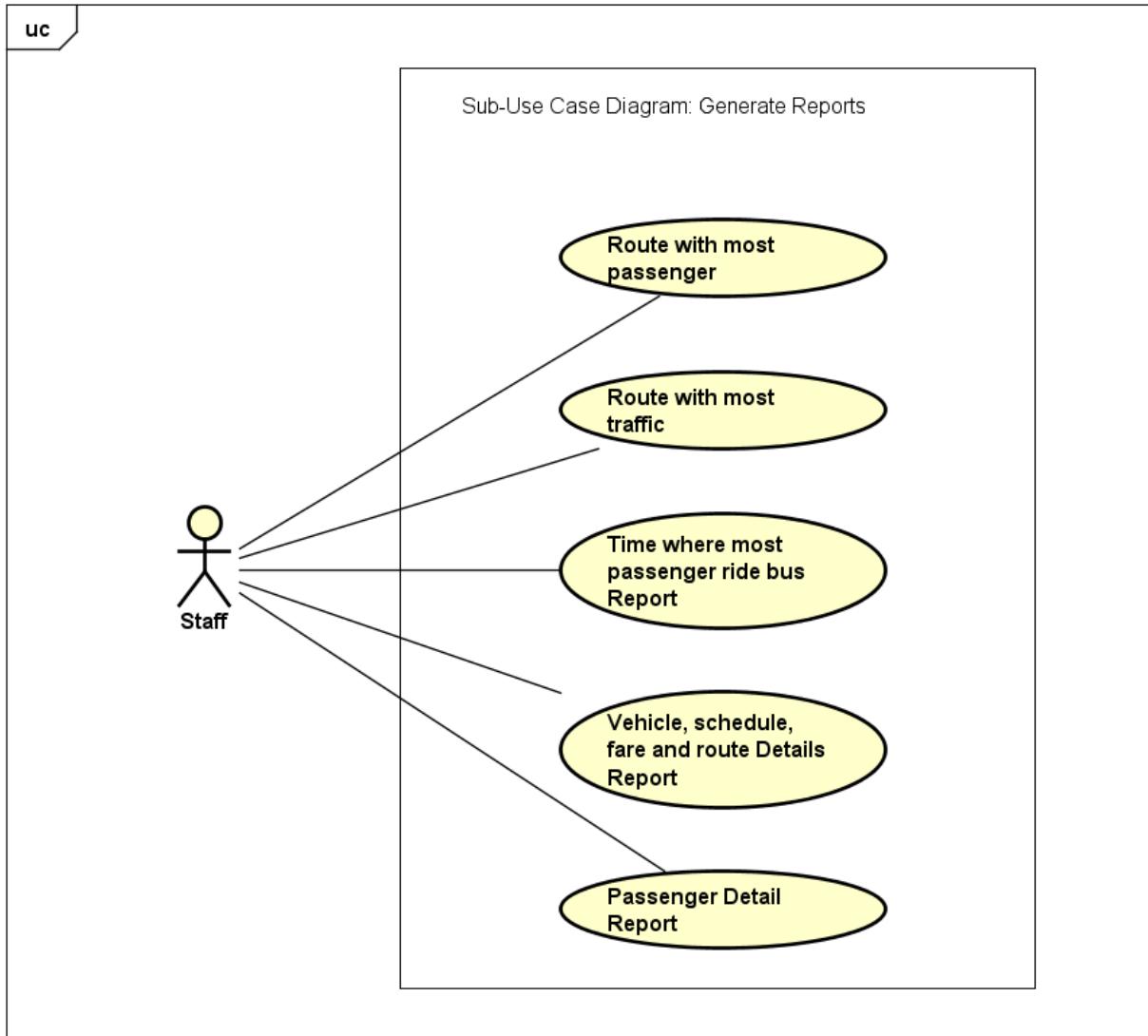
Type: Primary (Initiator)

Dialog:

Actor	System
1. Staff select to view the particular report.	2. System display the report details.

Alternative Course of action:

Line 2: If the report is not available then system will display report not available message.



3.4.1.16. Update vehicle location

Actors: Driver (Initiator)

Purpose:

Overview:

A driver first finds the correct route and vehicle name relevant to them and get the schedule in order to start vehicle. Then, driver can either start by clicking start button or stop the location update by clicking stop button.

Type: Primary (Initiator)

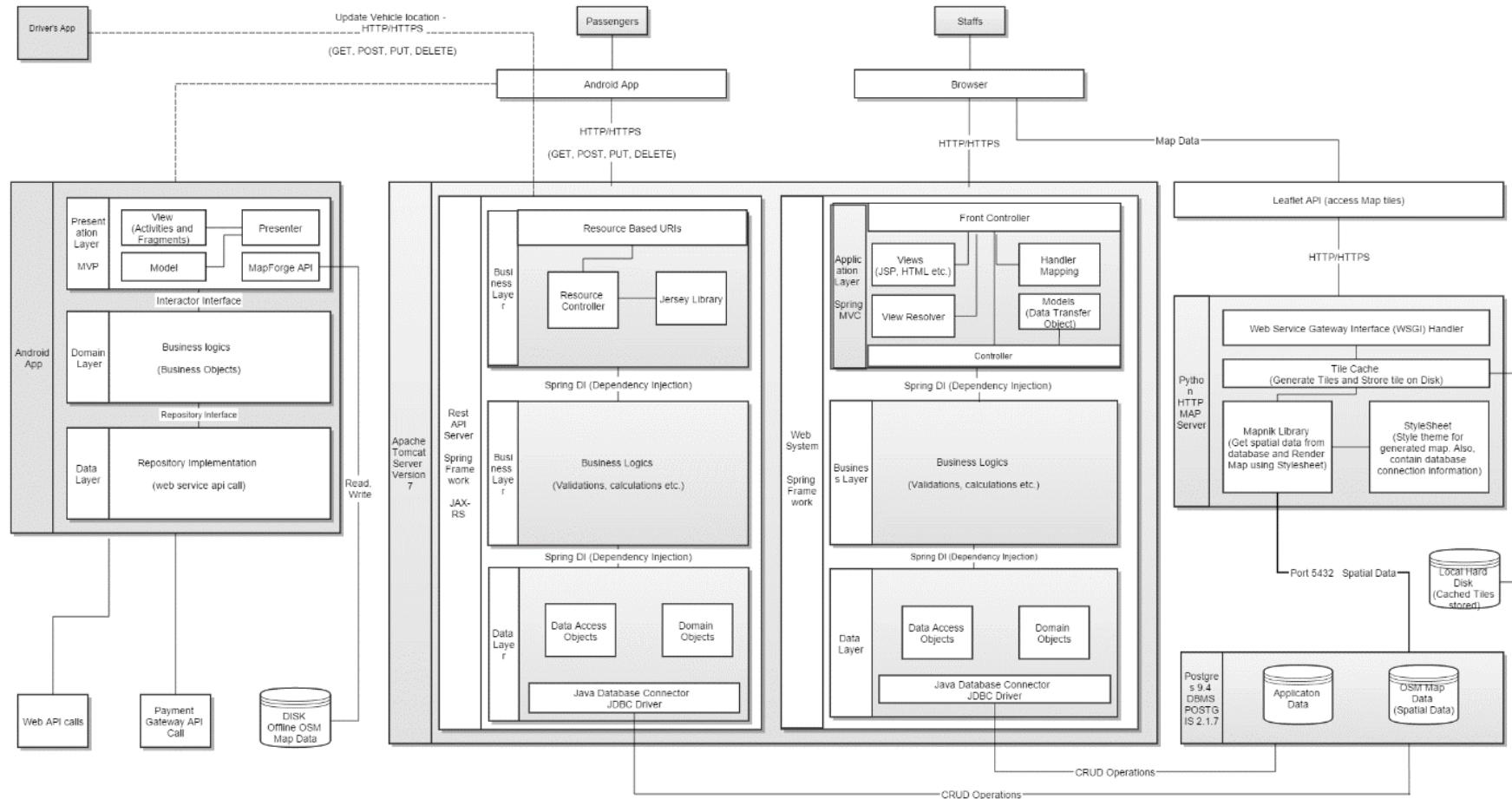
Dialog:

Actor	System
1. Driver click to find schedule. 4. Select route name and vehicle name from the form and then submit. 7. Click Start button to update the current location of the device location. 9. Click stop button to delete the current vehicle session.	2. System get route and vehicle details. 3. Then display the schedule search form. 5. Check if the schedule is available or not. 6. Display the location update page along with vehicle schedule detail. 8. update vehicle location in loop. 10. Delete the vehicle session.

Alternative Course of action:

Line 2: If no schedules are available, then system must display proper error message.

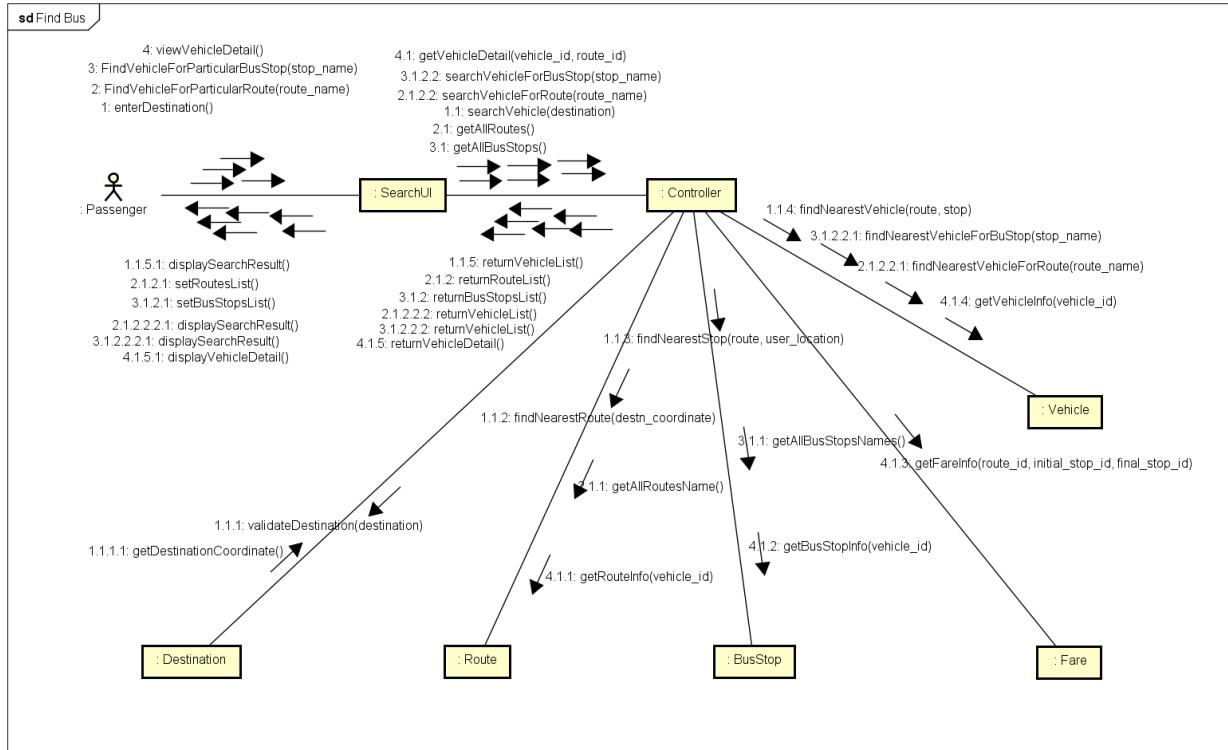
3.4.2. Architecture Design



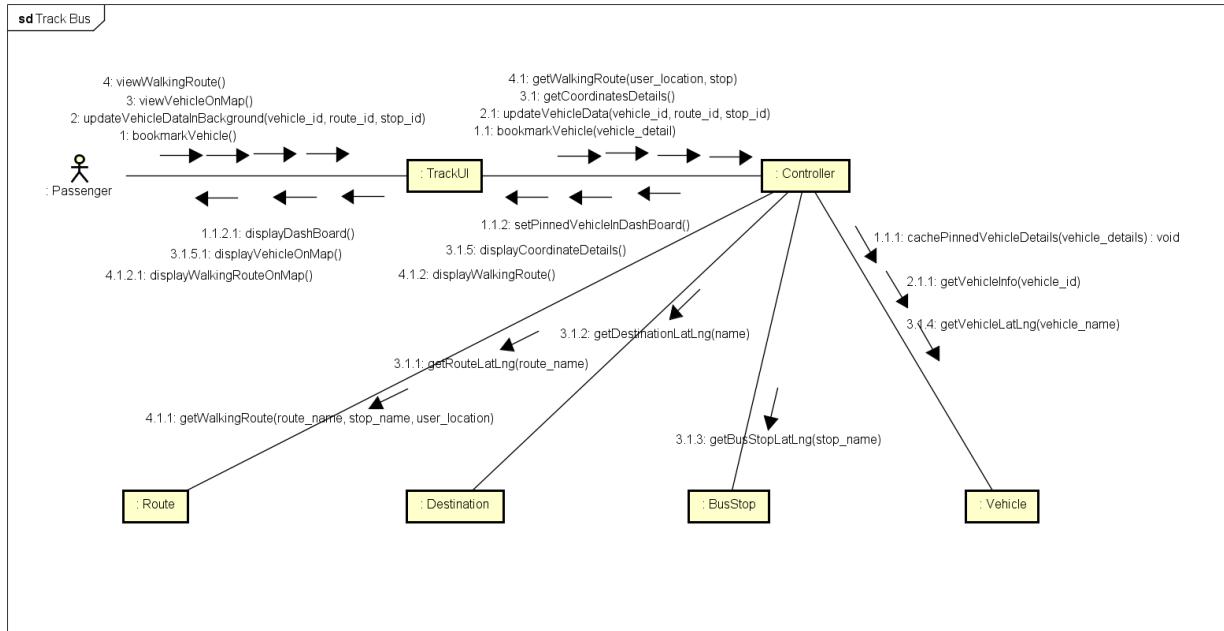
3.4.3. Process Designs

3.4.3.1. Communication Diagram

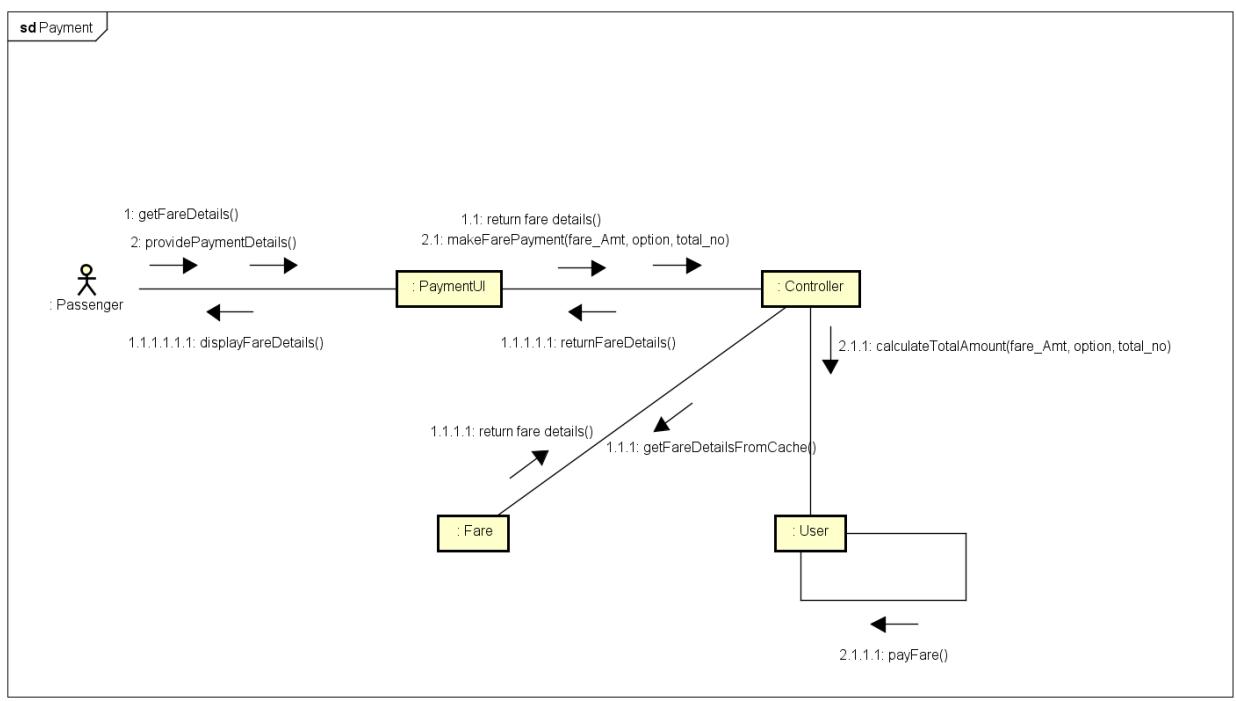
3.4.3.1.1. Find/Search Bus



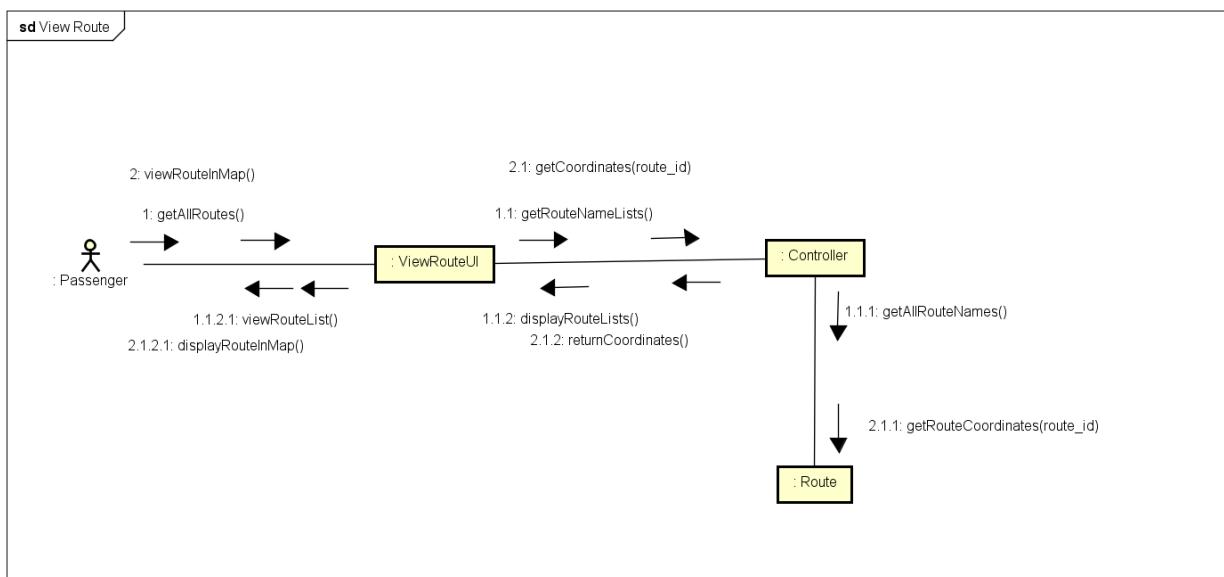
3.4.3.1.2. Track Bus



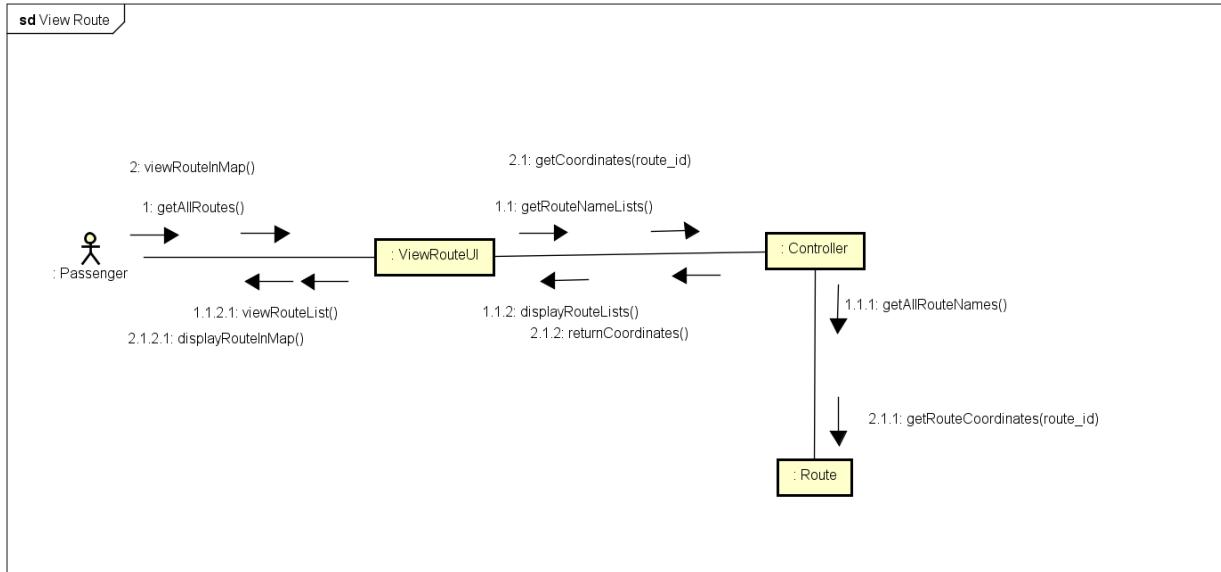
3.4.3.1.3. Payment



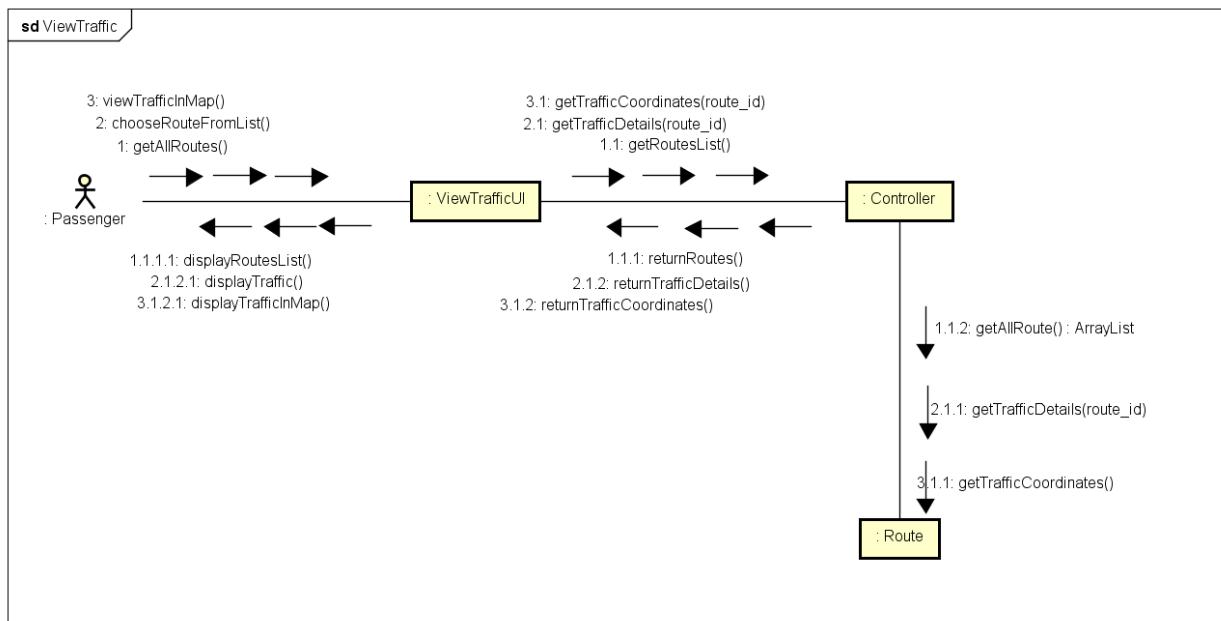
3.4.3.1.4. View Route



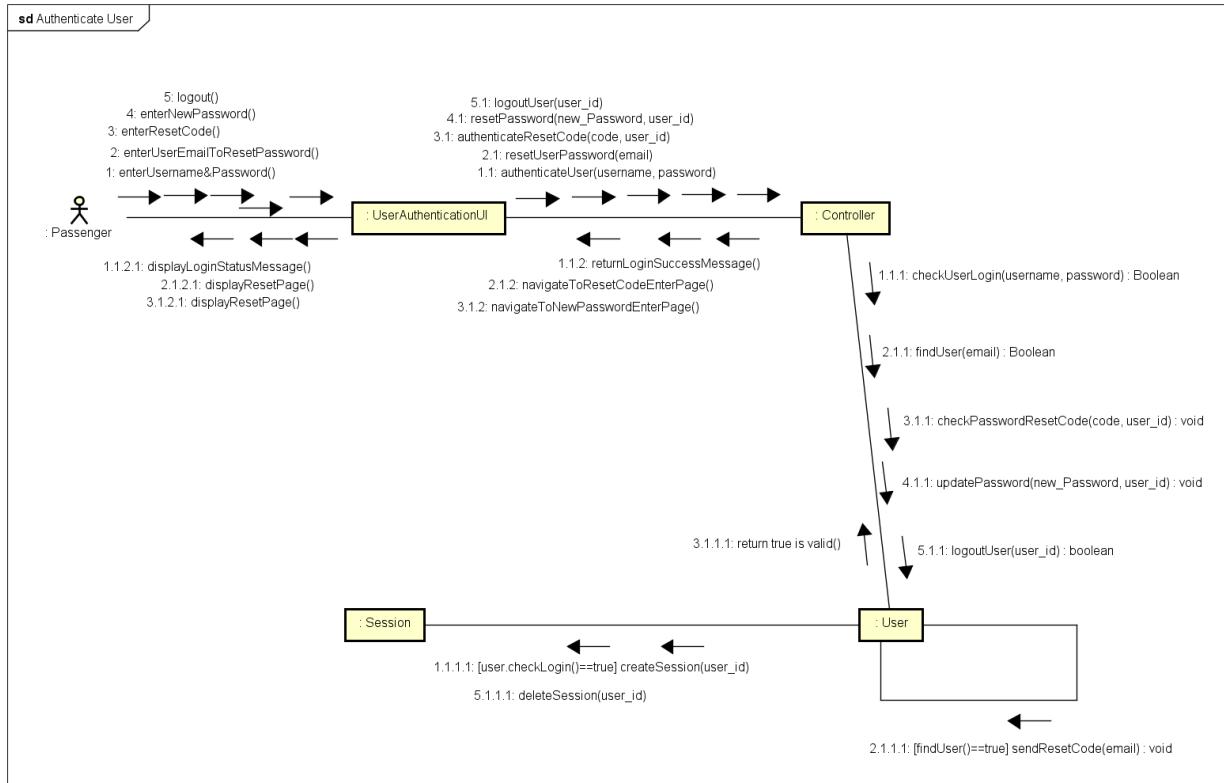
3.4.3.1.5. View Route



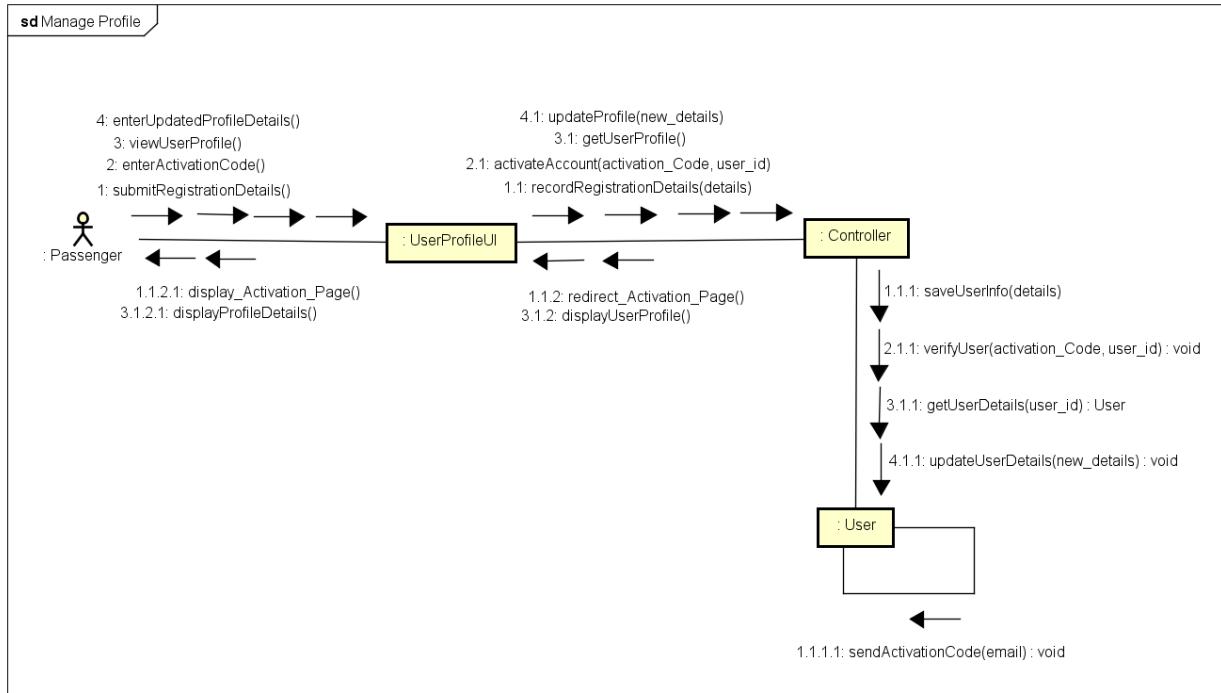
3.4.3.1.6. View Traffic



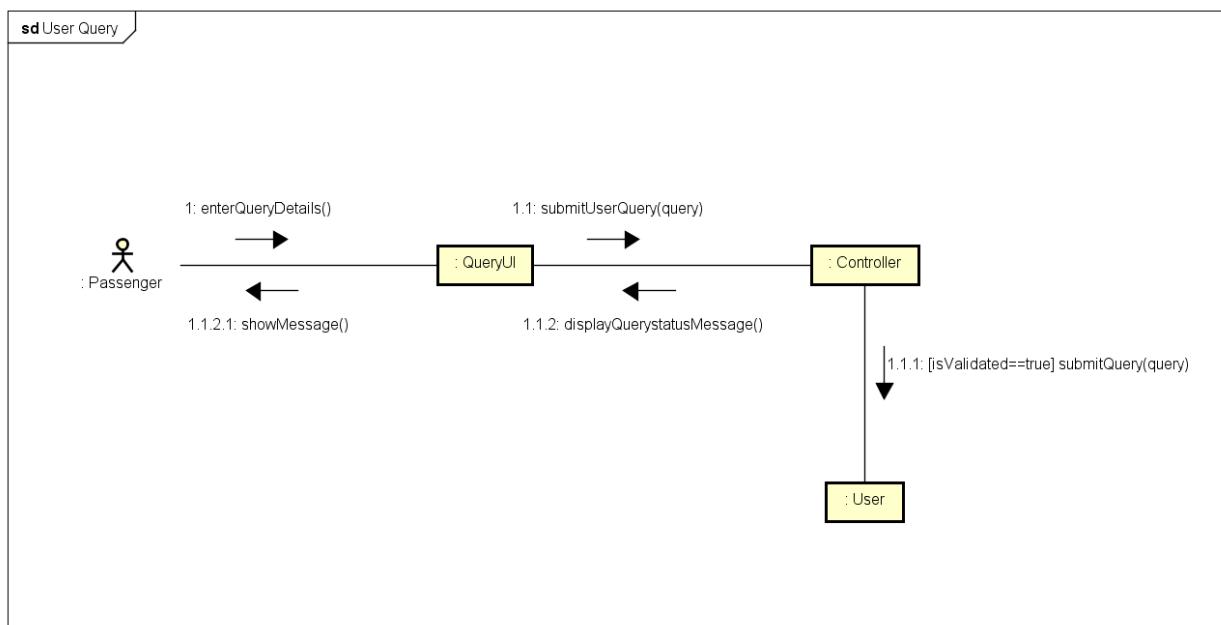
3.4.3.1.7. Authenticate User



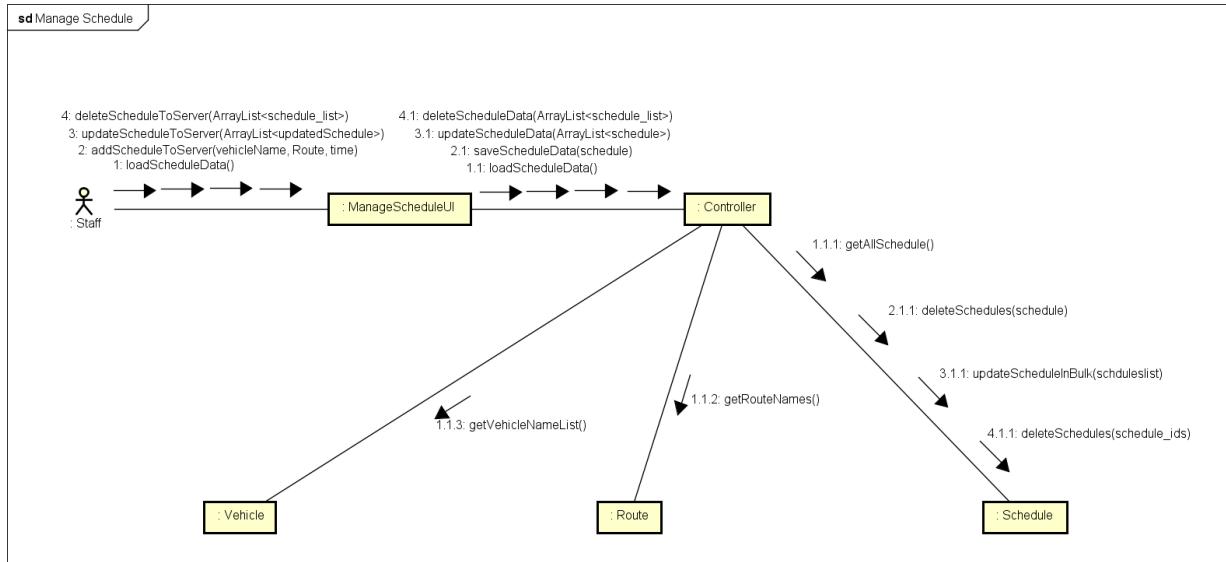
3.4.3.1.8. Manage Profile



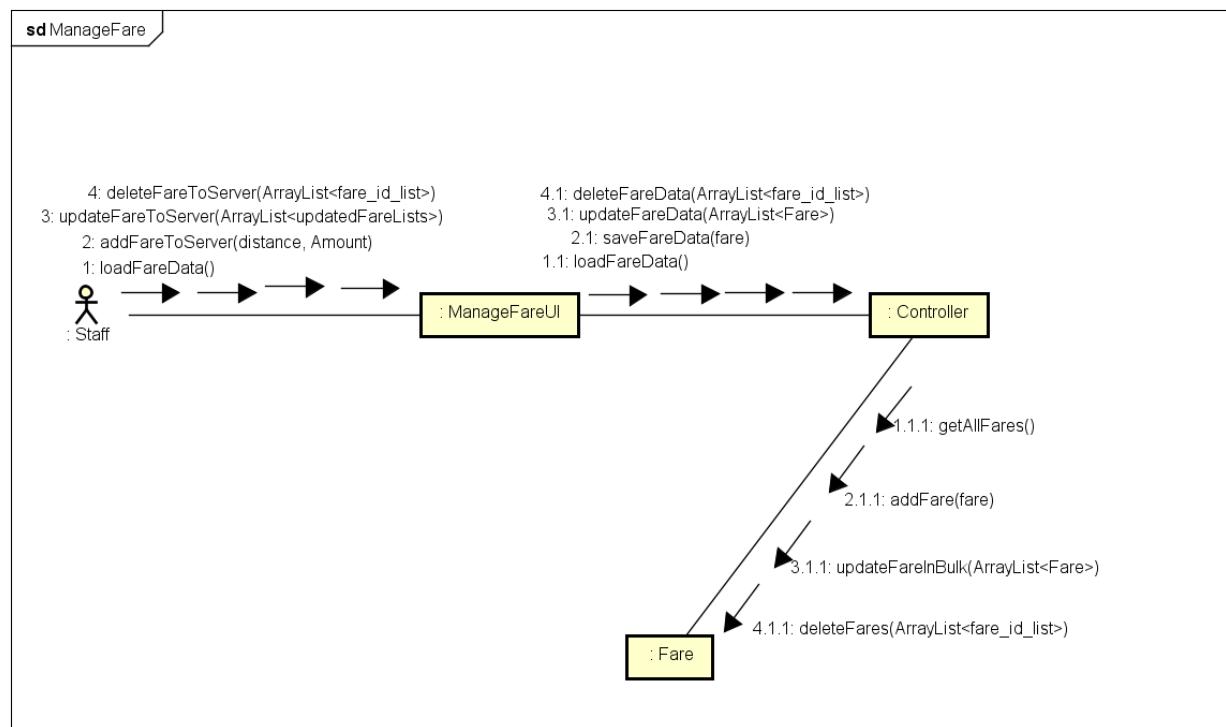
3.3.3.1.8. Query and Feedback



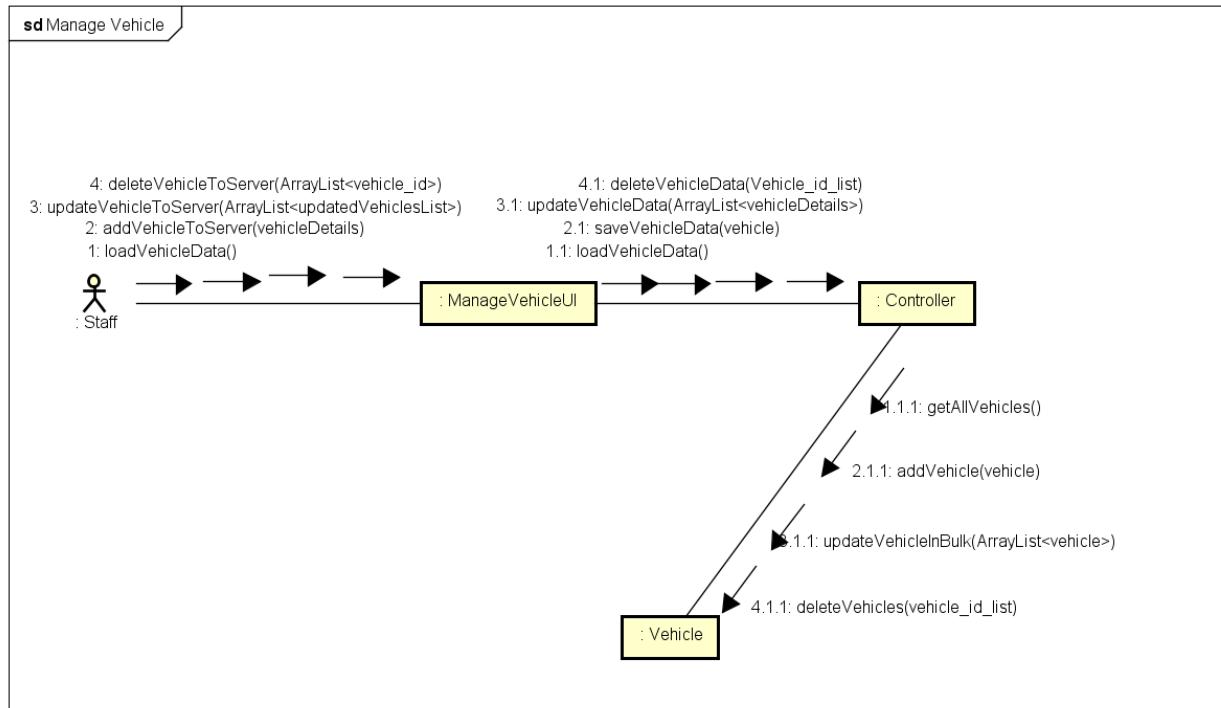
3.4.3.1.9. Manage Schedule



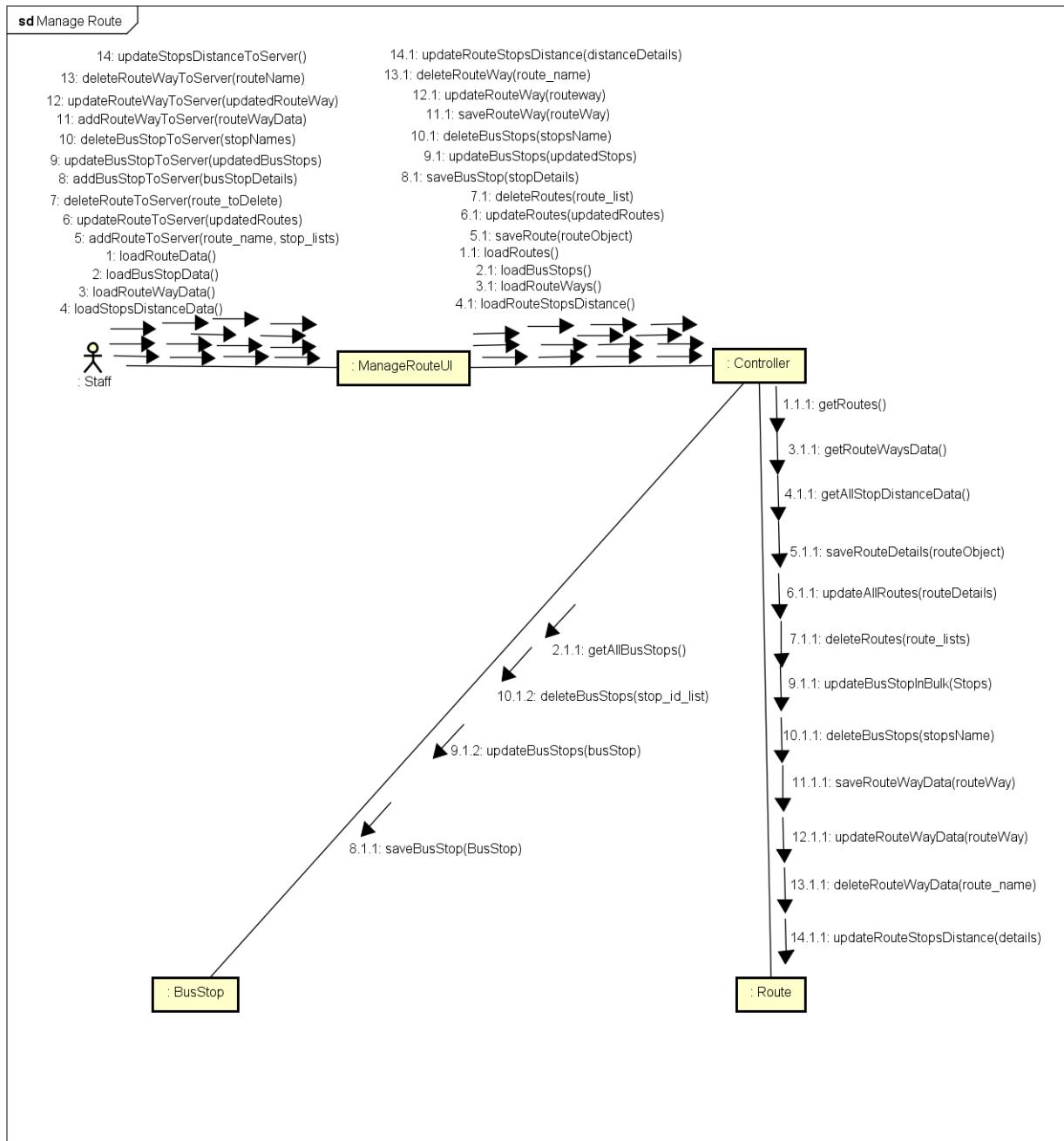
3.4.3.1.10. Manage Fare



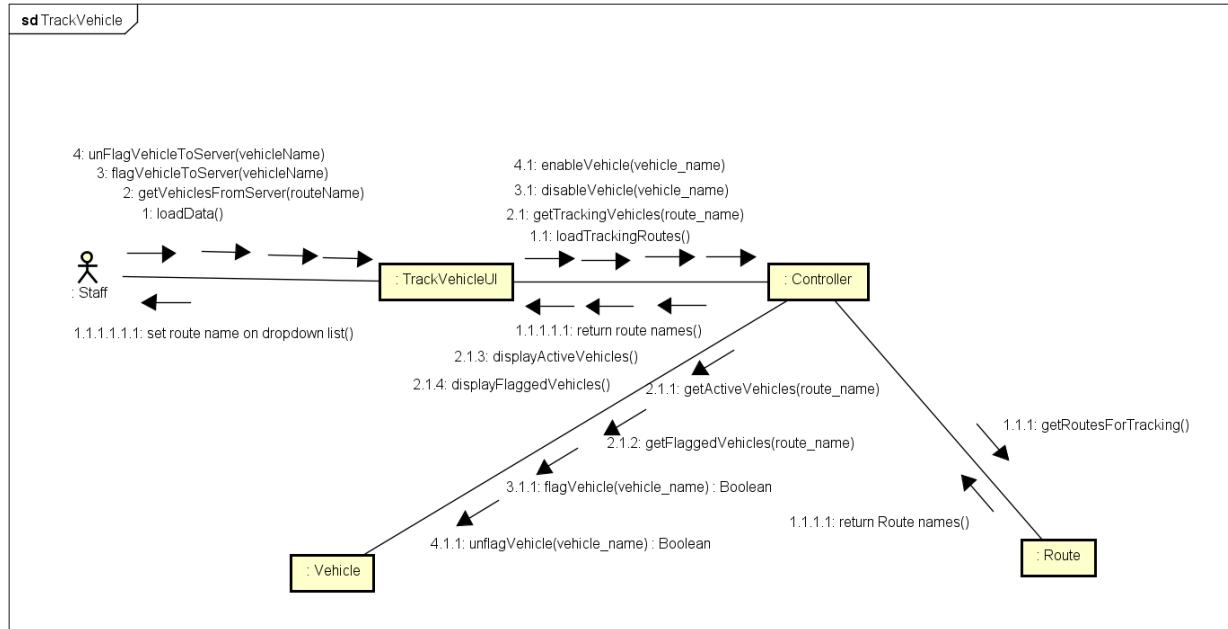
3.4.3.1.11. Manage Vehicle Information



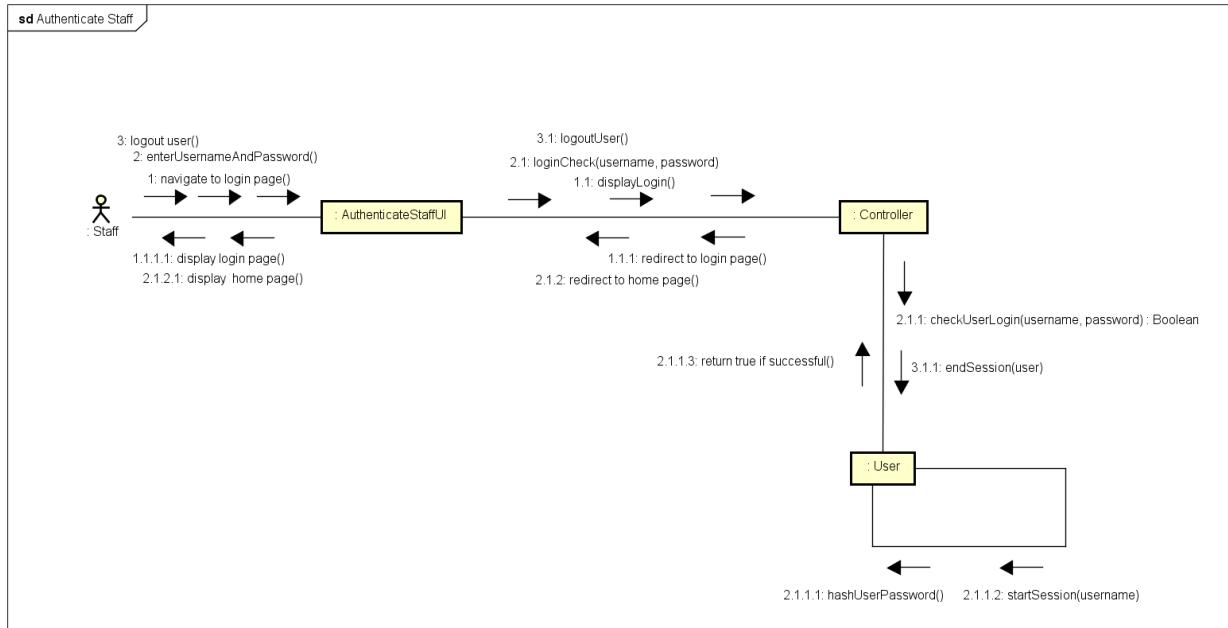
3.4.3.1.12. Manage Route Information



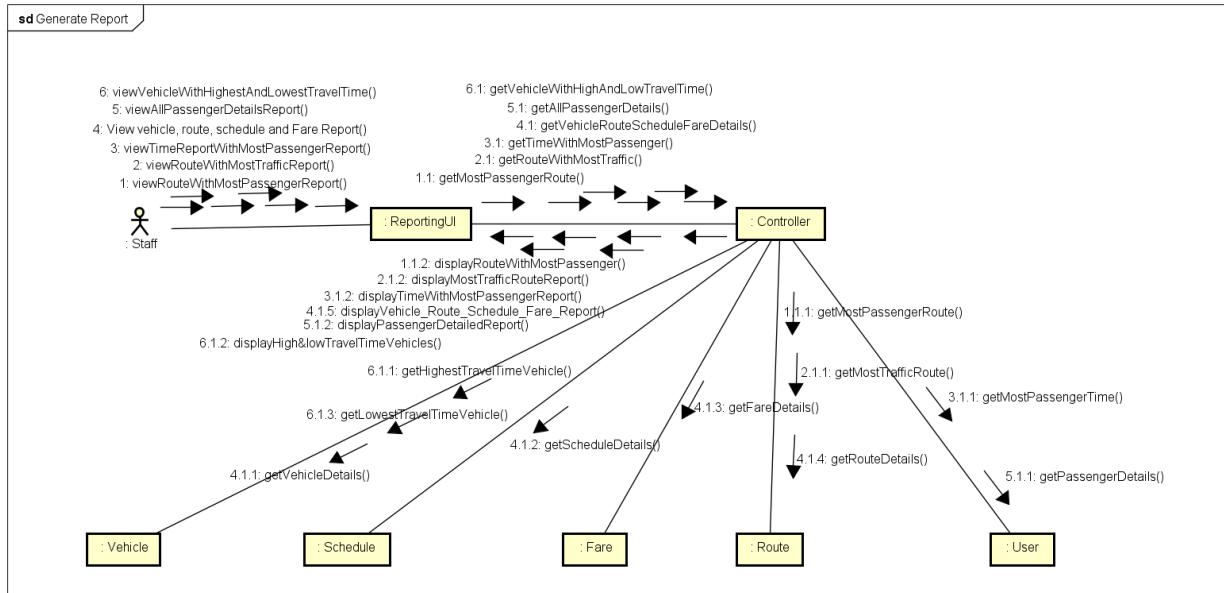
3.4.3.1.13. Track Bus



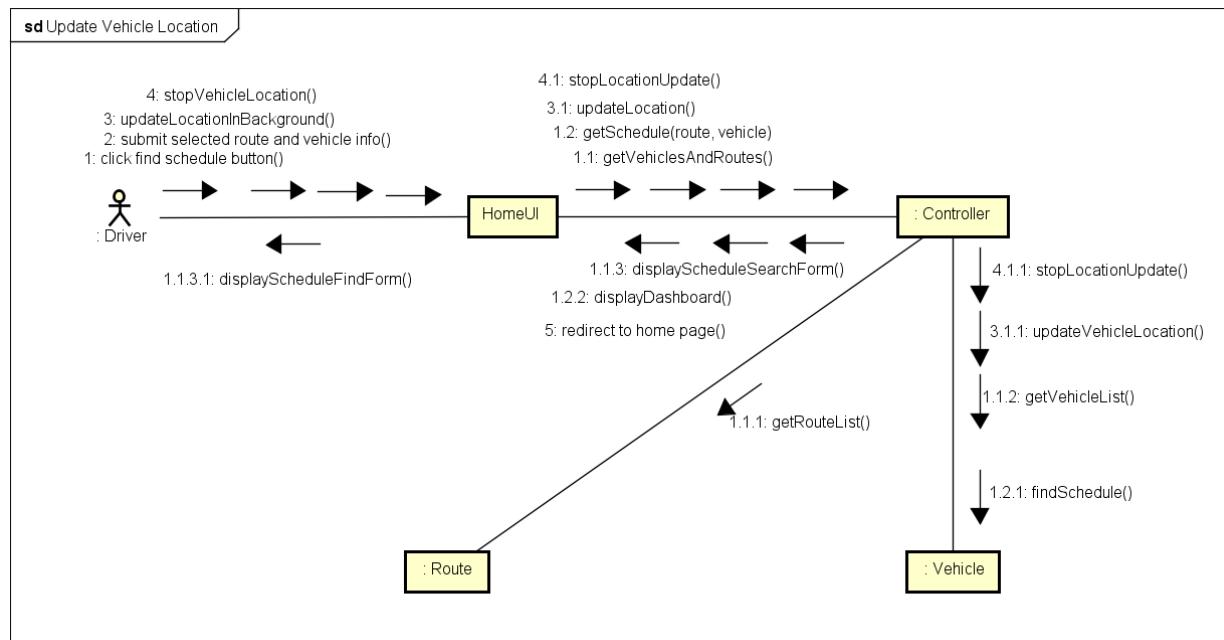
3.4.3.1.14. Authenticate Staff



3.4.3.1.15. Generate Reports

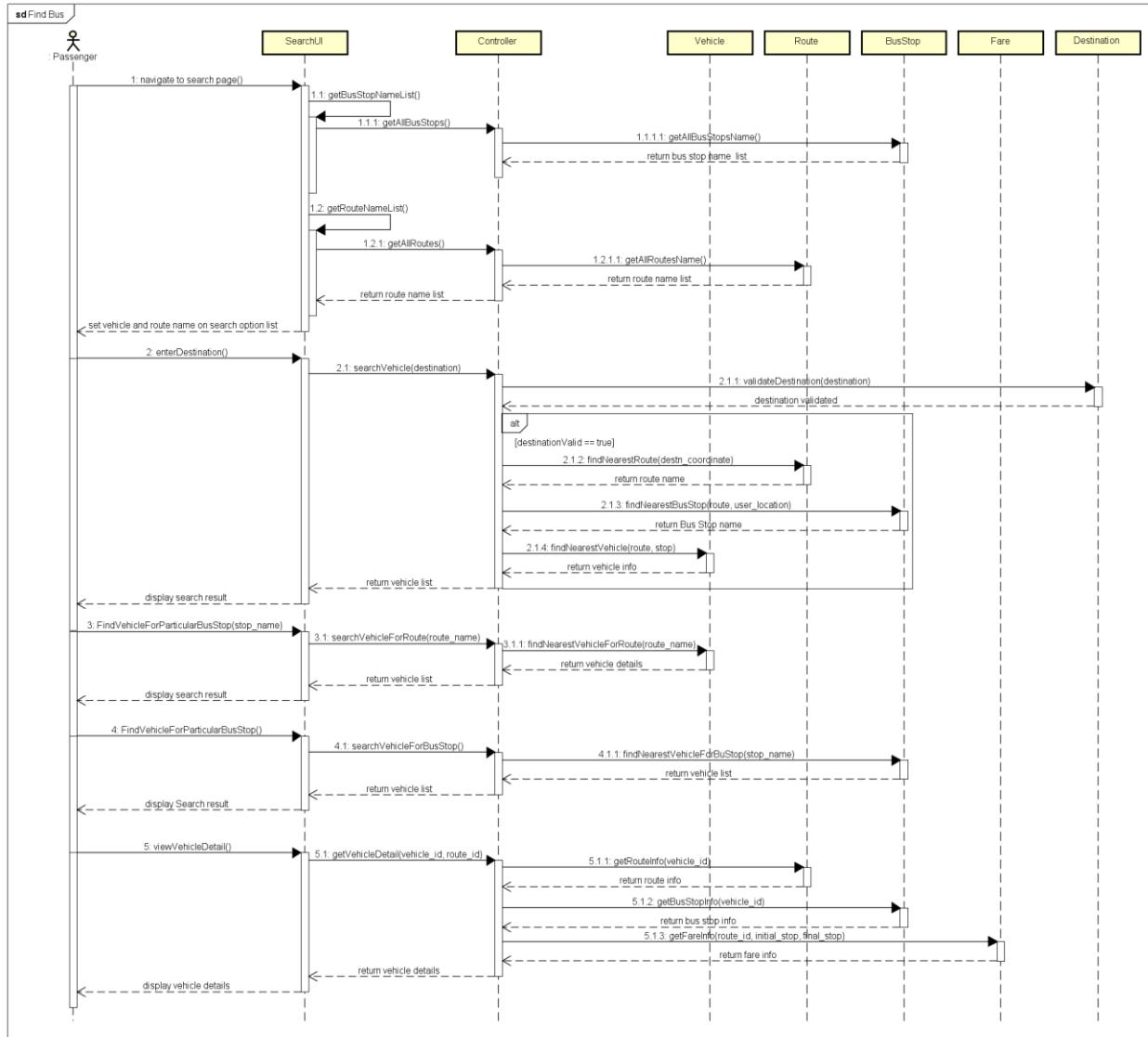


3.4.3.1.16. Update Vehicle Location

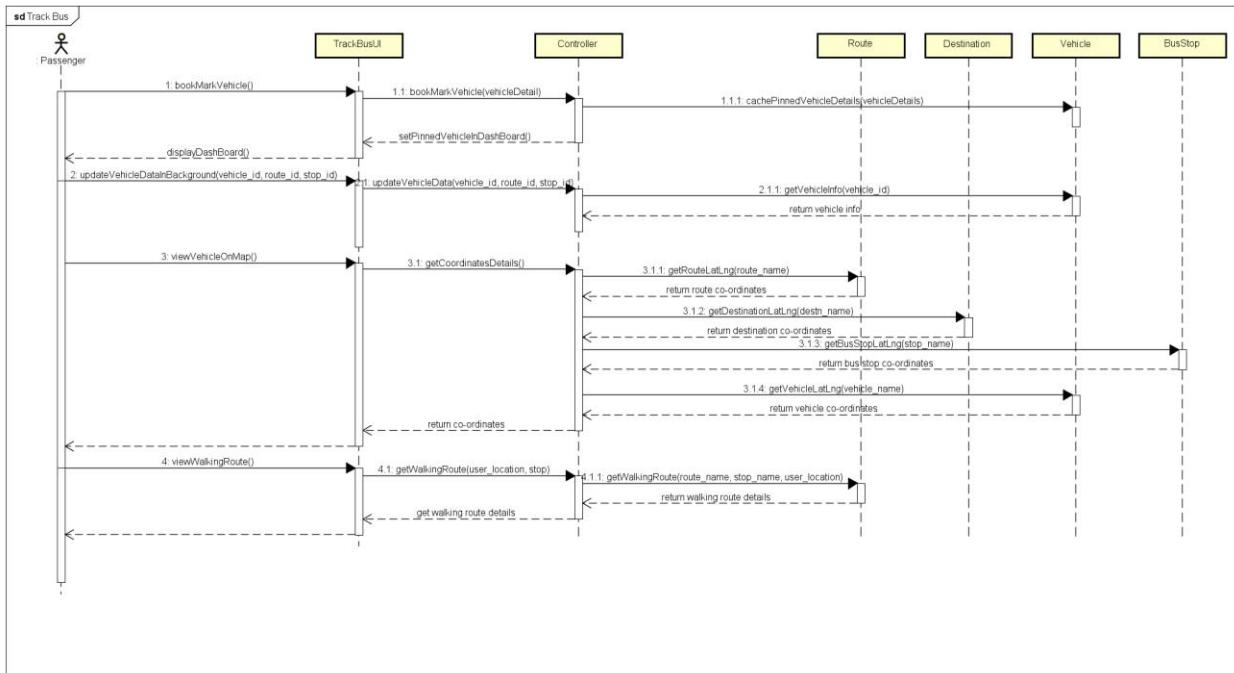


3.4.3.2. Sequence Diagram

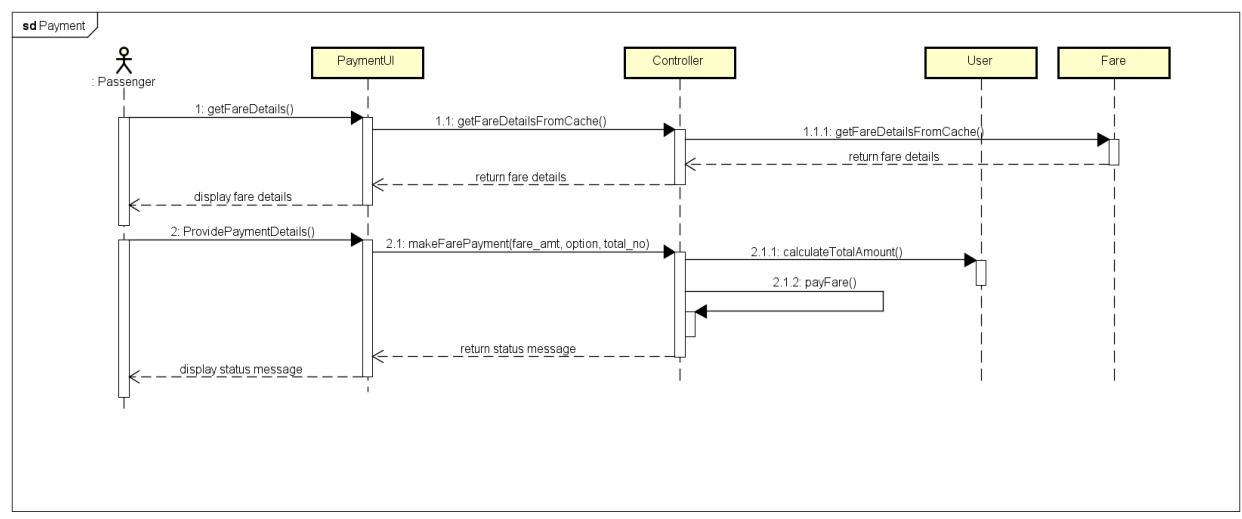
3.4.3.2.1. Find/Search Bus



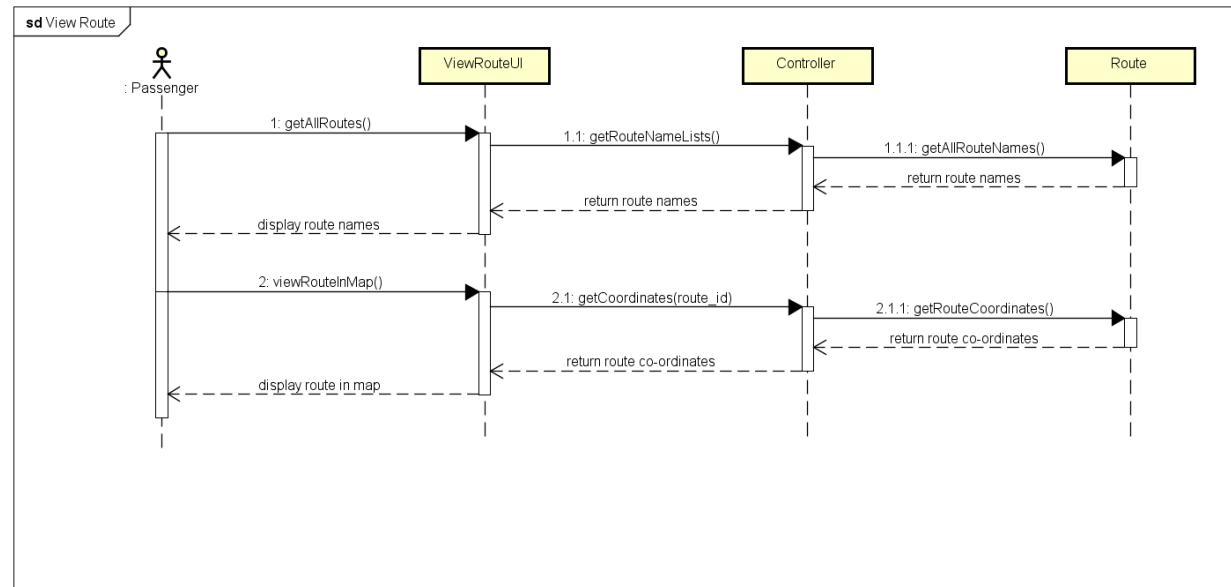
3.4.3.2.2. Track Bus



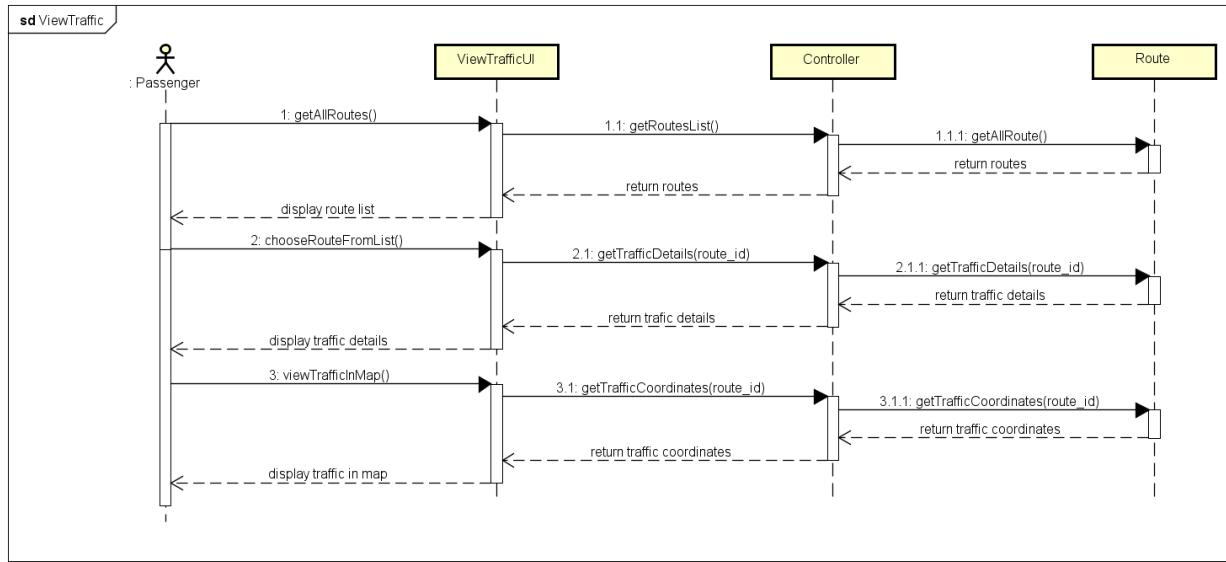
3.4.3.2.3. Payment



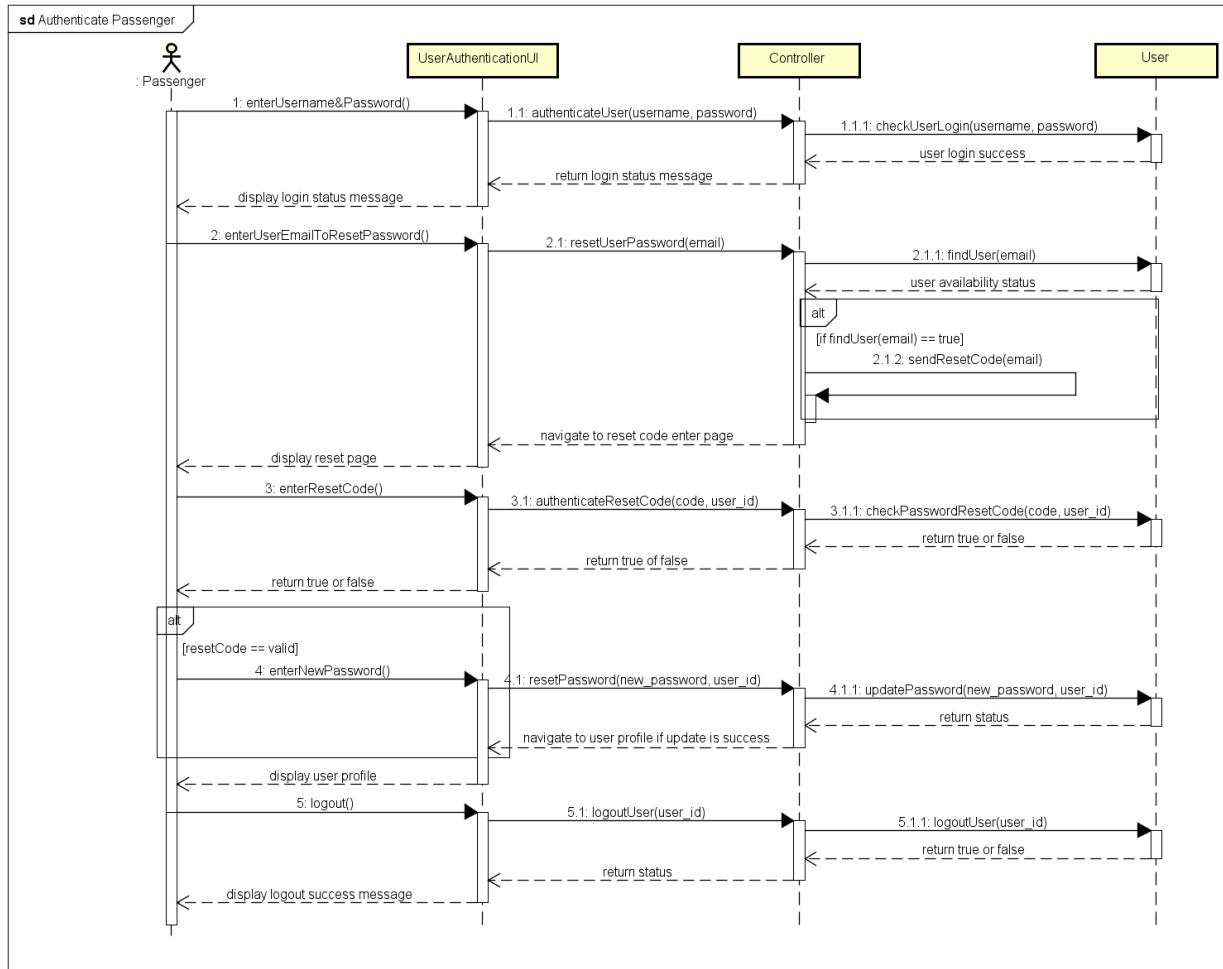
3.4.3.2.4. View Route



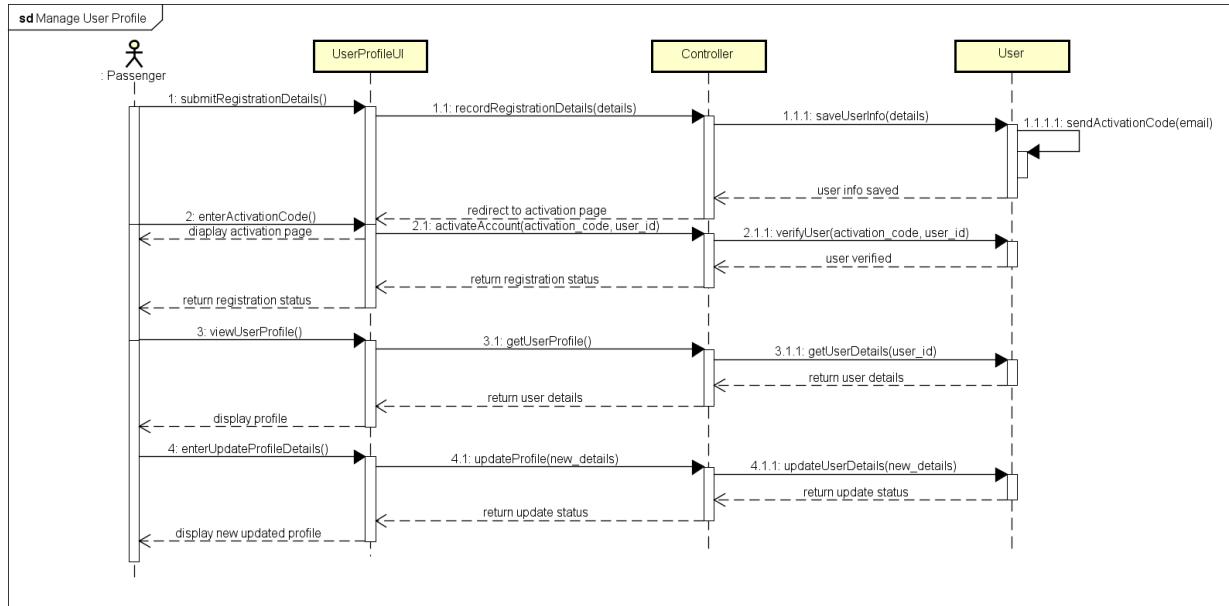
3.4.3.2.5. View Traffic



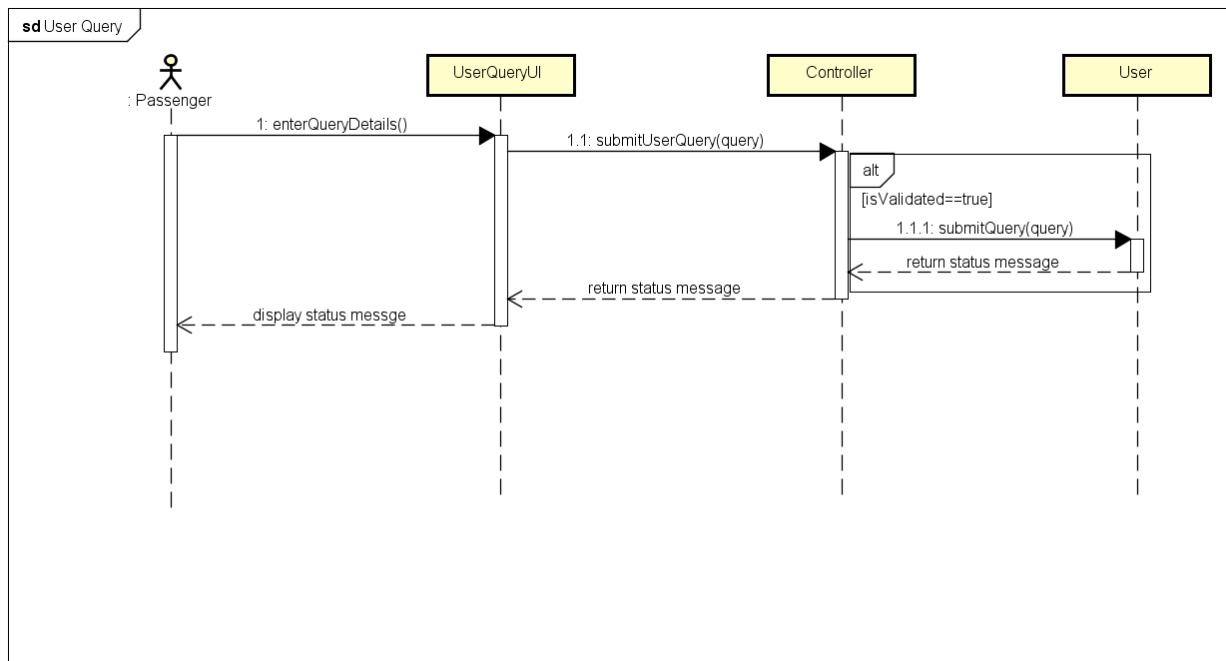
3.4.3.2.6. Authenticate User



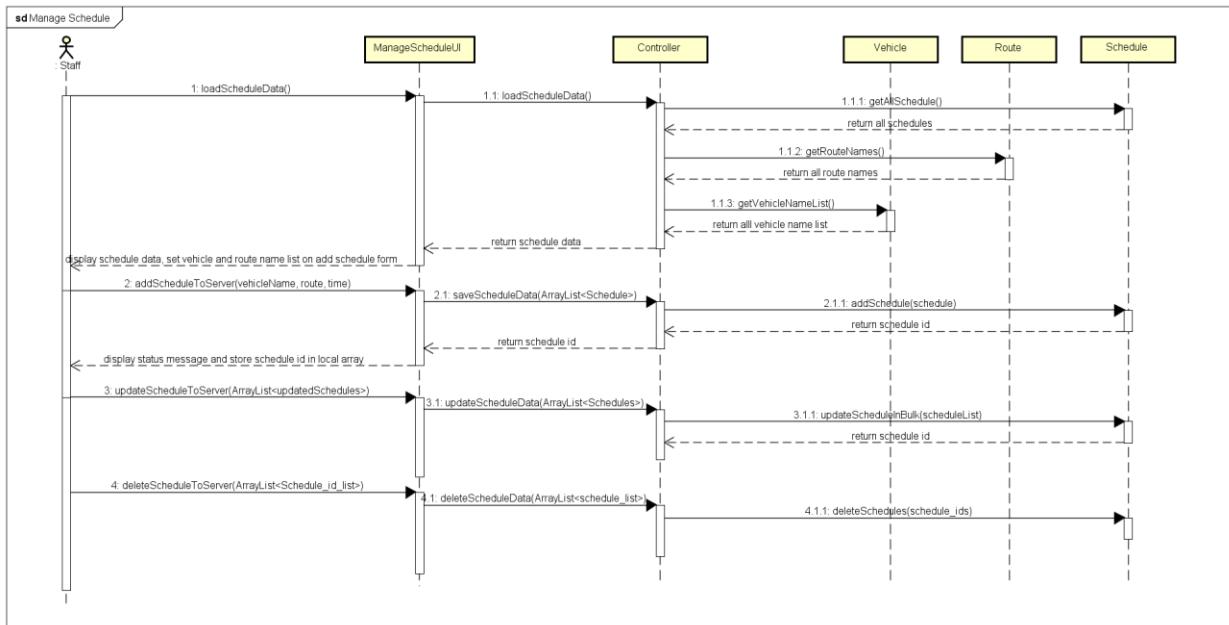
3.4.3.2.7. Manage Profile



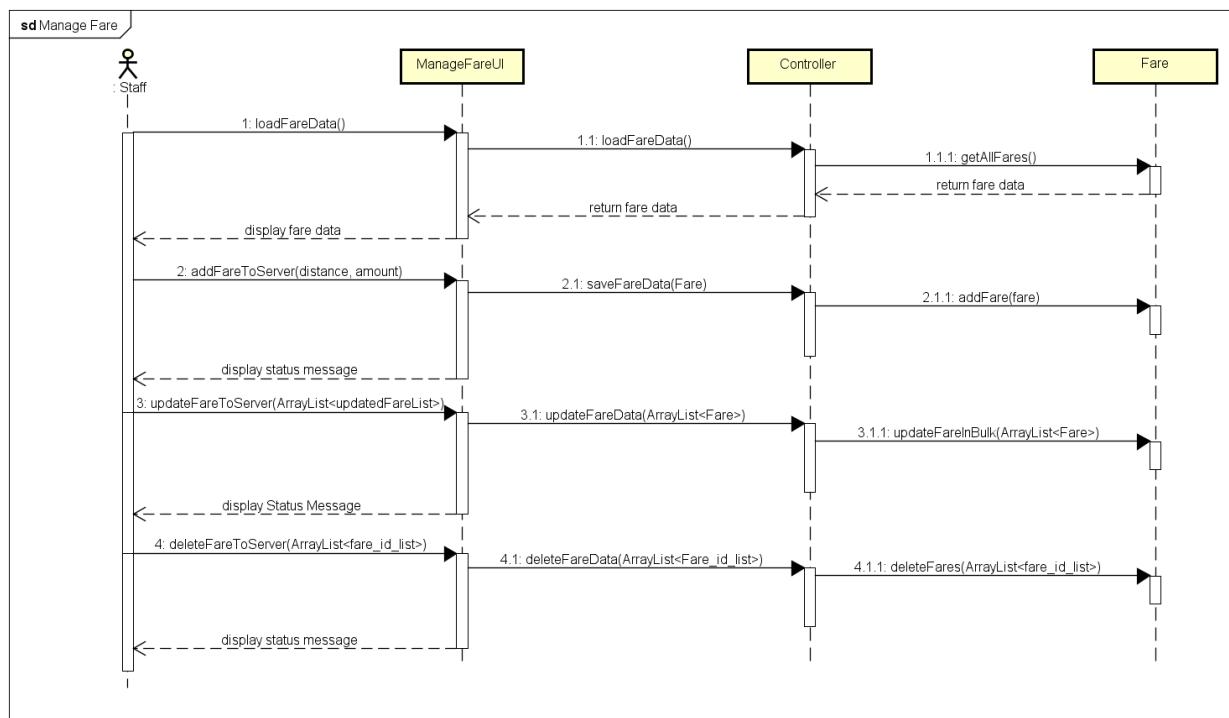
3.4.3.2.8. Query and Feedback



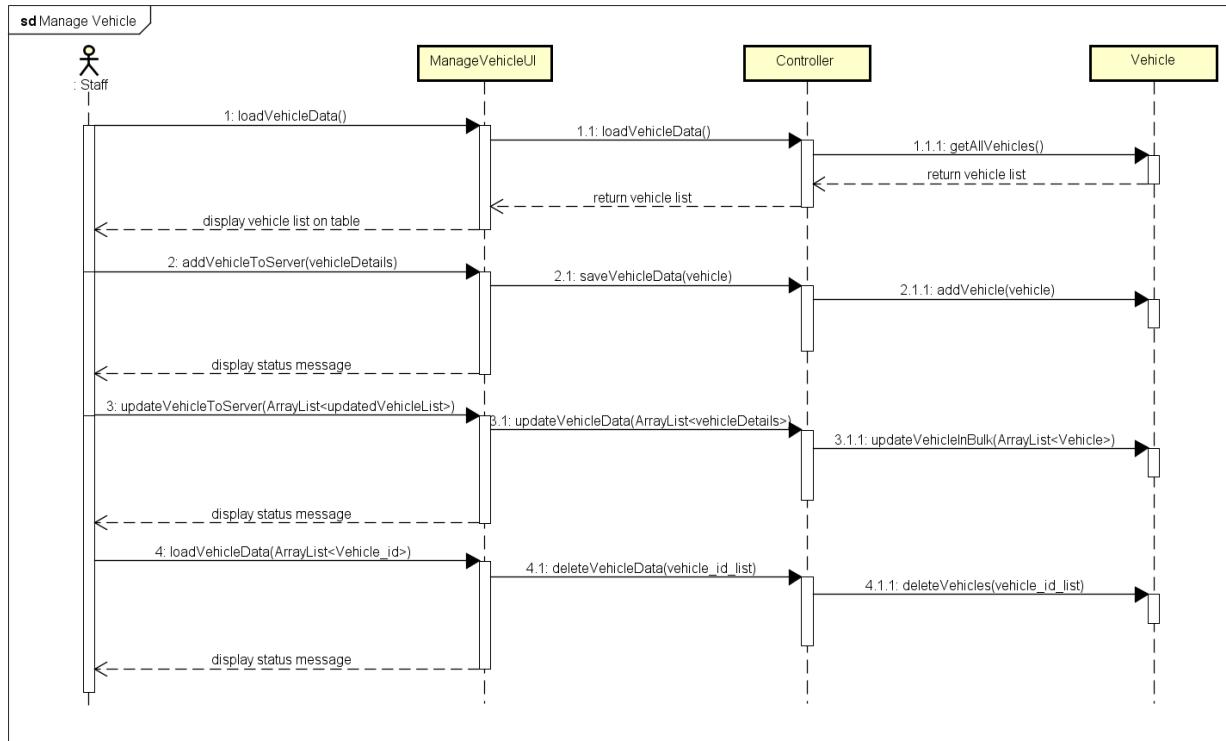
3.4.3.2.9. Manage Schedule



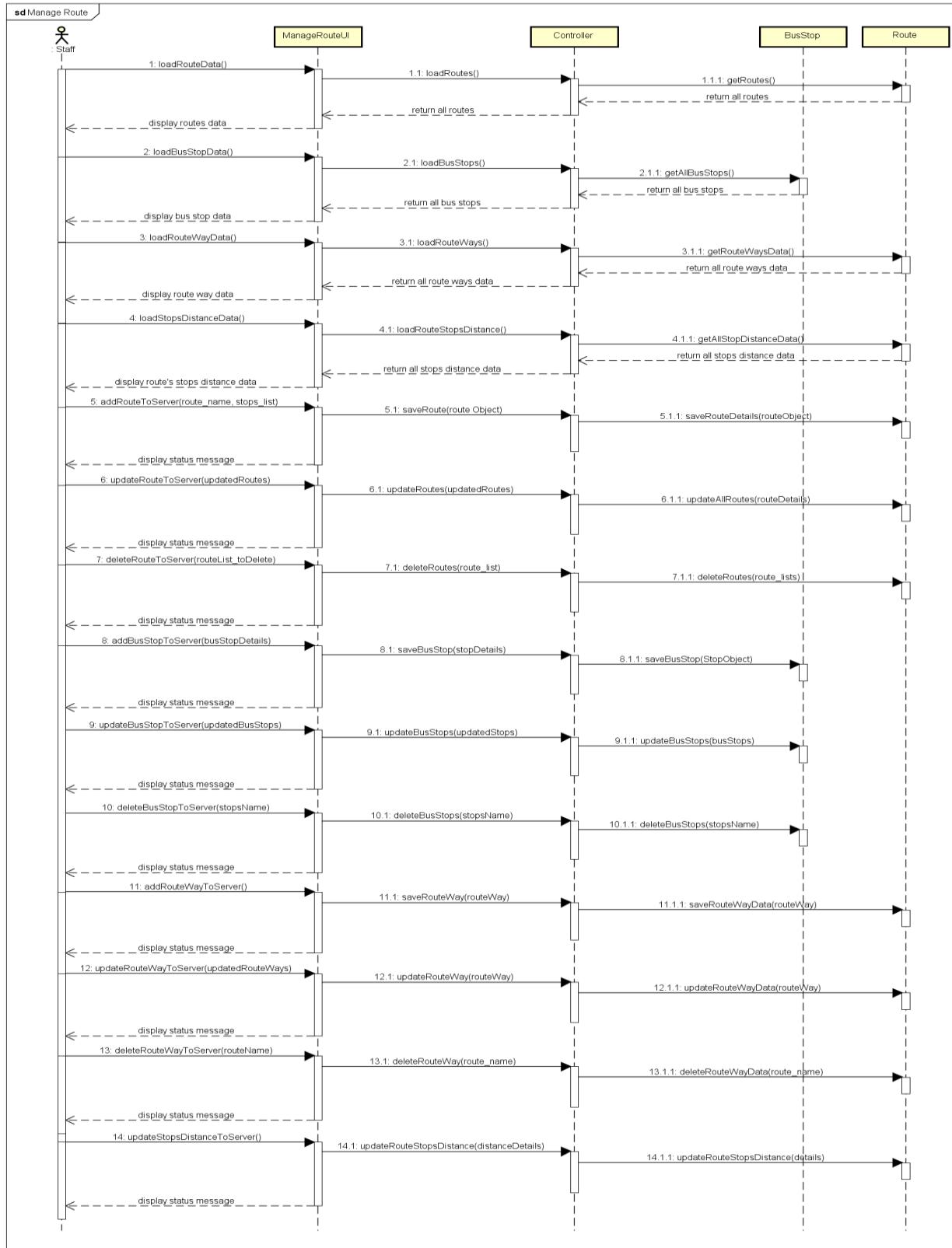
3.4.3.2.10. Manage Fare



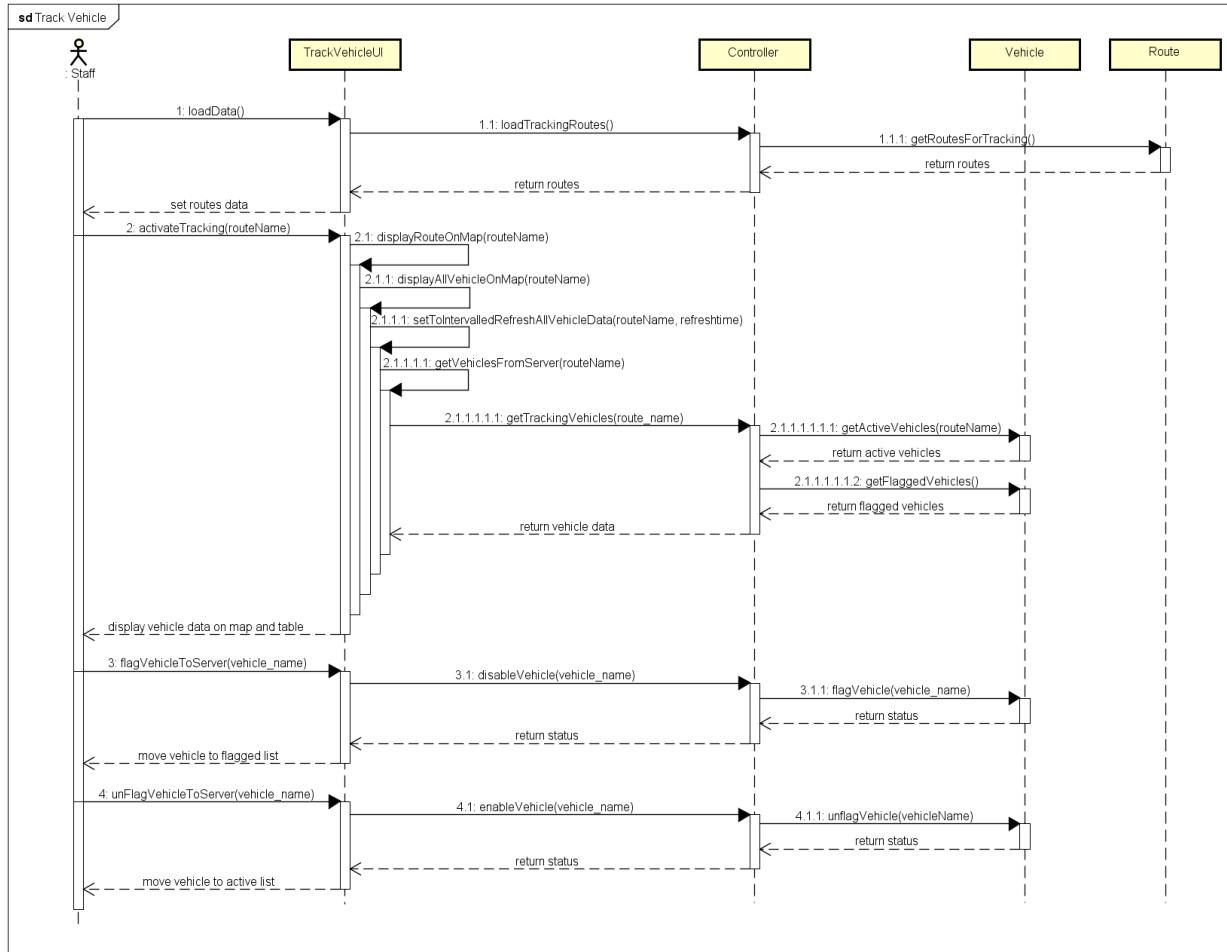
3.4.3.2.11. Manage Vehicle Information



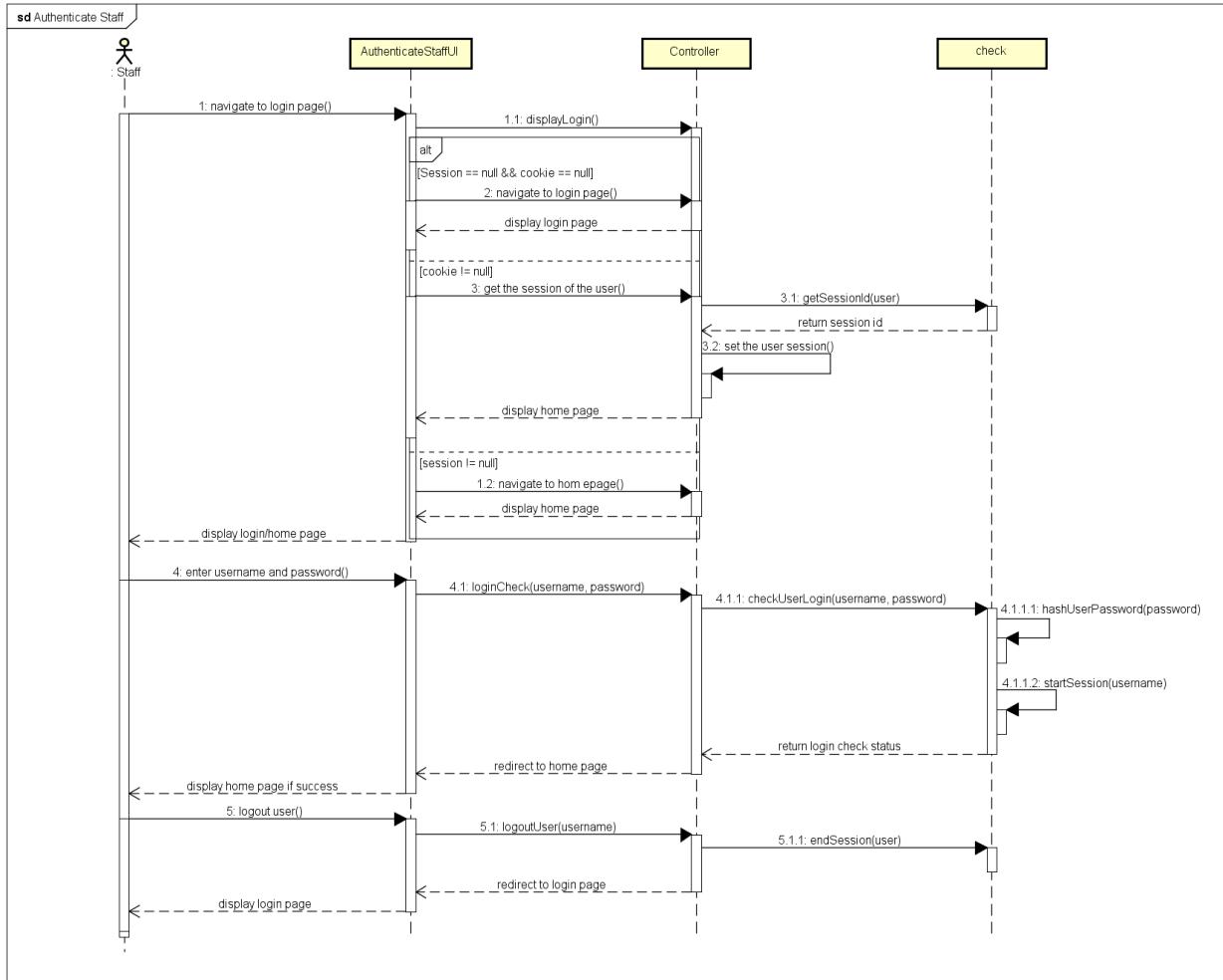
3.4.3.2.12. Manage Route Information



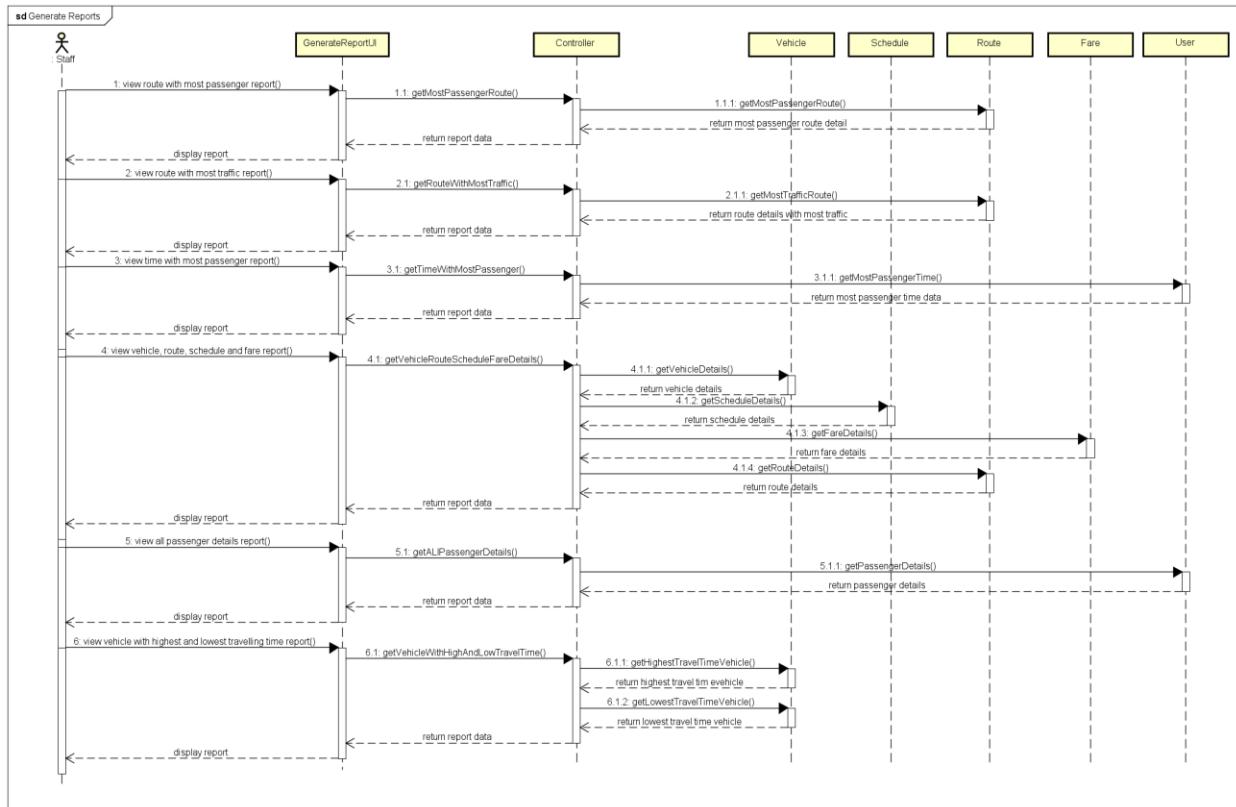
3.4.3.2.13. Track Vehicle



3.4.3.2.14. Authenticate Staff

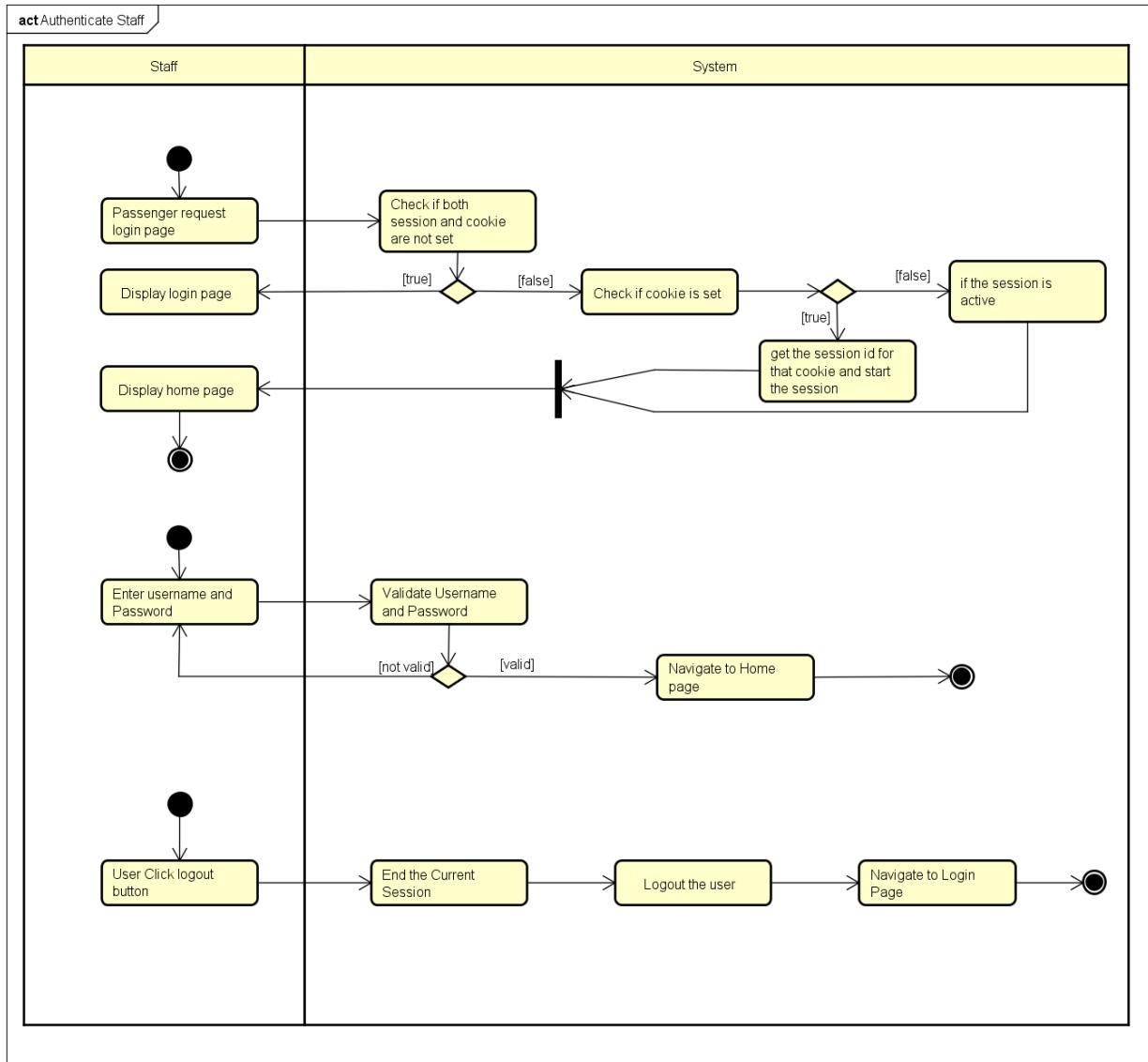


3.4.3.2.15. Generate Reports

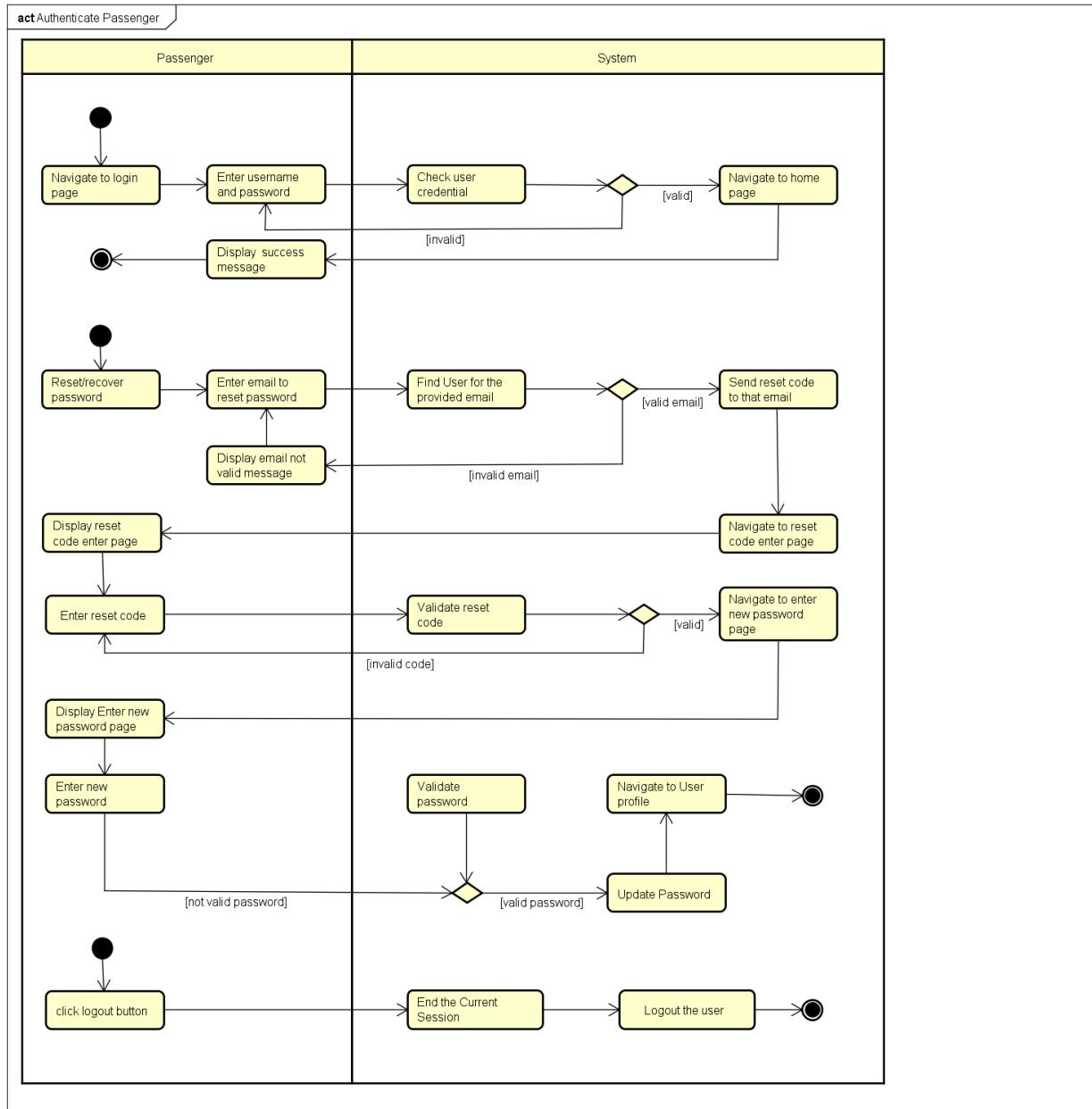


3.4.4. Activity Diagram

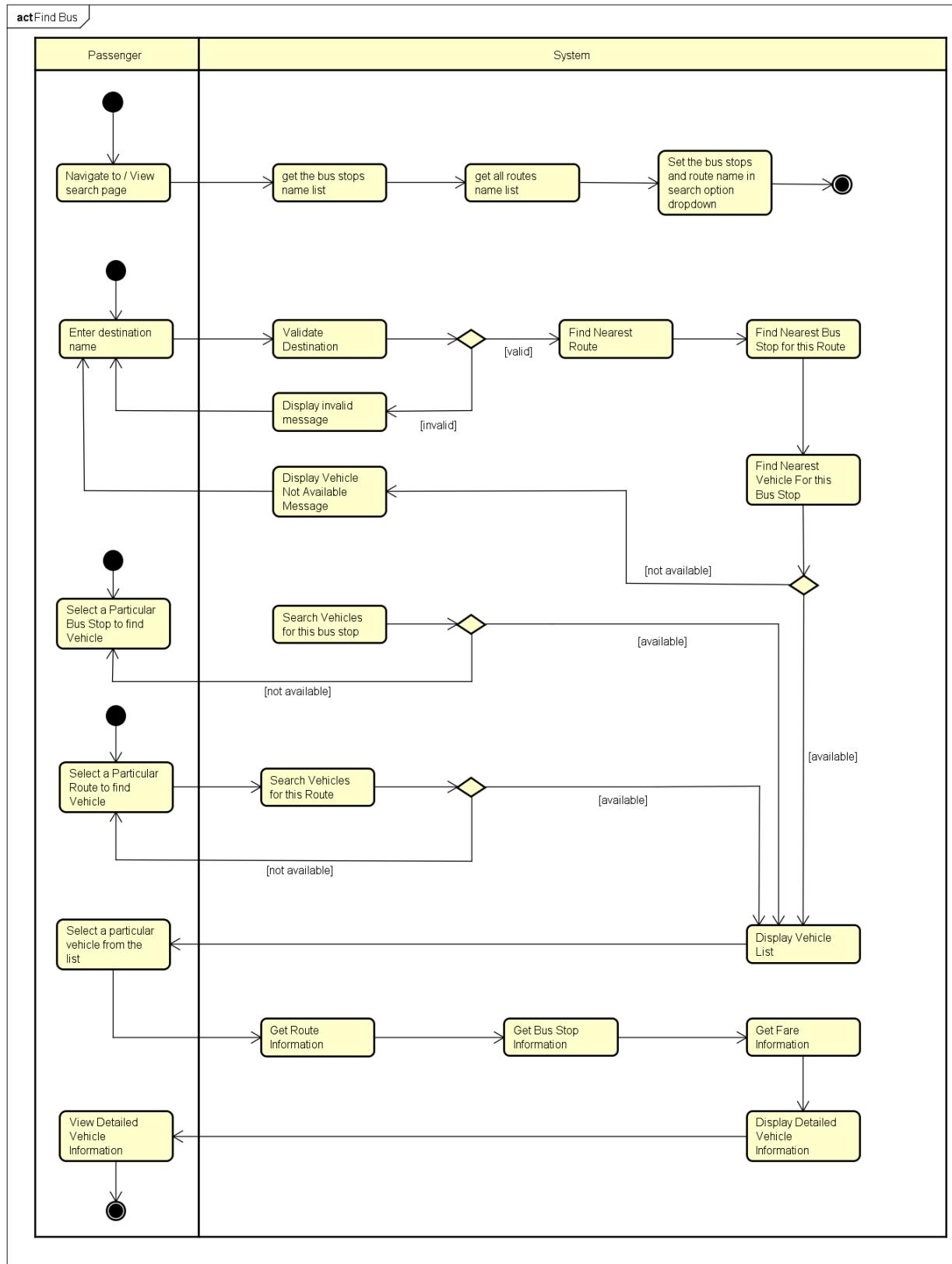
3.4.4.1. Authenticate Staff



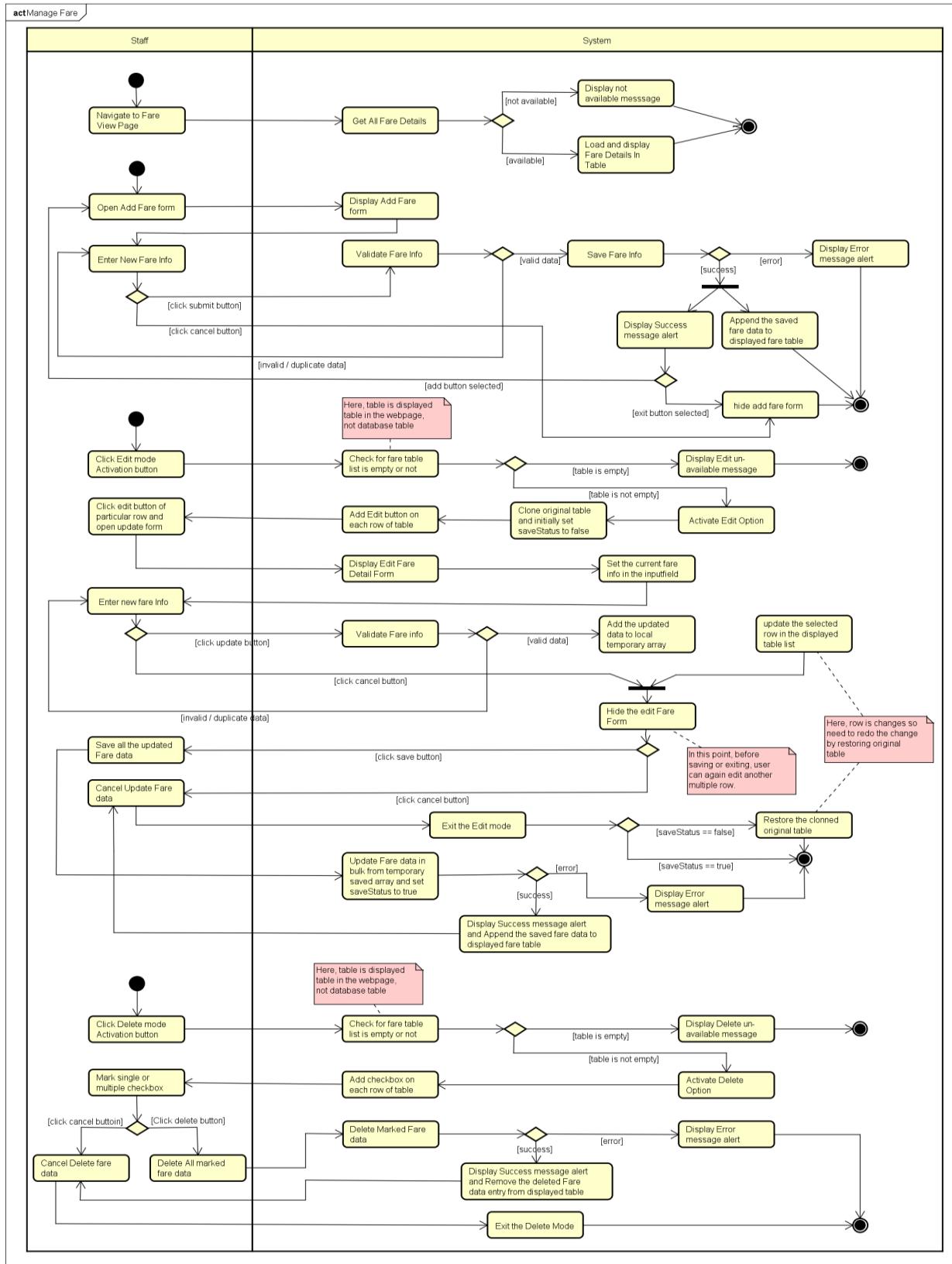
3.4.4.2. Authenticate Passenger



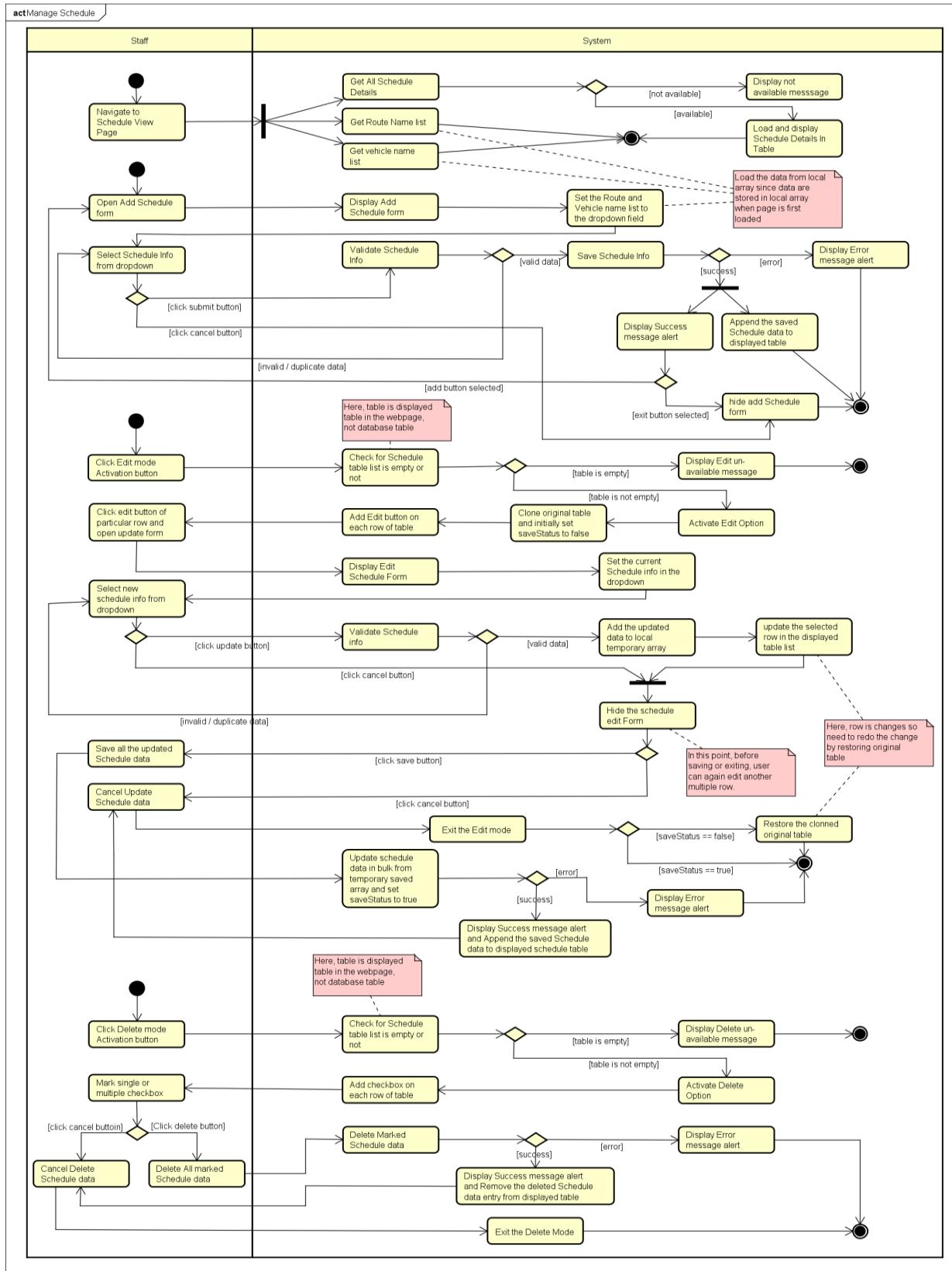
3.4.4.3. Find Bus



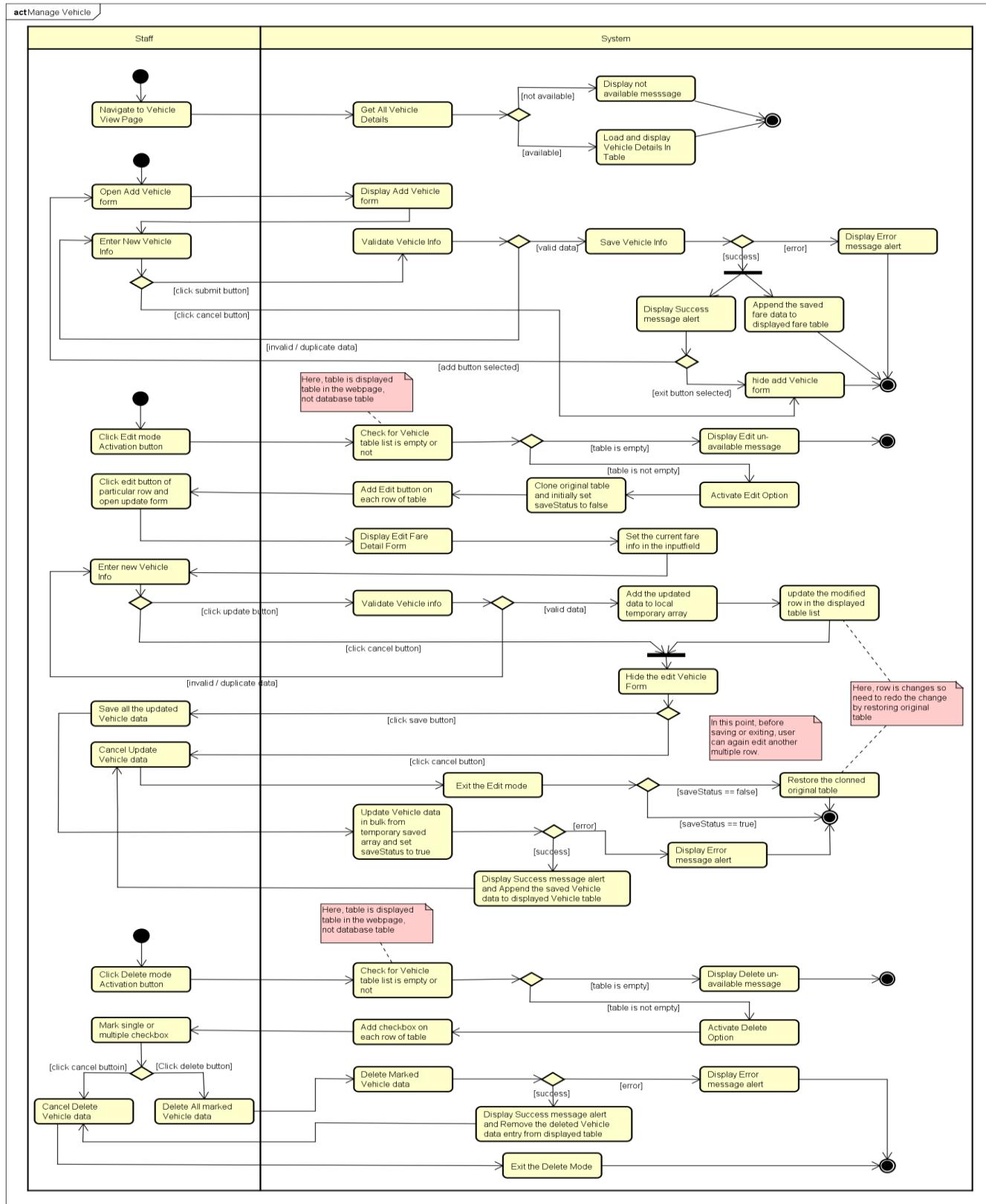
3.4.4.4. Manage Fare



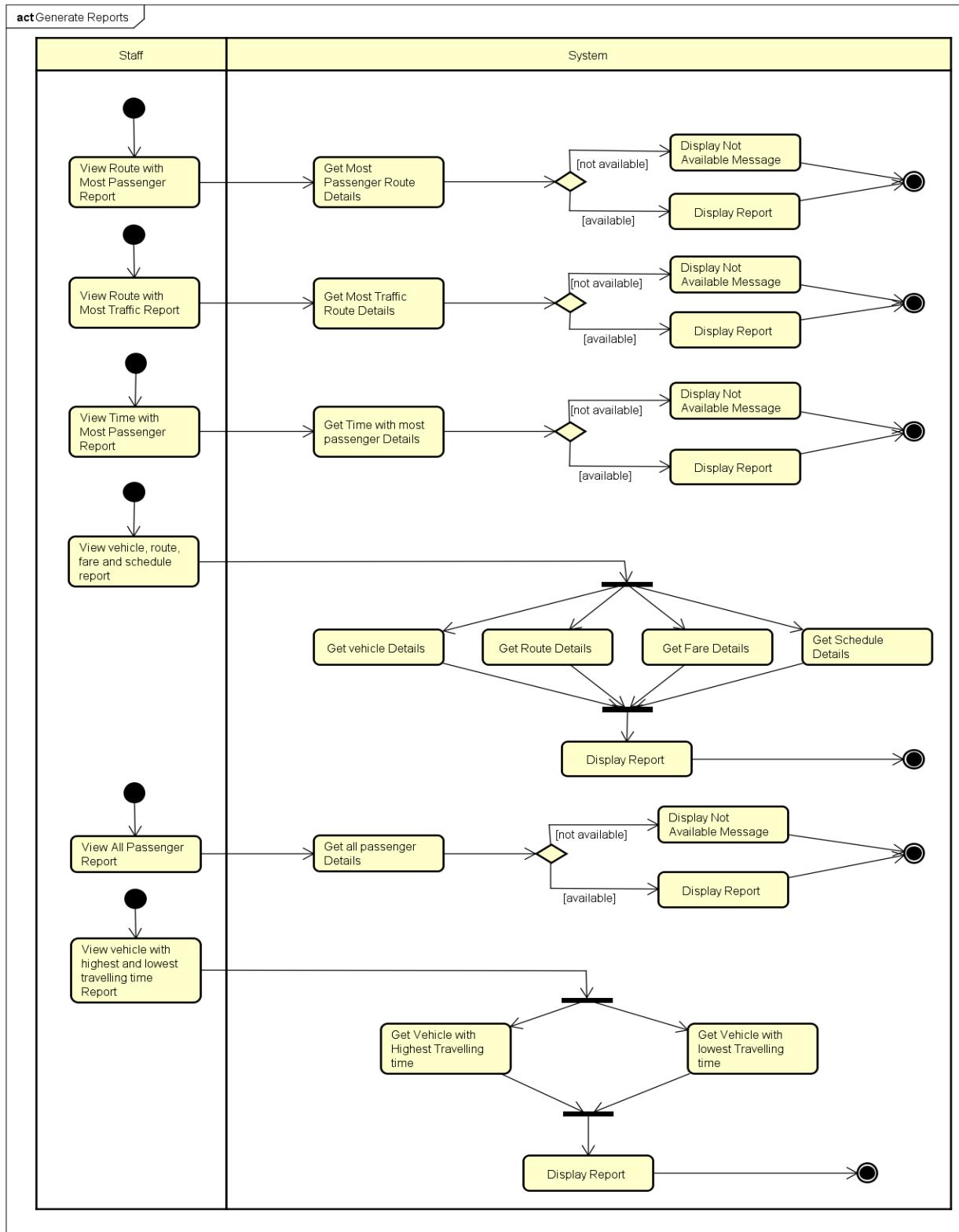
3.4.4.5. Manage Schedule



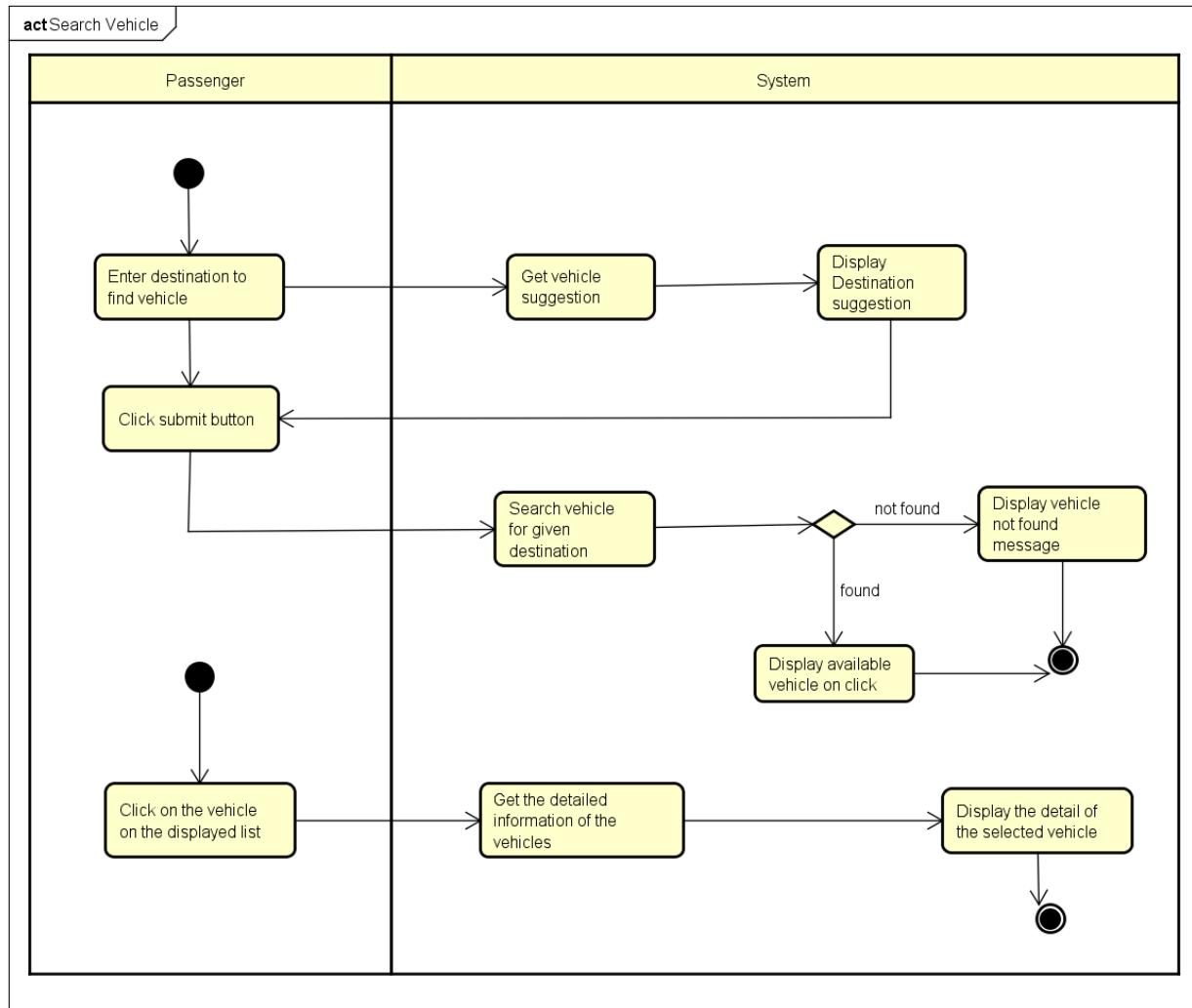
3.4.4.6. Manage Vehicles



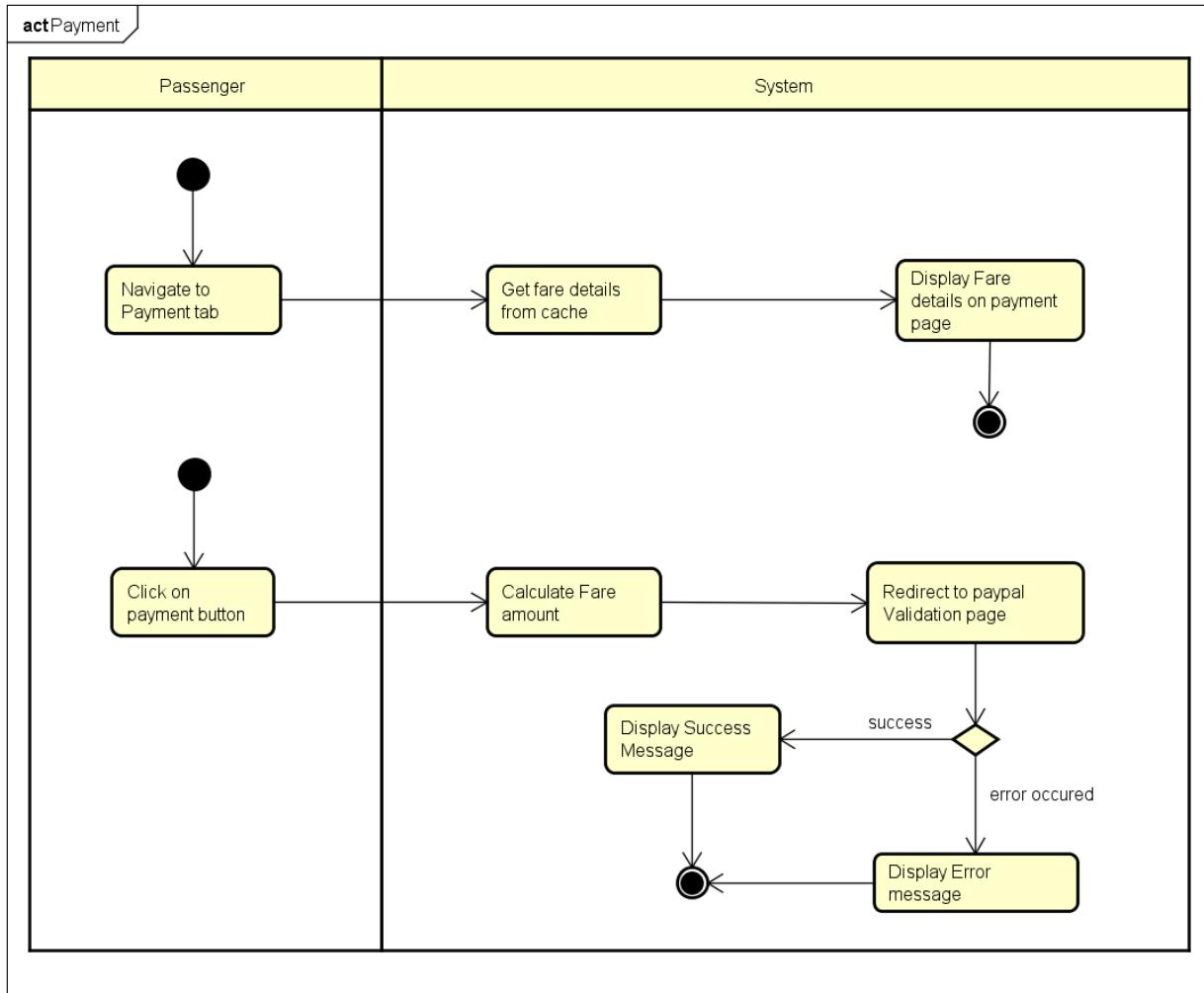
3.4.4.7. Generate Reports



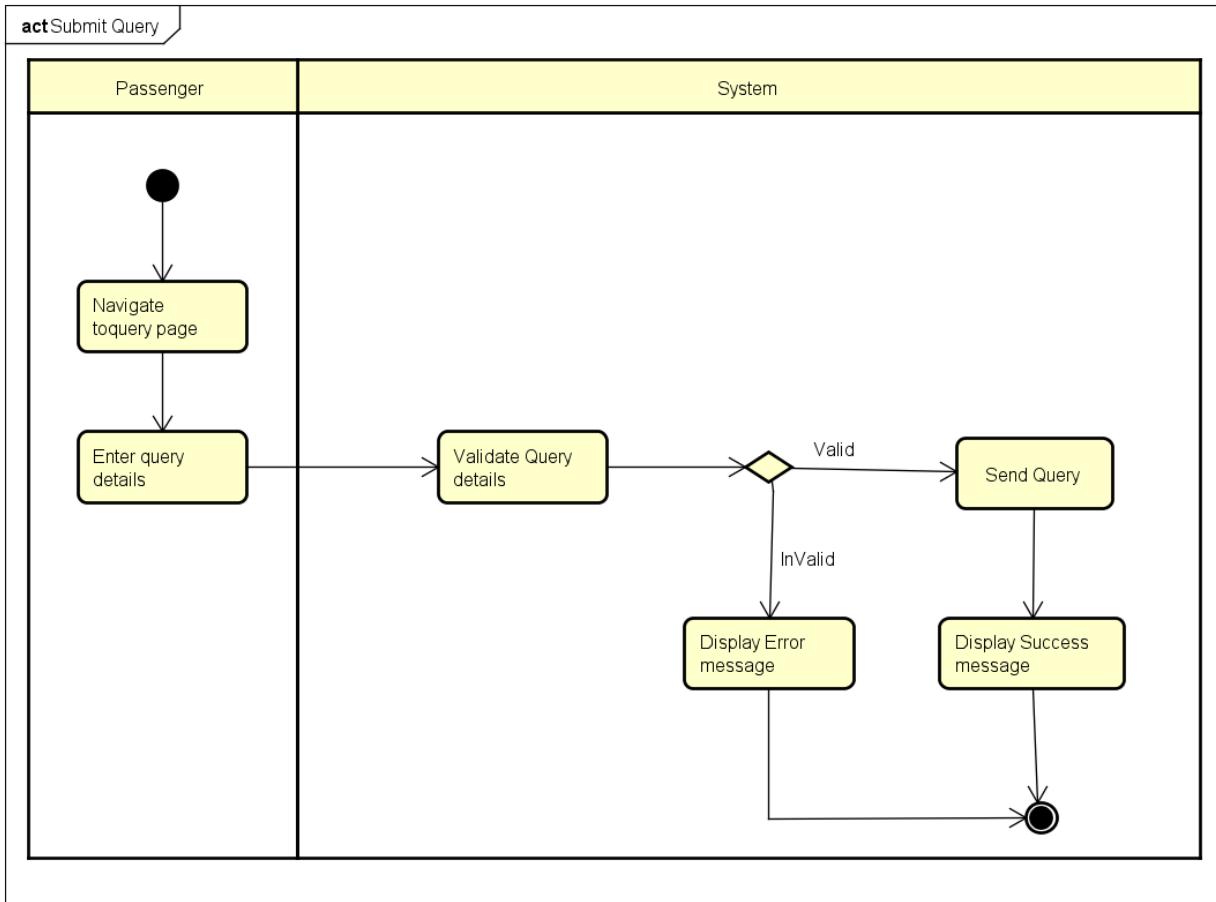
3.4.4.8. Search Vehicle



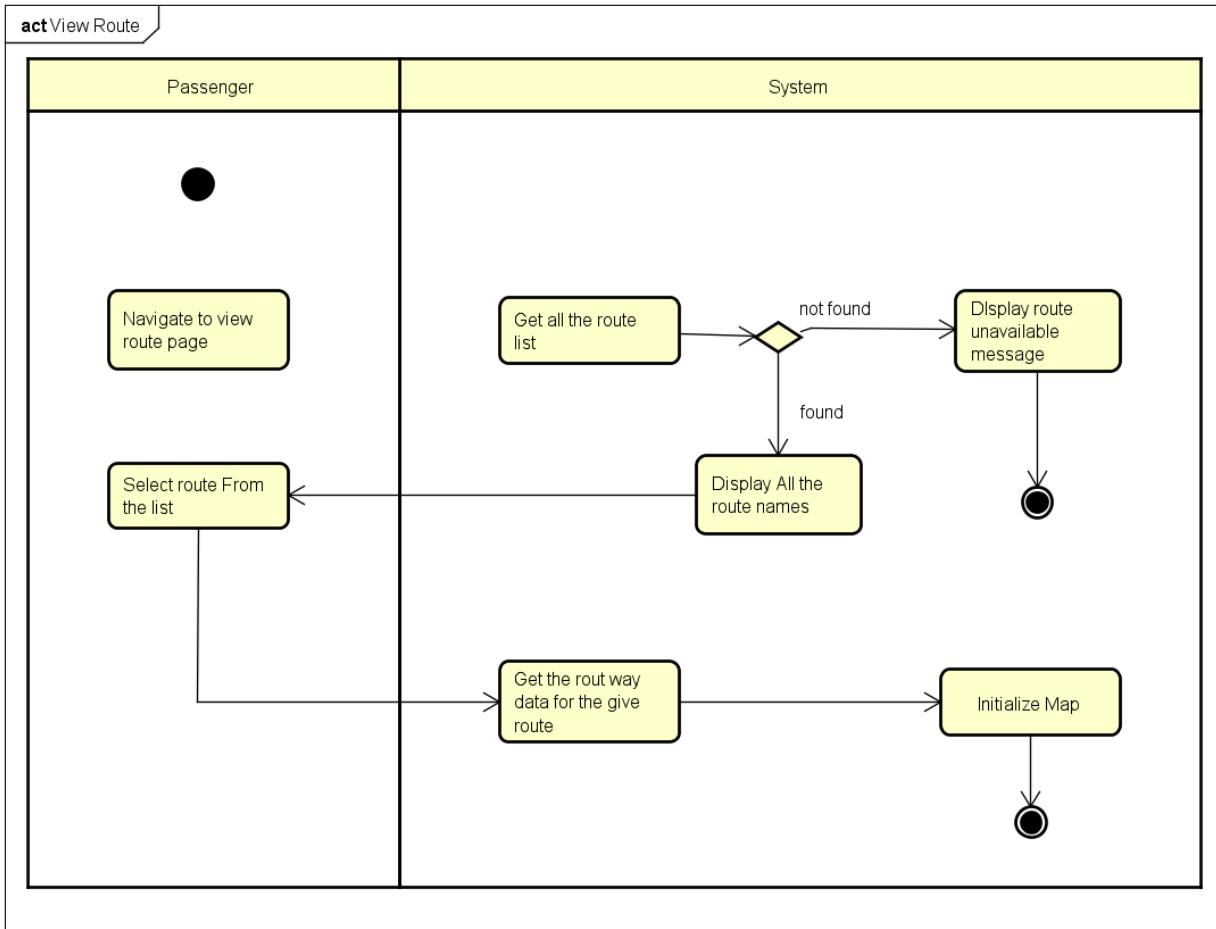
3.4.4.9. Make Payment



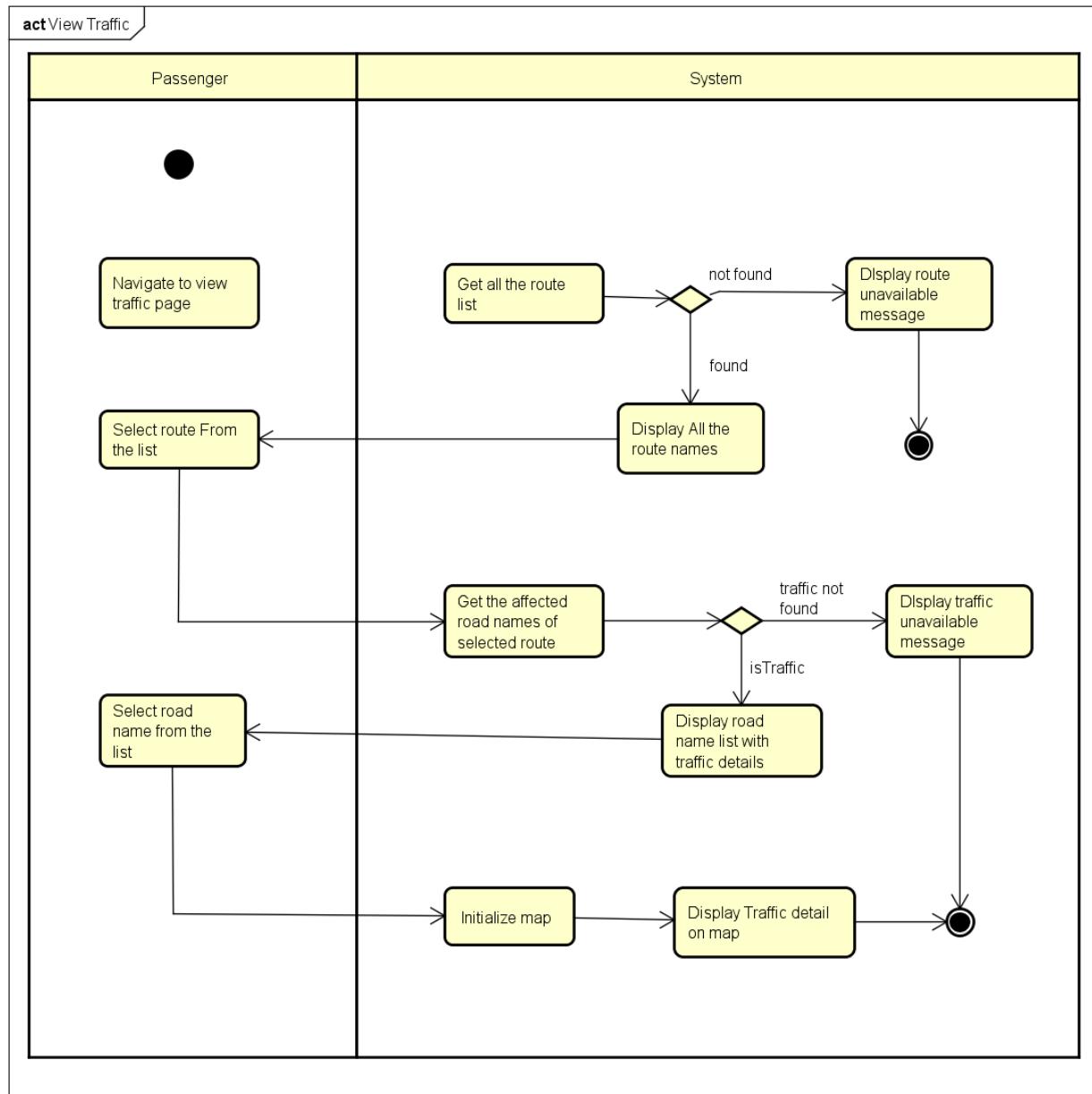
3.4.4.10. Query Submit



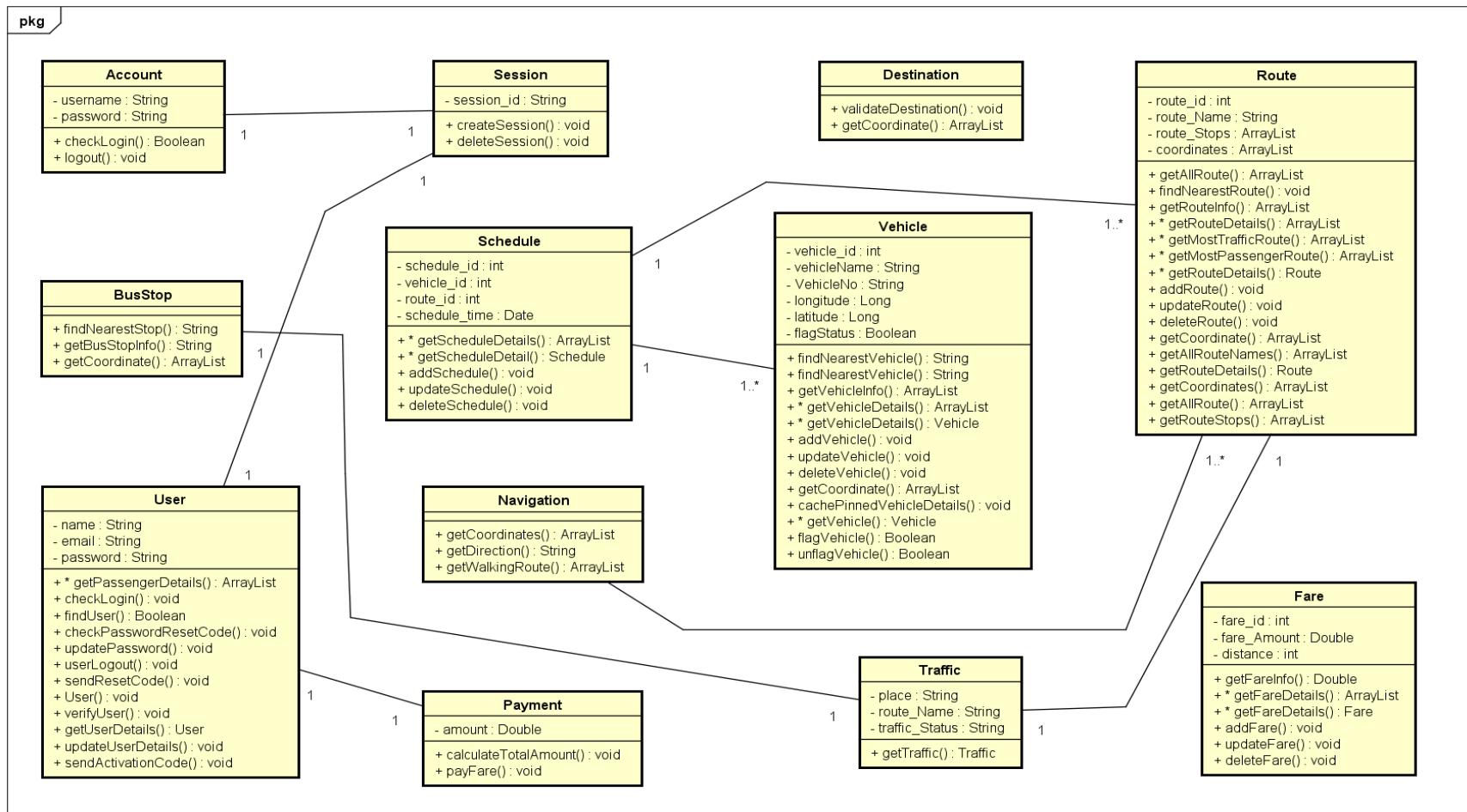
3.4.4.11. View Route



3.4.4.12. View Traffic



3.4.5. Class Diagram



3.4.6. Database Design

3.4.6.1. Business Rules

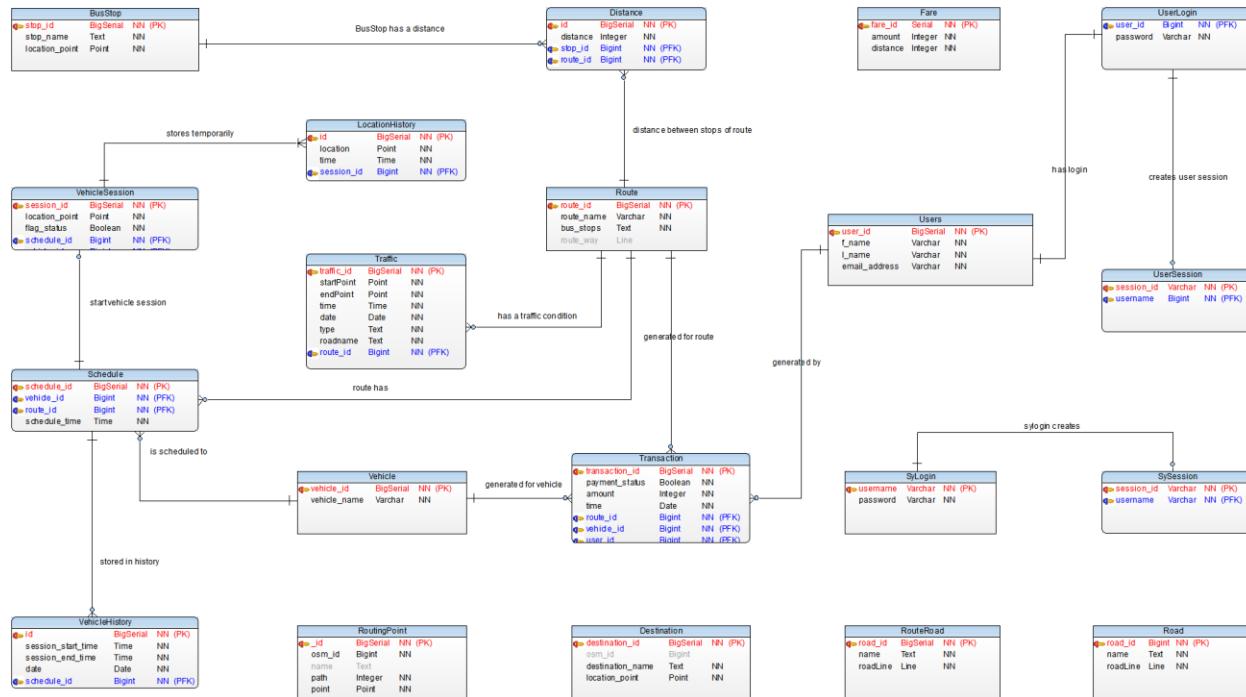
- A single Route can have more than 1 bus stop.
- A bus stop cannot be included in single route more than once.
- A single bus stop can be included in 1 or more routes.
- The user (staff/passenger) session will expire after 30 minutes if the user is inactive.
- The distance is used to calculate fare amount only not for routing.
- A Bus can have one or more than one schedule for single day.
- Only the active session bus which is not flagged can be tracked by the user.
- An active Bus can be flagged by staff user only, anytime and vice-versa.
- The Passengers cannot view the flagged bus and Staff user can view the flagged bus.
- The bus session will be expired after completing the route.
- The vehicle history data is removed after the bus active session is expired.
- In order to use payment option, user must register and login to the system.
- Only the registered user can share location to only another registered user and user can share location to 0 or many user or vice-versa.

3.4.6.2. Entities

TABLE NAME	Description
Traffic	This table holds all the information related to traffic such as traffic road coordinates, road name, time etc.
LocationHistory	This table holds the location history of the particular vehicle such as start time, end time etc. and this table is used for reporting mainly.
RouteRoad	This table holds all the route road of Kathmandu valley including coordinates and data will be imported from OSM data. This table is used to find current road name of the vehicle.

RoutingPoint	This table holds all the location point such as of Kathmandu valley from the OSM data. Used for routing.
VehicleSession	This table holds the vehicle session information such as vehicle current location etc.
VehicleHistory	This table holds the vehicle location history each time the vehicle location is updated from the vehicle. Used for traffic and estimated time analyzing.
Vehicle	This table holds vehicle information.
UserSession	This table holder user/passenger login session.
UserLogin	This table holds passenger login credential.
Users	This table holds personal information of passengers.
Transaction	This table holds payment information of user.
SySession	This table holds information of staff login session.
SyLogin	This table holds information of staff login credential.
Schedule	This table holds schedule information of vehicle.
Route	This table holds route information such as route name, route way data.
Road	This table holds various roads name and its co-ordinates.
Fare	This table holds fare information.
Distance	This table holds distance information between bus stops of a route.
Destination	This table holds destination name and co-ordinate and used to provide suggestion to user when searching vehicle for particular destination.
BusStop	This table holds table bus stop information including its location.

3.4.6.3. ER Diagram



3.4.6.4. Data Dictionary

3.4.6.4.1. TABLE NAME: TRAFFIC

Column	Type	Modifiers
Table "public.traffic"		
traffic_id	bigint	not null default nextval('traffic_traffic_id_seq'::regclass)
startpoint	geometry(Point,900913)	not null
endpoint	geometry(Point,900913)	not null
time	time without time zone	not null
date	date	not null
type	text	not null
roadname	text	not null
route_id	bigint	not null
Indexes:		
"key14" PRIMARY KEY, btree (traffic_id)		
Foreign-key constraints:		
" has traffic condition " FOREIGN KEY (route_id) REFERENCES route(route_id)		

3.4.6.4.2. TABLE NAME: LOCATIONHISTORY

Column	Type	Modifiers
Table "public.locationhistory"		
id	bigint	not null default nextval('locationhistory_id_seq'::regclass)
location	geometry(Point,900913)	not null
time	time without time zone	not null
session_id	bigint	not null
Indexes:		
"key13" PRIMARY KEY, btree (id)		

3.4.6.4.3. TABLE NAME: ROUTEROAD

Table "public.routeroad"		
Column	Type	Modifiers
road_id	bigint	not null default nextval('routeroad_road_id_seq'::regclass)
name	text	not null
roadline	geometry(LineString,4326)	not null
Indexes:		
	"key12"	PRIMARY KEY, btree (road_id)

3.4.6.4. TABLE NAME: ROUTINGPOINT

Table "public.routingpoint"		
Column	Type	Modifiers
_id	bigint	not null default nextval('routingpoint__id_seq'::regclass)
osm_id	bigint	not null
name	text	
path	integer	not null
point	geometry(Point,900913)	not null
isjunction	boolean	not null
Indexes:		
	"junctionid"	PRIMARY KEY, btree (_id)

3.4.6.5. TABLE NAME: VEHICLESESSION

Table "public.vehiclesession"		
Column	Type	Modifiers
session_id	bigint	not null default nextval('vehiclesession_session_id_seq'::regclass)
location_point	geometry(Point,900913)	not null
flag_status	boolean	not null
schedule_id	bigint	not null
Foreign-key constraints:		
	" start vehicle session "	FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id)

3.4.6.6. TABLE NAME: VEHICLEHISTORY

Column	Type	Modifiers
id	bigint	not null default nextval('vehiclehistory_id_seq'::regclass)
session_start_time	time without time zone	not null
session_end_time	time without time zone	not null
date	date	not null
schedule_id	bigint	not null

Indexes:

- "id" PRIMARY KEY, btree (id)

Foreign-key constraints:

- " stored in history " FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id)

3.4.6.7. TABLE NAME: VEHICLE

Column	Type	Modifiers
vehicle_id	bigint	not null default nextval('vehicle_vehicle_id_seq'::regclass)
vehicle_name	character varying	not null

Indexes:

- "key11" PRIMARY KEY, btree (vehicle_id)

Referenced by:

- TABLE "transaction" CONSTRAINT " generated for vehicle " FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)
- TABLE "schedule" CONSTRAINT " is scheduled to " FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)

3.4.6.8. TABLE NAME: USERSESSION

Column	Type	Modifiers
session_id	character varying	not null
username	bigint	not null

Indexes:

- "sessionid" PRIMARY KEY, btree (session_id)

Foreign-key constraints:

- " creates user session " FOREIGN KEY (username) REFERENCES userlogin(user_id)

3.4.6.9. TABLE NAME: USERLOGIN

```

Table "public.userlogin"
 Column |      Type       | Modifiers
-----+-----+
 user_id | bigint        | not null
 password | character varying | not null
Indexes:
    "key10" PRIMARY KEY, btree (user_id)
Foreign-key constraints:
    " has login " FOREIGN KEY (user_id) REFERENCES users(user_id)
Referenced by:
    TABLE "usersession" CONSTRAINT " creates user session " FOREIGN KEY (username) REFERENCES userlogin(user_id)

```

3.4.6.10. TABLE NAME: USERS

```

Table "public.users"
 Column |      Type       | Modifiers
-----+-----+
 user_id | bigint        | not null default nextval('users_user_id_seq'::regclass)
 f_name   | character varying | not null
 l_name   | character varying | not null
 email_address | character varying | not null
Indexes:
    "userid" PRIMARY KEY, btree (user_id)
Referenced by:
    TABLE "transaction" CONSTRAINT " generated by " FOREIGN KEY (user_id) REFERENCES users(user_id)
    TABLE "userlogin" CONSTRAINT " has login " FOREIGN KEY (user_id) REFERENCES users(user_id)

```

3.4.6.11. TABLE NAME: TRANSACTION

```

Table "public.transaction"
Column      | Type       | Modifiers
transaction_id | bigint     | not null default nextval('transaction_transaction_id_seq'::regclass)
payment_status | boolean    | not null
amount         | integer    | not null
time           | date       | not null
route_id       | bigint     | not null
vehicle_id     | bigint     | not null
user_id        | bigint     | not null
Indexes:
  "key9" PRIMARY KEY, btree (transaction_id)
Foreign-key constraints:
  " generated by " FOREIGN KEY (user_id) REFERENCES users(user_id)
  " generated for route " FOREIGN KEY (route_id) REFERENCES route(route_id)
  " generated for vehicle " FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)

```

3.4.6.12. TABLE NAME: SYSESSION

```

Table "public.sysession"
Column      | Type       | Modifiers
session_id | character varying | not null
username   | character varying | not null
Indexes:
  "key8" PRIMARY KEY, btree (session_id)
Foreign-key constraints:
  "sylogin creates " FOREIGN KEY (username) REFERENCES sylogin(username)

```

3.4.6.13. TABLE NAME: SYLOGIN

```

Table "public.sylogin"
Column      | Type       | Modifiers
username   | character varying | not null
password   | character varying | not null
Indexes:
  "username" PRIMARY KEY, btree (username)
Referenced by:
  TABLE "sysession" CONSTRAINT " sylogin creates " FOREIGN KEY (username) REFERENCES sylogin(username)

```

3.4.6.14. TABLE NAME: SCHEDULE

Column	Type	Table "public.schedule"	Modifiers
schedule_id	bigint	not null default nextval('schedule_schedule_id_seq'::regclass)	
vehicle_id	bigint	not null	
route_id	bigint	not null	
schedule_time	time without time zone	not null	
Indexes:			
"scheduleid" PRIMARY KEY, btree (schedule_id)			
Foreign-key constraints:			
" is scheduled to " FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id)			
" route has " FOREIGN KEY (route_id) REFERENCES route(route_id)			
Referenced by:			
TABLE "vehiclesession" CONSTRAINT " start vehicle session " FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id)			
TABLE "vehiclehistory" CONSTRAINT " stored in history " FOREIGN KEY (schedule_id) REFERENCES schedule(schedule_id)			

3.4.6.15. TABLE NAME: ROUTE

Column	Type	Table "public.route"	Modifiers
route_id	bigint	not null default nextval('route_route_id_seq'::regclass)	
route_name	character varying	not null	
bus_stops	text	not null	
route_way	geometry(LineString,900913)		
Indexes:			
"routeno" PRIMARY KEY, btree (route_id)			
Referenced by:			
TABLE "distance" CONSTRAINT " distance between stops of route " FOREIGN KEY (route_id) REFERENCES route(route_id)			
TABLE "transaction" CONSTRAINT " generated for route " FOREIGN KEY (route_id) REFERENCES route(route_id)			
TABLE "traffic" CONSTRAINT " has traffic condition " FOREIGN KEY (route_id) REFERENCES route(route_id)			
TABLE "schedule" CONSTRAINT " route has " FOREIGN KEY (route_id) REFERENCES route(route_id)			

3.4.6.16. TABLE NAME: ROAD

Column	Type	Table "public.road"	Modifiers
road_id	bigint	not null	
name	text	not null	
roadline	geometry(LineString,4326)	not null	
Indexes:			
"roadid" PRIMARY KEY, btree (road_id)			

3.4.6.17. TABLE NAME: FARE

```

Table "public.fare"
Column | Type | Modifiers
-----+-----+-----
fare_id | integer | not null default nextval('fare_fare_id_seq'::regclass)
amount | integer | not null
distance | integer | not null
Indexes:
"fareid" PRIMARY KEY, btree (fare_id)

```

3.4.6.18. TABLE NAME: DISTANCE

```

Table "public.distance"
Column | Type | Modifiers
-----+-----+-----
id | bigint | not null default nextval('distance_id_seq'::regclass)
distance | integer | not null
stop_id | bigint | not null
route_id | bigint | not null
Indexes:
"distanceid" PRIMARY KEY, btree (id)
Foreign-key constraints:
" BusStop has a distance " FOREIGN KEY (stop_id) REFERENCES busstop(stop_id)
" distance between stops of route " FOREIGN KEY (route_id) REFERENCES route(route_id)

```

3.4.6.19. TABLE NAME: DESTINATION

```

Table "public.destination"
Column | Type | Modifiers
-----+-----+-----
destination_id | bigint | not null default nextval('destination_destination_id_seq'::regclass)
osm_id | bigint | default nextval('destination_osm_id_seq'::regclass)
destination_name | text | not null
location_point | geometry(Point,900913) | not null
Indexes:
"destinationid" PRIMARY KEY, btree (destination_id)
"destination_index" btree (destination_name)

```

3.4.6.20. TABLE NAME: BUSSTOP

Column	Type	Modifiers
stop_id	bigint	not null default nextval('busstop_stop_id_seq'::regclass)
stop_name	text	not null
location_point	geometry(Point,900913)	not null
Indexes:		
"stopid" PRIMARY KEY, btree (stop_id)		
Referenced by:		
TABLE "distance" CONSTRAINT " BusStop has a distance " FOREIGN KEY (stop_id) REFERENCES busstop(stop_id)		

3.5. Development Phase

Following are the list of the some of the important algorithms (Java and SQL) of the system to perform major task of the system such as vehicle tracking, route finding, traffic finding etc.

3.5.1. Finding Vehicle for Particular Route

The following code find vehicle for particular route. First, route name and bus stop name is passed as parameter and vehicle with next bus stop to give busstop is displayed. This code is iteratively used in business class to get vehicle for multiple route.

WITH

```
temp_routeName As (SELECT 'Route 1'::text As name),
temp_busstopName As (SELECT 'Tripureshwor1 Bus Station'::text As name),
temp_route_way As (SELECT ST_line_substring(route_way, ST_line_locate_point(route_way,
ST_StartPoint(route_way)), st_line_locate_point(route_way, (SELECT location_point FROM
busstop WHERE stop_name=(SELECT name FROM temp_busstopName)))) As line FROM
route WHERE route_name=(SELECT name FROM temp_routeName))
```

```
SELECT ST_X(ST_Transform(location_point, 4326)), ST_Y(ST_Transform(location_point,
4326)), (SELECT vehicle_name FROM vehicle WHERE vehicle_id=(SELECT vehicle_id
FROM schedule WHERE schedule_id=vehiclesession.schedule_id)) As vehicleName, session_id
FROM vehiclesession WHERE flag_status IS false AND schedule_id = ANY(SELECT
schedule_id FROM schedule WHERE route_id=(SELECT route_id FROM route WHERE
route_name=(SELECT name FROM temp_routeName))) AND
ST_Dwithin(Geography(ST_Transform(location_point, 4326)),
Geography(ST_Transform(ST_ClosestPoint((SELECT line FROM
temp_route_way),location_point), 4326)), 5) IS TRUE;
```

Result:

st_x	st_y	vehiclename	session_id
85.299757	27.698543	BA 11 PA 1111	56
(1 row)			

3.5.2. Finding Route near user destination

The code below finds nearest route and busstop to user destination. It requires destination coordinate as parameter (calculate in business logic) and then return all the nearest bus stop on each route.

```

WITH destination As (SELECT ST_SetSRID(ST_MakePoint( 85.3166801684716,
27.6992704632346 ) /* destination location */, 4326)),

/*Finding all the available nearest route for given destination in given range.*/

routeName As (SELECT name FROM (SELECT ST_ClosestPoint(route_way,
ST_Transform((SELECT * FROM destination), 900913)) As way, route_name As name FROM
ROUTE WHERE route_way IS NOT NULL) As x WHERE ST_Distance(ST_Transform(way,
4326)::geography, (SELECT * FROM destination)) <= (5000+1000))

SELECT

/*Calculate the distance between destination and busstop */

routeName.name As route_name1, ST_Distance((SELECT * FROM destination),
ST_Transform(busstop.location_point, 4326)::geography) As distance1, busstop.stop_name As
stop_name

FROM

busstop, routeName,(SELECT routeName.name As route_name2, MIN(ST_Distance((SELECT *
FROM destination), ST_Transform(busstop.location_point, 4326)::geography)) As distance2
FROM busstop, routeName WHERE busstop.stop_name = ANY(string_to_array((SELECT

```

```
bus_stops FROM route WHERE route_name= routeName.name), ',') GROUP BY
routeName.name HAVING MIN(ST_Distance((SELECT * FROM destination),
ST_Transform(busstop.location_point, 4326)::geography)) < 1000 /* range */) As z
```

WHERE

```
routeName.name = route_name2 AND ST_Distance((SELECT * FROM destination),
ST_Transform(busstop.location_point, 4326)::geography) = distance2;
```

Result:

route_name1	distance1	stop_name
Route 2	736.194261961	Tripureshwor1 Bus Station
Route 5	398.796293044	RNAC Bus Station
Route 1	398.796293044	RNAC Bus Station
(3 rows)		

3.5.3. Auto correcting vehicle location on given route

The code below, auto correct vehicle current position to the route's way near point.

```
SELECT ST_X(ST_Transform(way, 4326)), ST_Y(ST_Transform(way, 4326)) FROM (SELECT
ST_ClosestPoint(route_way, ST_transform(ST_setSRID(ST_makePoint( 85.310379, 27.694757
), 4326), 900913)) As way FROM ROUTE WHERE route_name = 'Route 1') As z;
```

Result:

st_x	st_y
85.3104651532904	27.6949423862285
(1 row)	

3.5.4. Finding Estimated Time of Arrival

The below code find estimated arrival time of particular vehicle to particular bus stop in a given route.

WITH

```
/*User input data*/
```

```
temp_routeName As (SELECT 'Route 1'::text As name),
```

```
temp_vehicleName As (SELECT 'BA 12 PA 2222'::text As name),
```

```
temp_stopName As (SELECT 'Tripureshwor1 Bus Station'::text As name),
```

```
time_range As (SELECT 5::integer As time),
```

```
/* Getting schedule for provided vehicle and route*/
```

```
temp_scheduleId As (SELECT schedule_id FROM schedule WHERE route_id=(SELECT route_id FROM route WHERE route_name=(SELECT name FROM temp_routeName)) AND vehicle_id=(SELECT vehicle_id FROM vehicle WHERE vehicle_name=(SELECT name FROM temp_vehicleName))),
```

```
/*Getting user specified rows (5 mins i.e. 5*4 20 rows since vehicle is updated every 15 secs. so 1 min = 4 row) of vehicle history*/
```

```
temp_locationHistory As (SELECT location, time FROM locationhistory WHERE session_id=(SELECT session_id FROM vehiclesession WHERE schedule_id=(SELECT * FROM temp_scheduleId)) ORDER BY time DESC LIMIT ((SELECT * FROM time_range)*4 /*One minute contain 4 row*/)),
```

```
/*Getting last and 20th last or first points*/
```

```
temp_firstPoint As (SELECT location FROM temp_locationHistory ORDER BY TIME ASC
LIMIT 1),
```

```
temp_lastPoint As (SELECT location FROM temp_locationHistory ORDER BY TIME DESC
LIMIT 1),
```

/*Getting a linestring between user bus stop and vehicle location*/

```
temp_vehicle_user_way           As           (SELECT           st_line_substring(route_way,
st_line_locate_point(route_way, (SELECT location_point FROM vehiclesession WHERE
schedule_id=(SELECT * FROM temp_scheduleId))), st_line_locate_point(route_way, (SELECT
location_point FROM busstop WHERE stop_name=(SELECT name FROM temp_stopName))))
FROM route WHERE route_name=(SELECT name FROM temp_routeName)),
```

/*Calculating total no. of bus stops between above line string*/

```
temp_bus_stop_count As (SELECT COUNT(*) FROM busstop WHERE stop_name =
ANY(string_to_array((SELECT bus_stops FROM route WHERE route_name= (SELECT name
FROM      temp_routeName)), ','))      AND      st_dwithin((st_transform(location_point,
4326)::Geography,          (ST_Transform(ST_ClosestPoint((SELECT      *
FROM
temp_vehicle_user_way), location_point), 4326)::Geography, 20)),
```

/* Calculating vehicle to user busstop distance*/

```
user_stop_to_vehicle_distance As (SELECT ST_Length((ST_Transform( /*Substring between
vehicle location and user bus stop.*/ (SELECT * FROM temp_vehicle_user_way)
,4326)::Geography)),
```

/*Calculating speed, note that, average 1.5 min time waste has been added on each bus stop. Speed
= Distance / Time */

/*Calculating the substring from route way and finding its length and then finding the speed by dividing 5 mins i.e. location history range 20 row*/

```
past_vehicle_speed      As      (SELECT      ((ST_Length(ST_line_substring(route_way,
ST_line_locate_point(route_way,      (SELECT      *      FROM      temp_firstPoint)),
ST_line_locate_point(route_way, (SELECT * FROM temp_lastPoint)))))/(( (SELECT * FROM
time_range) + (SELECT COUNT(*)*1.5 /*Adding 1.5 min extra as bus stop time waste on each
bus stop*/ FROM temp_bus_stop_count )/60::float /*Dividing by 60 to convert 5 mins() to
hour*/))/1000 /* Divided by 1000 to convert from meter to km.*/ FROM route WHERE
route_name=(SELECT * FROM temp_routeName))
```

/*Calculating Time from vehicle to bus stop and current road name of vehicle. Time = Distance/Speed */

```
SELECT ROUND(((SELECT * FROM user_stop_to_vehicle_distance)/1000 /*Converting into
km by multiplying by 1000* / (SELECT * FROM past_vehicle_speed)*60) /* Converting into
minute from hour */ As ETA, roadName, (SELECT * FROM temp_bus_stop_count) As
stops_between FROM (SELECT name As roadName FROM routeRoad WHERE
ST_Dwithin(ST_Transform(roadline,                                4326)::Geography,
ST_Transform(ST_ClosestPoint((SELECT route_way FROM ROUTE WHERE
route_name='Route 1'), (SELECT location_point FROM vehiclesession WHERE
schedule_id=(SELECT * FROM temp_scheduleId))), 4326)::Geography,1) LIMIT 1) As
current_vehicle_name;
```

Result:

eta	roadname	stops_between
4	Ganeshman Singh Path	2

(1 row)

3.5.5. Finding Traffic

The following code check if the given vehicle is in traffic condition or not.

WITH

```
/*User input data*/  
  
temp_routeName As (SELECT 'Route 1'::text As name),  
  
temp_vehicleName As (SELECT 'BA 12 PA 2222'::text As name),  
  
time_range As (SELECT 4::integer As time),  
  
first_speed_range As (SELECT 4.6::float As value),  
  
last_speed_range As (SELECT 7.5::float As value),  
  
traffic_check_type As (SELECT 'Low Traffic'::varchar As value),  
  
  
  
/* Getting schedule for provided vehicle and route*/  
  
temp_scheduleId As (SELECT schedule_id FROM schedule WHERE route_id=(SELECT  
route_id FROM route WHERE route_name=(SELECT name FROM temp_routeName)) AND  
vehicle_id=(SELECT vehicle_id FROM vehicle WHERE vehicle_name=(SELECT name FROM  
temp_vehicleName))),  
  
  
  
/*All the location history for given time range*/  
  
temp_locationHistory As (SELECT location, time FROM locationhistory WHERE  
session_id=(SELECT session_id FROM vehiclesession WHERE schedule_id=(SELECT *  
FROM temp_scheduleId)) ORDER BY time DESC LIMIT ((SELECT * FROM time_range)*4  
/*One minute contain 4 row*/)),
```

```
/*Getting last 16 rows from the above calculated locaitonhistory e.g. if user submit 8 min timerange, then this only get for last 5-8 min history exclude first 1-4 since it is already calculated when user input 4 timerange (note. time range is increase by every 4 mins.).*/
```

```
last_4_min_location_history As (SELECT location, time FROM (SELECT location, time FROM temp_locationHistory ORDER BY time ASC LIMIT 16) As history_location /*limiting last, 4 minutes rows from the calculated time range location history table above*/),
```

```
temp_firstPoint As (SELECT location FROM last_4_min_location_history LIMIT 1),
```

```
/*Since the last_4_min_location_history is in DESC format, the last one is first point and first one is last point.*/
```

```
temp_lastPoint As (SELECT location, time FROM last_4_min_location_history ORDER BY TIME DESC LIMIT 1),
```

```
/*Calculating sub string of traffic road*/
```

```
temp_traffic_road As (SELECT ST_line_substring(route_way, ST_line_locate_point(route_way, (SELECT location FROM temp_firstPoint)), ST_line_locate_point(route_way, (SELECT location FROM temp_lastPoint))) FROM route WHERE route_name=(SELECT * FROM temp_routeName)),
```

```
/*Calculating length travelled from route way for given vehicle in given route for last given time. e.g. 5 mins.*/
```

```
vehicle_distance_travelled As (SELECT (ST_Length((SELECT * FROM temp_traffic_road))) As distance),
```

/*Calculating speed FROM above distance and user provided time range and then checking if the speed between given speed range or not*/

```
is_traffic_situation      As      (SELECT      (((SELECT      distance      FROM
vehicle_distance_travelled)/1000::float/*Km*))/((SELECT      time      FROM
time_range)/60::float/*Convert to hour*/) BETWEEN (SELECT value FROM first_speed_range)
AND (SELECT value FROM last_speed_range) AND
```

/*Ensuring that the traffic situation is being calculated not for same point repeated in loop e.g. if this condition is not specified, the traffic result will be send true on same data until given time loop finished. i.e. if there is only 20 entry on location history, but the time is 10 mins, then data is differnt until first 5mins (5*4=20row) and then after that on each minute the same last row is calculated which is unnecessary, and this condition help to exit out of that loop. Also, this helps to avoid traffic update until first 4 mins of vehicle start.*/

```
(SELECT COUNT(*) FROM temp_locationHistory)>=((SELECT * FROM time_range)*4)) As
is_traffic),
```

/*Checking if any previous traffic record in traffic table on same road/line*/

```
previous_traffic_check As (SELECT COUNT(*)=1 As is_previous_traffic_present FROM traffic
/*Checking the date is of today*/ WHERE traffic.date = current_date AND
traffic.route_id=(SELECT route_id FROM route WHERE route_name=(SELECT * FROM
temp_routeName)) AND /*Checking if both are same line string*/ST_Intersects(/*Traffic table
line string*/ ST_line_substring((SELECT route_way FROM route WHERE
route_name=(SELECT * FROM temp_routeName)), ST_line_locate_point((SELECT route_way
FROM route WHERE route_name=(SELECT * FROM temp_routeName)), traffic.startPoint),
ST_line_locate_point((SELECT route_way FROM route WHERE route_name=(SELECT *
FROM temp_routeName)), traffic.endPoint))), /*The result line string*/ SELECT * FROM
temp_traffic_road)))
```

```
SELECT  is_traffic,  is_previous_traffic_present,  (SELECT  ST_AsText(location)  FROM temp_firstPoint) As first,(SELECT ST_AsText(location) FROM temp_lastPoint) As last,
```

CASE

WHEN is_traffic = true AND is_previous_traffic_present = true THEN

'Update'

WHEN is_traffic = true AND is_previous_traffic_present = false THEN

'Insert'

WHEN is_traffic = false AND is_previous_traffic_present = true THEN

'delete'

ELSE

'no action'

END As action

```
FROM is_traffic_situation, previous_traffic_check;
```

Result:

is_traffic	is_previous_traffic_present	first	last	action
f	f	POINT(9495704.73841015 3210989.42217862)	POINT(9496772.40364635 3210543.85343115)	no action

4. Testing

4.1. Introduction

Software testing is the process of evaluating the difference between given input to the system and output result from the system. (Zafar, R., 2012). There are various types of testing in software engineering such as unit testing, integration testing, system testing, performance testing. Software testing is one of the important phase of software development. According to one survey software errors costs U.S economy 0.6 percent of the gross domestic product and about 80% of the software development costs of a project are spent on identifying and fixing errors (Khan, M. & Khan, F., 2014).

SAJHA YATAYAT VEHICLE MANAGEMENT AND TRACKING SYSTEM comprise of 3 subsystem. The Web system to manage various data, Passenger App for vehicle tracking and Driver App to update vehicle location to the main system. In the testing, all these 3 systems are tested and performed various checks. In addition, the main purpose of this testing is to test and check if the all the system functionality are working correctly. Following is the main objectives of this testing.

- To ensure that the system meets all the design specification
- To ensure the system meets all the end-user requirements.
- To validate and verify data through user interface.

4.2. Scope

In this testing phase, all the possible areas where the system seem to be critical is performed tests. Following are the list of the scope for this testing:

- Check if the data is correctly stored.
- Check the accuracy of data and if the system provides correct result or not.
- Check the User Interface is correctly functioning. E.g. buttons, navigation to pages, right pages are loaded when option is selected etc.

- Check if the system has proper error handling or not. Provide, various invalid data and check for the result.

4.3. Out of Scope

Following are the tasks that will not be performed in this testing.

- Testing the performance and speed of the system is out of the scope.
- Testing the memory usage of the system is out of the scope.

4.4. Test Assumptions

This system, requires a real time data such as vehicle location data, in order to function certain modules such as Vehicle tracking, traffic Status etc. Therefore, for the testing a test data has been generated in order to mimic the real time scenario.

4.5. Test cases

4.5.1. Passenger Android App

Test Case ID	Description	Precondition	Test Steps	Test Data	Expected Result	Status	Actual Result
TC_login_01	To test if user is successfully logged in to the system.	Login page is launched.	Enter username. Enter password. Click login button.	“ranjit@g mail.com” “12345”	User is successfully logged in and redirected to home page.	Success	The user successfully logged in.
TC_login_02	To test if an appropriate message is displayed when user does not enter username or password field.	Login page is launched.	Enter invalid username. Enter invalid password. Click login button.	“12121sdf” “xyxyxy”	User is displayed “Invalid username/password” error.	Success	Error message displayed.
TC_login_03	To test if an error message is displayed when user enter wrong username or password.	Login page is launched.	Leave username empty. Leave password empty. Click login button.	none	User is displayed “Empty username/password” error.	Success	Error message displayed.
TC_reset_01	To test if user can successfully reset the password or not.	Reset page is launched.	Enter reset code. Enter new password.	“65443” “abcdef”	User successfully reset the password.	Success	User successfully reset the password.

		User provide email to reset password.					
TC_register_01	To test if user can successfully register an account.	Register page is launched.	Enter user name. Enter email. Enter password. Click submit button.	“Ranjit Kaliraj” “ranjit@g mail.com” “12345”	User successfully registered.	Success	User successfully registered.
TC_register_03	To test if an appropriate error message is displayed when user enter blank information.	Register page is launched.	Leave username field blank. Leave email blank. Leave password blank. Click submit.	none	User is displayed error message.	Success	User is displayed error message.
TC_profile_update_01	To test if user successfully update user details or not.	User is successfully logged in.	Enter user name. Enter email. Enter password. Click submit button.	“Rajip Kaliraj” “rajip@gm ail.com” “2222”	User is displayed successful message.	Success	User is displayed successful message.

TC_view_route_01	To test if user can successfully view route on map.	All the route list is retrieved.	Select a route from the list.	none	User is displayed route detail in the map.	Success	User is displayed route detail in the map.
TC_view_traffic_01	To test if user can successfully view traffic details and view affected road on map.	All the route list is retrieved.	Select a route from the list.	none	User is displayed traffic detail for that route.	Success	User is displayed traffic detail for that route.
TC_submit_query_01	To test if user can successfully submit query or not.	Query page is launched.	Enter email. Enter subject. Enter description. Click submit button.	“ranjit@g mail.com” “How to use app” “Some details....”	User is displayed successful query send message.	Success	User is displayed successful query send message.
TC_search_vehicle_01	To test if user is displayed list of result when user search for vehicles.	Search page is launched.	Enter destination name. Click find button.	“Ratopul”	User is displayed list of available vehicles.	Success	User is displayed list of available vehicles.
TC_search_vehicle_02	To test if an appropriate message is displayed provide empty destination.	Search page is launched.	Enter destination name. Click find button.	“xyxyxy”	User is displayed “vehicle not available for	Success	User is displayed “vehicle not available for

					destination” error message.		destination” error message.
TC_search_vehicle_03	To test if an error message is displayed when user provide empty or invalid destination name.	Search page is launched.	Leave destination field empty. Click find button.	none	User is displayed prompt message “Please enter destination”.	Success	User is displayed prompt message “Please enter destination”.
TC_search_vehicle_04	To test if user is displayed appropriate vehicle details for selected vehicle from search list.	Search page is launched.	Select a particular vehicle from the list.	none	User is displayed a vehicle detail.	Success	User is displayed a vehicle detail.
TC_track_vehicle_01	To test a vehicle is correctly bookmarked or not and vehicle details is displayed in dashboard.	Vehicle detail is displayed	Click bookmark/pin button.	none	User is navigated to the dashboard and vehicle is pinned.	Success	User is navigated to the dashboard and vehicle is pinned.
TC_track_vehicle_02	To test if vehicle is correctly displayed in map or not.	Vehicle pinned dashboard is displayed	Click on view vehicle on map button.	none	User is displayed vehicle detail on map.	Success	User is displayed vehicle detail on map.

TC_track_vehicle_03	To test if walking route from user location to the vehicle busstop is displayed or not.	Vehicle is pinned on dashboard.	Click view walking route button.	None	User is displayed walking route from user location to the busstop on map.	Success	User is displayed walking route from user location to the busstop on map.
---------------------	---	---------------------------------	----------------------------------	------	---	---------	---

4.5.2. Web System

Test Case ID	Description	Precondition	Test Steps	Test Data	Expected Result	Status	Actual Result
TC_login_01	To test if user is successfully logged in to the system.	Login page is launched	Enter username. Enter password. Click Login button.	“abcd” “1234”	User is navigated to the home page.	Success	User is navigated to the home page.
TC_login_02	To test if an appropriate message is displayed when user does not enter username or password field.	Login page is launched	Enter invalid username. Enter invalid password. Click login button.	“xyz” “222222”	User is displayed “Invalid Username/Password!!!” message.	Success	User is displayed “Invalid Username/Password!!!” message.

TC_track_vehicle_01	To test if all the available vehicle is displayed in the map or not.	User is logged in. User is in tracking page. Route list is loaded.	Select a route from the dropdown list. “Route 1”		User is displayed list of available vehicle for that route on the map.	Success	User is displayed list of available vehicle for that route on the map.
TC_track_vehicle_02	To test if vehicle details are displayed in the table or not.	User is in tracking page. User selected route.	Navigate to list view.	none	User is displayed vehicle detail on table.	Success	User is displayed vehicle detail on table.
TC_track_vehicle_03	To test if vehicle can be flagged and then successfully unflagged or not.	User is in tracking page. User selected route.	Click flag or unflag button from the vehicle list.	none	Selected vehicle is flagged or unflagged.	Success	Selected vehicle is flagged or unflagged.
TC_add_route_01	To test if user can successfully add route details or not.	User is in route manage page.	Click add vehicle option. Enter route name. Add route stops from list.	“Route 5” “RNAC Bus Station,	User is displayed route successfully added message.	Success	User is displayed route successfully added message.

			Click Add vehicle button.	Jamal Bus Station”			
TC_add_route_03	To test if user can successfully add route way for added route or not.	User is in route manage page.	Navigate to route way add tab. Pin markers on map. Click Save the route way button.	Marker list.	User is displayed route way successfully added message.	Success	User is displayed route way successfully added message.
TC_add_vehicle_01	To test if user can successfully add vehicle details or not.	User is in vehicle manage page.	Click add vehicle detail option button. Enter vehicle name. Click Add button.	“BA 33 KA 1234”	User is displayed vehicle successfully added message.	Success	User is displayed vehicle successfully added message.
TC_add_schedule_01	To test if user can successfully add schedule details or not.	User is in schedule manage page.	Click add schedule detail option button. Select route. Select vehicle. Click Add button.	“Route 1” “BA 33 KA 1234”	User is displayed schedule successfully added message.	Success	User is displayed schedule successfully added message.
TC_add_fare_01	To test if user can successfully add fare details or not.	User is in fare manage page.	Click add fare detail option button. Enter fare amount. Enter distance.	“Rs. 20” “15km”	User is displayed fare successfully added message.		User is displayed fare successfully added message.

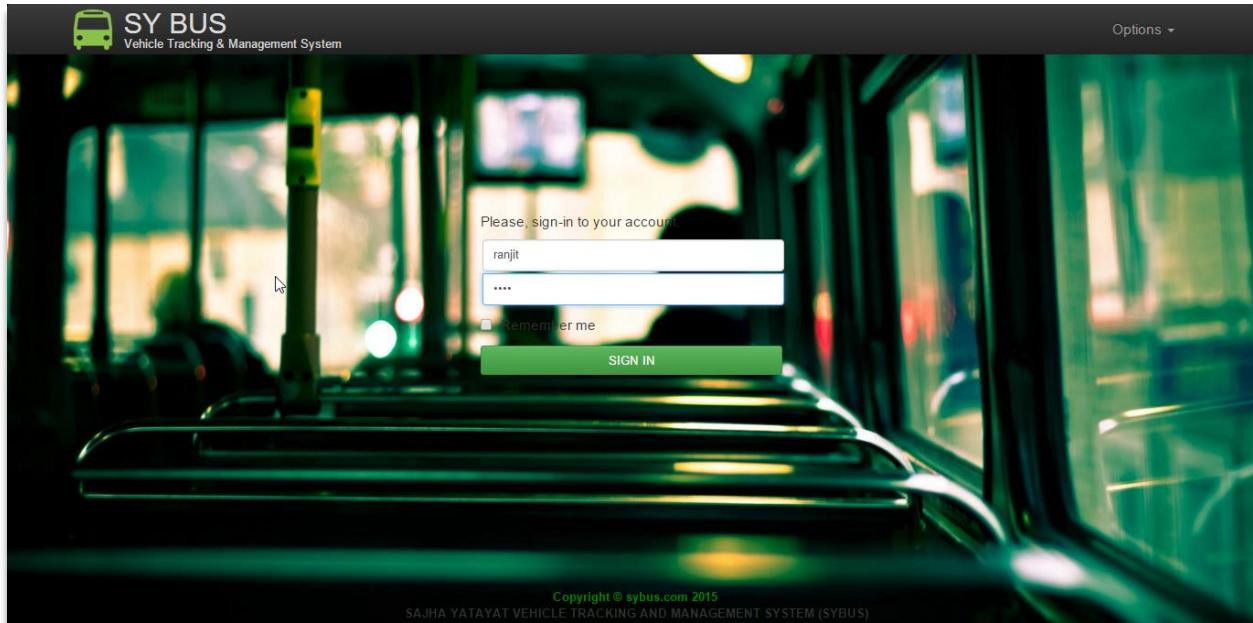
			Click Add button.				
--	--	--	-------------------	--	--	--	--

4.5.3. Driver App

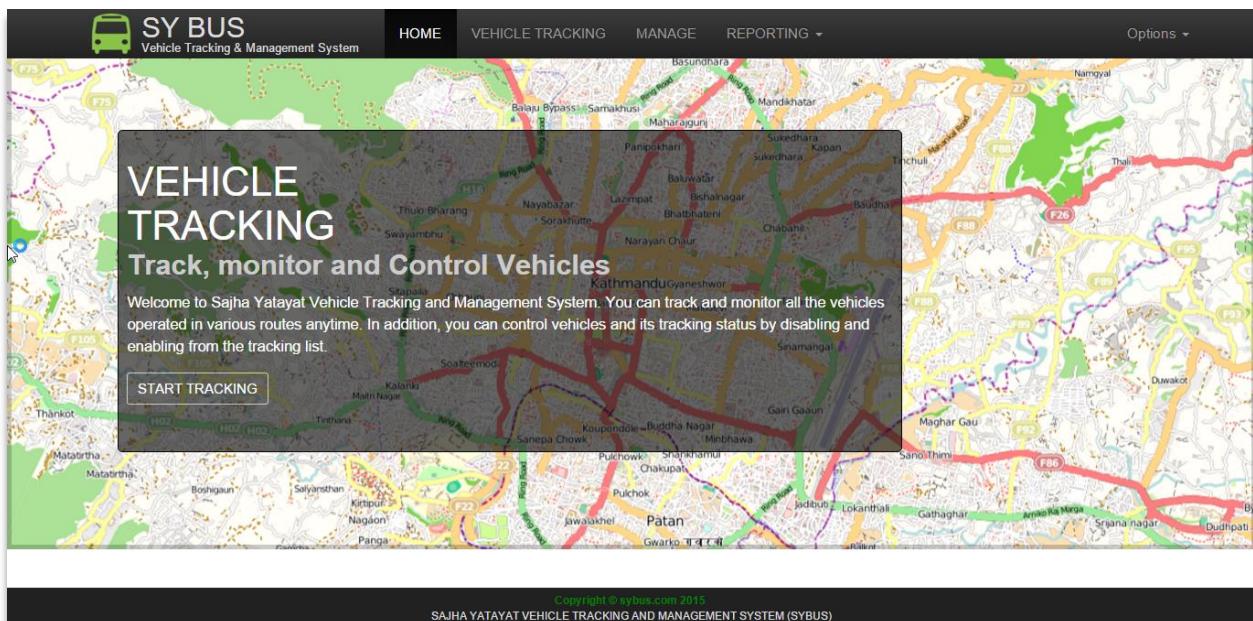
Test Case ID	Description	Precondition	Test Steps	Test Data	Expected Result	Status	Actual Result
TC_update_vehicle_location_01	To test vehicle location data is successfully send to the server.	Route name and vehicle is selected. Vehicle schedule is displayed.	Click start location update button.	none	Vehicle location is successfully updated to the server in interval time.	Success	Vehicle location is successfully updated to the server in interval time.

4.6. Test Execution

- TC_login_01

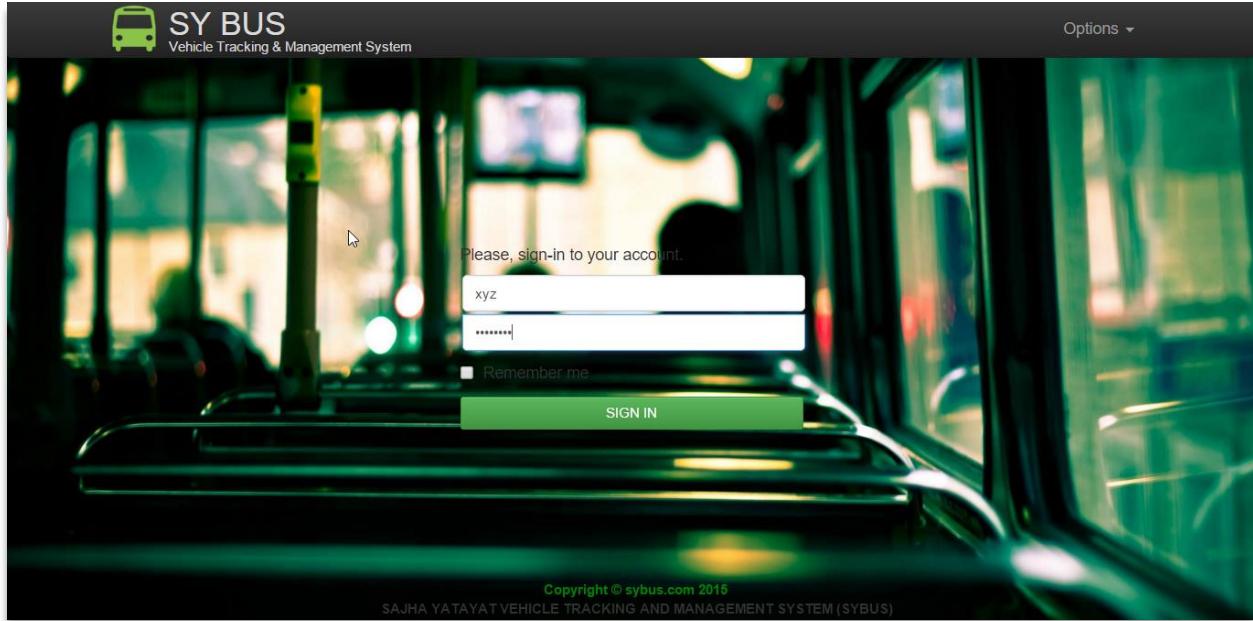


Input: Provide Correct Username and password

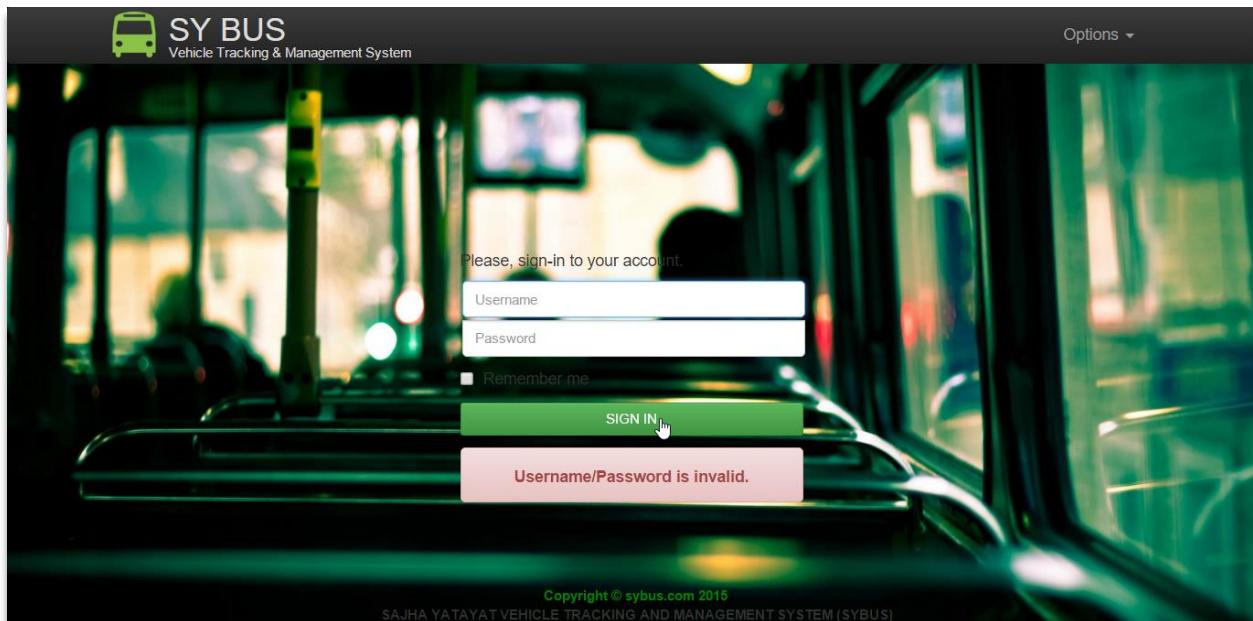


Output: Home page is displayed

- **TC_login_02**



Input: Enter wrong username and/or password



Output: Error message is displayed

- TC_track_vehicle_01

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

MAP VIEW VEHICLE LIST FLAGGED Not Selected Active Vehicles: N/A | Flagged Vehicles: N/A

SELECT ROUTE ▾ Route 1 Route 2

Active Vehicles 0 List of available active vehicles.

Not Available

Flagged Vehicles 0 List of in-active/flagged vehicles.

Not Available

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

© OpenStreetMap contributors

Input: Select a route from the list

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

MAP VIEW VEHICLE LIST FLAGGED Route 1 ▾

Route 1 Active Vehicles: 2 | Flagged Vehicles: 1

VIEW ALL VEHICLES

Active Vehicles 2 List of available active vehicles.

1 BA 11 PA 1111	BA 11 PA 1111
2 BA 12 PA 2222	BA 12 PA 2222

Flagged Vehicles 1 List of in-active/flagged vehicles.

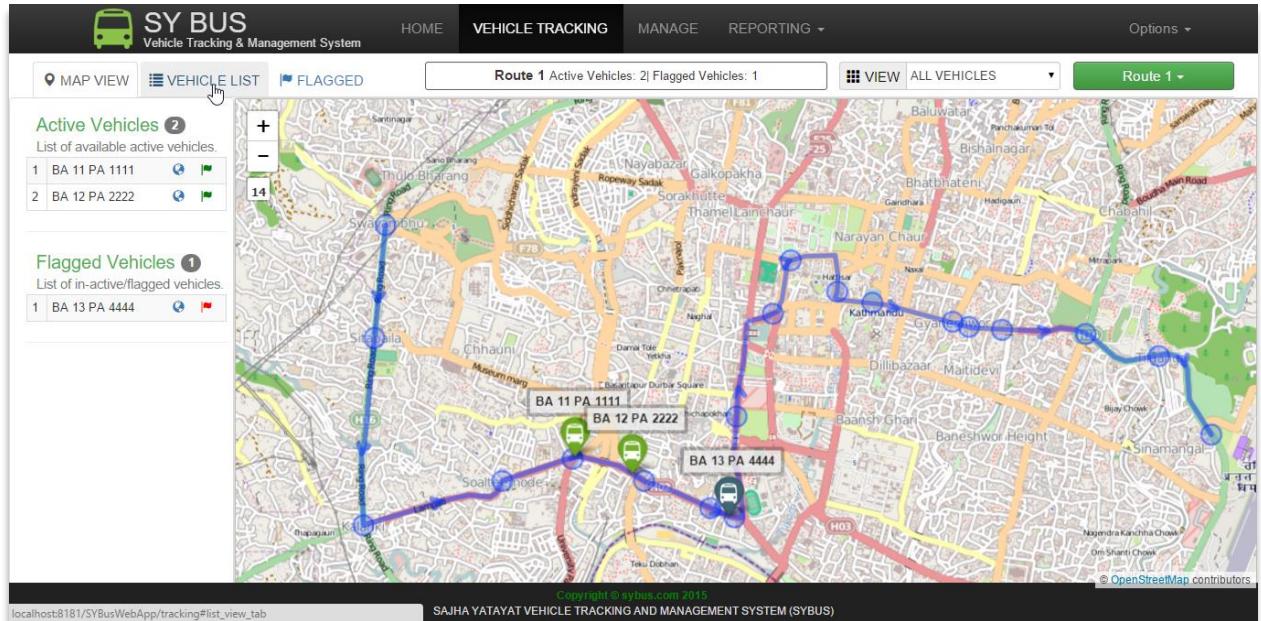
1 BA 13 PA 4444	BA 13 PA 4444
-----------------	---------------

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

© OpenStreetMap contributors

Output: List of available vehicles is displayed on map

- TC_track_vehicle_02



Input: User click Vehicle List tab

MAP VIEW VEHICLE LIST FLAGGED

Route 1 Active Vehicles: 2| Flagged Vehicles: 1

VIEW ALL VEHICLES Route 1

LIST OF VEHICLES Detail list view of all available active vehicles for Route 1

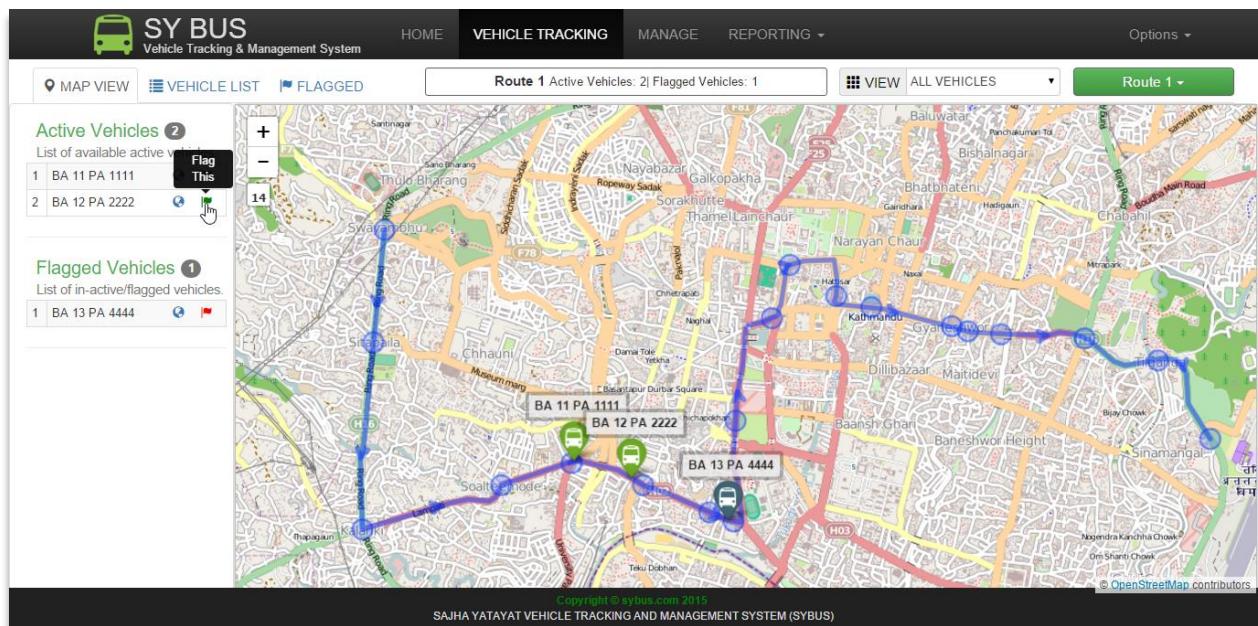
Last refreshed at: :: Refresh

SN	VEHICLE NO.	CURRENT LOCATION	LATITUDE	LONGITUDE	NEXT STOP	ETA	SCHEDULED ON	Flag It
1	BA 11 PA 1111	Soltimode	27.698544	85.29976	Teku Bus Station	3	06:07:00	<input type="checkbox"/> Flag It
2	BA 12 PA 2222	Ganeshman Singh Path	27.697247	85.304596	Teku Bus Station	1	16:09:00	<input type="checkbox"/> Flag It

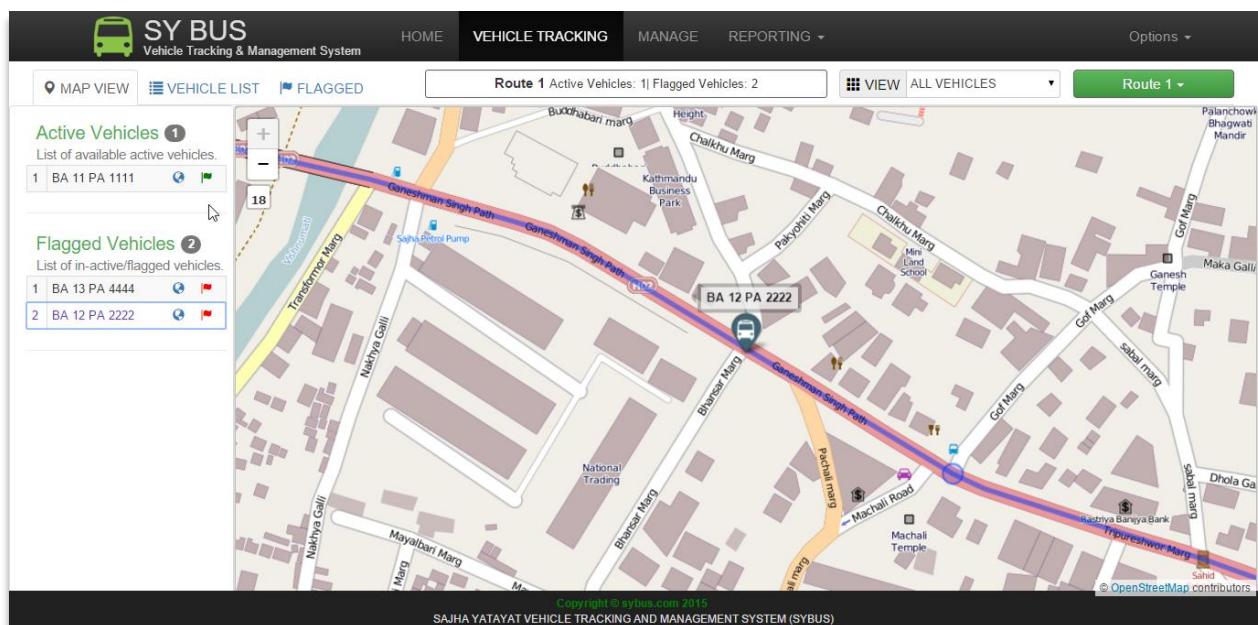
Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Output: List of active vehicle list is displayed correctly

- TC_track_vehicle_03



Input: Select a vehicle and flag a vehicle



Output: Vehicle is successfully flagged

```
sybus=# select * from vehiclesession;
session_id | location_point | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D00FDEE72B07C1C624109ED8EEC857F4841 | f | 1
 58 | 010100002031BF0D0058D6C5AE321D62417D2861DF6D7E4841 | t | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | f | 2
(3 rows)
```

```
sybus=# select * from vehiclesession;
session_id | location_point | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D00FDEE72B07C1C624109ED8EEC857F4841 | f | 1
 58 | 010100002031BF0D0058D6C5AE321D62417D2861DF6D7E4841 | t | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | t | 2
(3 rows)
```

Database data status

- **TC_add_route_01**

The screenshot shows the SY BUS Vehicle Tracking & Management System interface. The top navigation bar includes links for HOME, VEHICLE TRACKING, MANAGE, REPORTING, and Options. The MANAGE tab is selected. On the left, a sidebar menu has 'MANAGE ROUTE' selected, with options for MANAGE SCHEDULE, MANAGE VEHICLE, and MANAGE FARE. The main content area displays a form titled 'ADD ROUTE' with the sub-tutorial 'Add a new Route details.' Below the title is a button labeled 'Close'. The 'ADD NEW ROUTE' dialog box contains fields for 'ROUTE NAME' (Route 5) and 'BUS STOP' (Jamal Bus Station). A note says 'Select and add required Bus Stops from the dropdown list.' Below this is a 'BUS STOP LISTS' section with two items: 'RNAC Bus Station' and 'Jamal Bus Station'. At the bottom of the dialog are buttons for '+ ADD', 'X CANCEL', and a refresh icon.

CURRENT ROUTES List of currently available routes. Refresh

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Input: Enter Route name and bus stops from the list

The screenshot shows the SY BUS Vehicle Tracking & Management System interface. The top navigation bar includes links for HOME, VEHICLE TRACKING, MANAGE, REPORTING, and Options. The MANAGE section is currently active. A sub-menu on the left shows links for Home / Manage / Route, Overview, MANAGE ROUTE, MANAGE SCHEDULE, MANAGE VEHICLE, and MANAGE FARE. The main content area has tabs for ROUTE, BUS STOPS, ROUTE WAY, and STOPS DISTANCE. A modal window titled "ADD ROUTE" is open, displaying a success message: "Successful. You have successfully added new route details. Please, navigate to ROUTE WAY to add GPS co-ordinates." Below this, a table titled "CURRENT ROUTES" lists three routes: Route 1, Route 2, and Route 5. Each route entry includes a "MAP VIEW" link. At the bottom of the page, there is a copyright notice: "Copyright © sybus.com 2015 SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)".

Output: Route is added successfully

Route 1	Swayambhu Bus Station, Sitapaila Bus Station, Kalanki Bus Station, Soalitimood Bus Station, Kalimati Bus Station, Teku Bus Station, Tripureshwor1 Bus Station, Tripureshwor2 Bus Station, RNAC Bus Station, Jamal Bus Station, DurbarMarg Bus Station, Hattisar Bus Station, Kamalpokhari Bus Station, Gyaneshwor Bus Station, Maitidevi Bus Station, Ratnopol Bus Station, Gaushala Bus Station, Tilganga Bus Station, Airport Bus Station
Route 2	Swayambhu Bus Station, Kalanki Bus Station, Tripureshwor1 Bus Station
Route 5	RNAC Bus Station, Jamal Bus Station

(3 rows)

Database data status

- TC_add_route_02

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

Home / Manage / Route ROUTE BUS STOPS ROUTE WAY STOPS DISTANCE

OVERVIEW

MANAGE ROUTE

MANAGE SCHEDULE
MANAGE VEHICLE
MANAGE FARE

CURRENT ROUTE WAYS List of currently available Route Ways for Map.

WAYS ADDED
List of currently available Route ways.
1 Route 1 ...
2 Route 2 ...

NOT ADDED
Ways are not added in following routes.
1 Route 5 +

ADD WAY MODE
You can add marker following route way in the Map above. Note: Click once to add marker and twice to remove it.

ROUTE NAME: Route 5

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Input: Provide route way (markers collection)

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

Home / Manage / Route ROUTE BUS STOPS ROUTE WAY STOPS DISTANCE

OVERVIEW

MANAGE ROUTE

MANAGE SCHEDULE
MANAGE VEHICLE
MANAGE FARE

CURRENT ROUTE WAYS List of currently available Route Ways for Map.

WAYS ADDED
List of currently available Route ways.
1 Route 1 ...
2 Route 2 ...
3 Route 5 ...

NOT ADDED
Ways are not added in following routes.
1 SUCCESSFULLY ADDED. +

ADD WAY MODE
You can add marker following route way in the Map above. Note: Click once to add marker and twice to remove it.

ROUTE NAME: Route 5

(Please, select to add Route Way from the option menu for selected Route in Not Added Route Way section left.)

RAW LIST **+ ADD WAY**

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Output: Route way data is successfully added

```
+-----+
| Route 5 | 01020002031BF0D0006000000031AD2DBB3D1D62414EE9C88A51804841B0AF0CED3D1D6241BC5EA24599804841D32DC6343F1D6241DDF291F0E08048413C94FA7C411D624
5581484169FE3EEA461D6241EC1063899C814841
(1 row)

sybus=#
```

Database data status

- **TC_add_vehicle_01**

The screenshot shows the SY BUS management interface. On the left sidebar, under the 'MANAGE' section, the 'MANAGE VEHICLE' option is selected. The main content area displays the 'ADD VEHICLE' form with the instruction 'Add a new Vehicle details.' A table titled 'CURRENT VEHICLES' lists existing vehicles with their serial numbers and names.

SN	VEHICLE NAME
1	BA 10 PA 3477
2	BA 11 PA 1111
3	BA 12 PA 2222
4	BA 13 PA 4444
5	BA 14 PA 1212
6	BA 15 PA 3321

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Input: Provide Vehicle name

The screenshot shows the SY BUS Vehicle Tracking & Management System. The top navigation bar includes links for HOME, VEHICLE TRACKING, MANAGE, REPORTING, and Options. The left sidebar has links for Overview, MANAGE ROUTE, MANAGE SCHEDULE, MANAGE VEHICLE (which is selected), and MANAGE FARE. The main content area shows a table titled "CURRENT VEHICLES" with columns for SN and VEHICLE NAME. A green success message at the top right says "ADD SUCCESSFUL. You have successfully added a new vehicle details." Buttons for ADD NEW and EXIT are also present.

Output: Vehicle is successfully added

```
sybus=# select * from vehicle;
vehicle_id | vehicle_name
-----+-----
1 | BA 10 PA 3477
2 | BA 11 PA 1111
3 | BA 12 PA 2222
4 | BA 13 PA 4444
5 | BA 14 PA 1212
6 | BA 15 PA 3321
(6 rows)

sybus=# select * from vehicle;
vehicle_id | vehicle_name
-----+-----
1 | BA 10 PA 3477
2 | BA 11 PA 1111
3 | BA 12 PA 2222
4 | BA 13 PA 4444
5 | BA 14 PA 1212
6 | BA 15 PA 3321
7 | KA 22 BA 1111
(7 rows)

sybus=#

```

Database data status

- **TC_add_schedule_01**

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

SCHEDULE

ADD SCHEDULE Add a new Schedule details.

ROUTE: Route 5 VEHICLE: KA 22 BA 1111 TIME: 10:15 C + ADD X EXIT

CURRENT SCHEDULES List of currently available schedules.

SN	ROUTE NAME	VEHICLE NAME	SCHEDULE
1	Route 1	BA 11 PA 1111	06:07:00
2	Route 1	BA 12 PA 2222	16:09:00
3	Route 1	BA 13 PA 4444	01:03:00
4	Route 1	BA 10 PA 3477	04:07:00

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Input: Provide route name, vehicle and schedule time

SY BUS
Vehicle Tracking & Management System

HOME VEHICLE TRACKING MANAGE REPORTING Options

SCHEDULE

ADD SCHEDULE Add a new Schedule details.

ADD SUCCESSFUL. You have successfully added a new schedule details.

ADD NEW EXIT

CURRENT SCHEDULES List of currently available schedules.

SN	ROUTE NAME	VEHICLE NAME	SCHEDULE
1	Route 1	BA 11 PA 1111	06:07:00
2	Route 1	BA 12 PA 2222	16:09:00
3	Route 1	BA 13 PA 4444	01:03:00
4	Route 1	BA 10 PA 3477	04:07:00
5	Route 5	KA 22 BA 1111	10:15

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Output: Schedule is successfully added

```

sybus=# select * from schedule;
schedule_id | vehicle_id | route_id | schedule_time
-----+-----+-----+-----
 1 | 2 | 1 | 06:07:00
 2 | 3 | 1 | 16:09:00
 3 | 4 | 1 | 01:03:00
 4 | 1 | 1 | 04:07:00
 5 | 7 | 3 | 10:15:00
(5 rows)

sybus=# select * from vehicle where vehicle_id=7;
vehicle_id | vehicle_name
-----+-----
 7 | KA 22 BA 1111
(1 row)

sybus=# select * from route where route_id = 3;
route_id | route_name | bus_stops
-----+-----+-----+
 3 | Route 5 | RNAC Bus Station,Jamal Bus Station | 010200002031BF0D000600000031AD2DBB3
411D62418C6AD4A91C8148415A30281B451D62413BEDA4FF5581484169FE3EEA4610$241EC1063899C814841
(1 row)

sybus=#

```

Database Data status

- **TC_add_fare_01**

The screenshot shows the SY BUS software interface for managing fares. The top navigation bar includes links for HOME, VEHICLE TRACKING, MANAGE, REPORTING, and Options. The left sidebar has links for Overview, MANAGE ROUTE, MANAGE SCHEDULE, MANAGE VEHICLE, and MANAGE FARE. The main content area is titled 'ADD FARE' with the sub-instruction 'Add a new Fare details.' Below this is a 'ADD NEW FARE' form with fields for DISTANCE (30) and AMOUNT (50), and buttons for '+ ADD' and 'X EXIT'. Below the form is a table titled 'CURRENT FARES' showing a list of currently available fares with columns for SN, DISTANCE, and AMOUNT.

SN	DISTANCE	AMOUNT
1	5 Km	Rs. 10
2	6 Km	Rs. 12
3	10 Km	Rs. 15
4	20 Km	Rs. 18
5	30 Km	Rs. 22

Copyright © sybus.com 2015
SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)

Input: Provide distance and amount detail

The screenshot shows the SY BUS system interface. The top navigation bar includes links for HOME, VEHICLE TRACKING, MANAGE, REPORTING, and Options. The left sidebar has sections for Overview, MANAGE ROUTE, MANAGE SCHEDULE, MANAGE VEHICLE, and MANAGE FARE. The main content area is titled 'ADD FARE' and displays a success message: 'ADD SUCCESSFUL. You have successfully added a new fare details.' Below this, a table titled 'CURRENT FARES' lists fare details:

SN	DISTANCE	AMOUNT
1	5 Km	Rs. 10
2	6 Km	Rs. 12
3	10 Km	Rs. 15
4	20 Km	Rs. 18
5	30 Km	Rs. 22
6	30 Km	Rs. 50

At the bottom of the page, there is a copyright notice: 'Copyright © sybus.com 2015 SAJHA YATAYAT VEHICLE TRACKING AND MANAGEMENT SYSTEM (SYBUS)'.

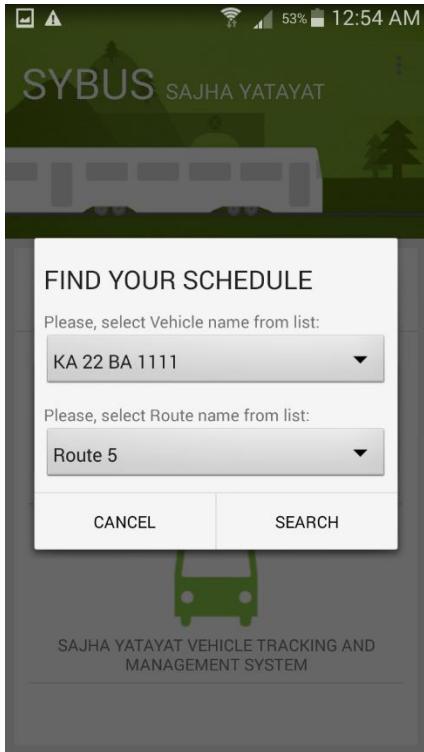
Output: Fare is successfully added

```
sybus=# 
sybus=# select * from fare;
fare_id | amount | distance
-----+-----+
      1 |    10 |      5
      2 |    12 |      6
      3 |    15 |     10
      4 |    18 |     20
      5 |    22 |     30
      6 |    50 |     30
(6 rows)
```

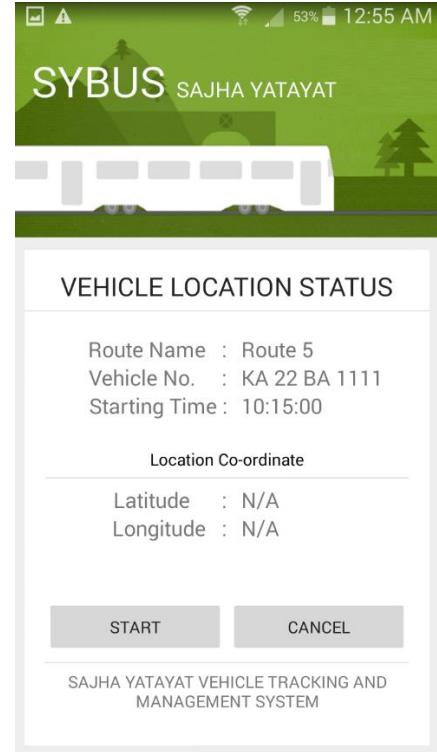
Database data status

Driver App:

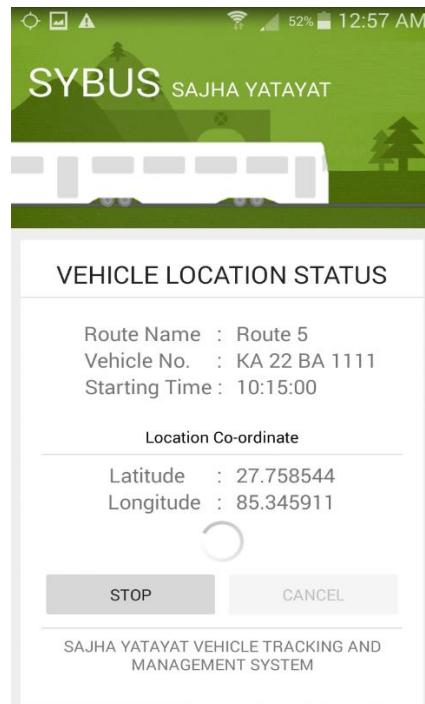
- **TC_update_vehicle_location_01**



Input: Provide vehicle name and route



Output: Display vehicle schedule detail



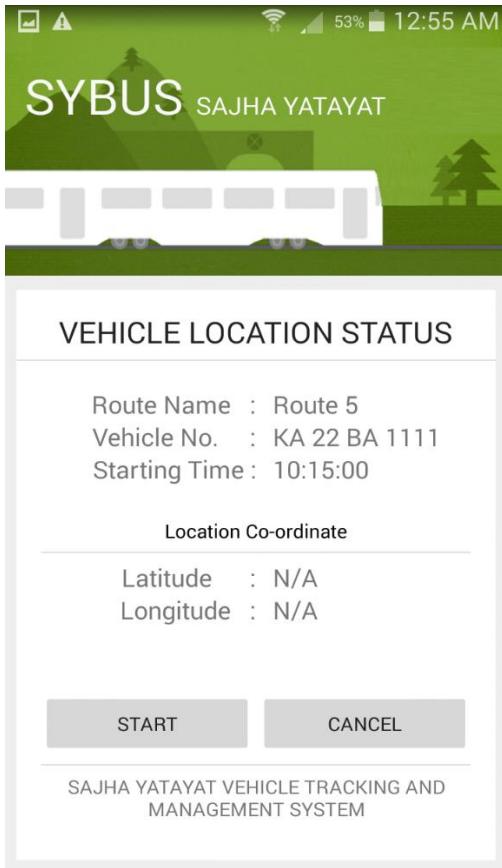
Output: Vehicle location update started

```
sybus=# select * from vehiclesession;
session_id | location_point | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D000FDEE72B07C1C624109ED8EEC857F4841 | f | 1
 58 | 010100002031BF0D00058D6C5AE321D62417D2861DF6D7E4841 | t | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | t | 2
(3 rows)
```

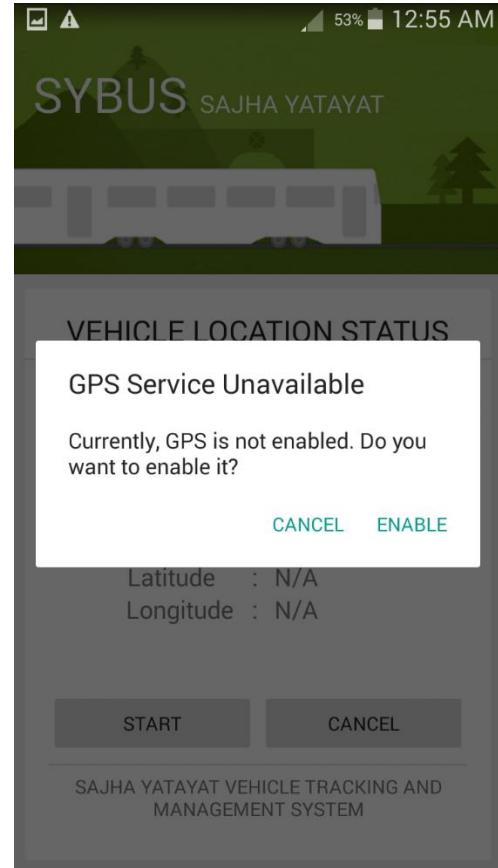
```
sybus=# select * from vehiclesession;
session_id | location_point | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D000FDEE72B07C1C624109ED8EEC857F4841 | f | 1
 58 | 010100002031BF0D00058D6C5AE321D62417D2861DF6D7E4841 | t | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | t | 2
 59 | 010100002031BF0D0069FE3EEA461D6241EC1063899C814841 | f | 5
(4 rows)
```

Database data status

- **TC_update_vehicle_location_02**

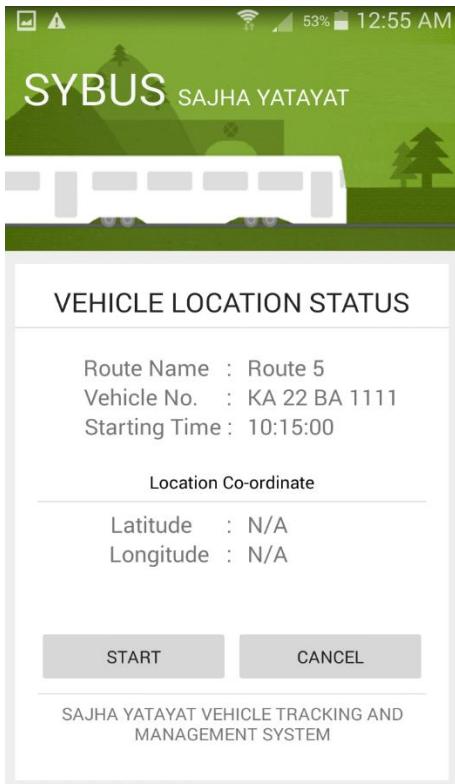


Input: User click start button

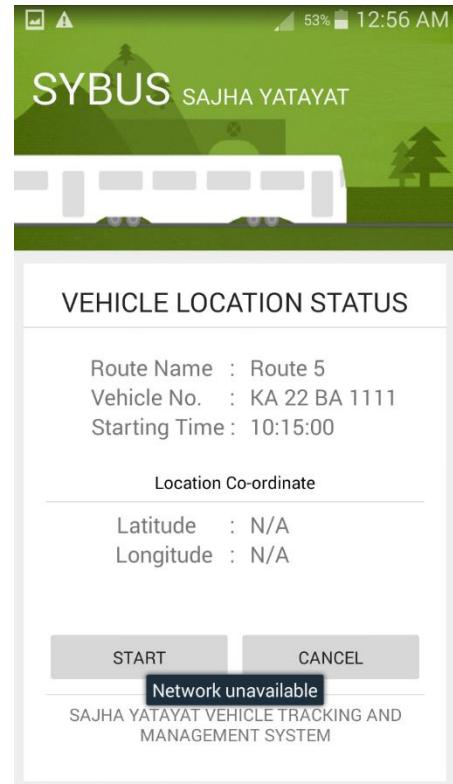


Output: GPS unavailable error message.

- **TC_update_vehicle_location_03**

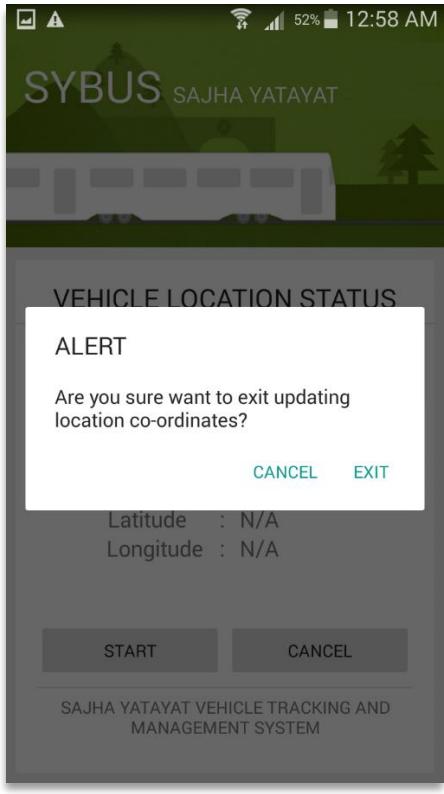


Input: User click start button

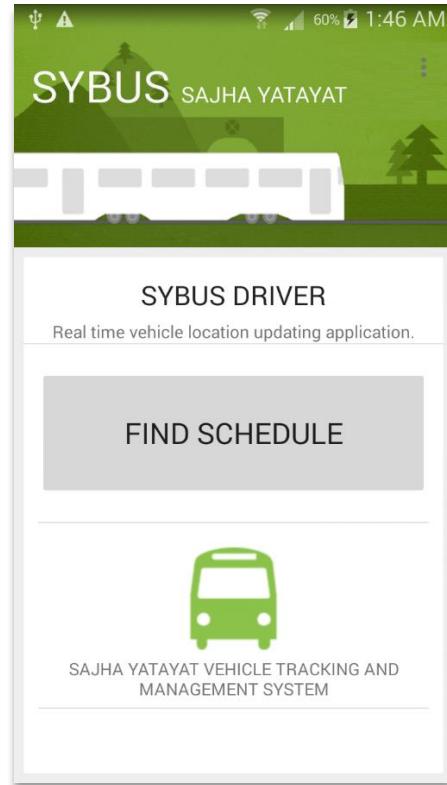


Output: GPS unavailable error message.

- TC_update_vehicle_location_04



Input: User click exit button.



Output: User is displayed home page

```
sybus=# select * from vehiclesession;
session_id |          location_point          | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D00FDEE72B07C1C624109ED8EEC857F4841 | f           | 1
 58 | 010100002031BF0D0058D6C5AE321D62417D2861DF6D7E4841 | t           | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | t           | 2
 59 | 010100002031BF0D0069FE3EEA461D6241EC1063899C814841 | f           | 5
(4 rows)

sybus=# select * from vehiclesession;
session_id |          location_point          | flag_status | schedule_id
-----+-----+-----+-----+
 56 | 010100002031BF0D00FDEE72B07C1C624109ED8EEC857F4841 | f           | 1
 58 | 010100002031BF0D0058D6C5AE321D62417D2861DF6D7E4841 | t           | 3
 57 | 010100002031BF0D0008AA0C06C01C62413F55F763347F4841 | t           | 2
(3 rows)
```

5. Ethical, Legal and Social Issues

“*Ethics* refers to the principles of right and wrong that individuals, acting as free moral agents, use to make choices to guide their behaviors” (Zheng, E., 2013). In information System, Ethical, Legal and Social Issues are major problem today. In addition, these issues are closely related to each other. Following are detail explanation of these issues on context of this system.

5.1. Ethical Issues

Following, are possible ethical issues for this System.

5.1.1. Privacy and Data Protection

Privacy is one of the major issue today in computer system. This system require various personal and critical information of the users including location information. In addition, this system also feature online payment and this is another major concerns of user’s privacy and data protection.

Solution: In order to protect user personal information, this system use encryption in order to store user information such as password. In addition, when finding and tracking vehicles, the user location is never send to the server and therefore there is no issue for user-location privacy expose. Also, user email address is only used for system access and not used for any promotional activities. Furthermore, the online payment is transparent. User is redirected to the payment gateway page in order to make payment and the user personal information is not stored in device or server.

5.1.2. Liability

Liability is another issue in software system. The computer system must be reliable and must store and provide correct data to the user correctly and there are no error in the system. Various vehicle tracking related data and passenger’s sensitive data are stored in the system and retrieved. These data must be stored correctly. In addition, various information, such as traffic information, vehicle ETA and current location information etc. must be accurate.

Solution: In order to make system reliable, this system regularly backup various data including user data regularly. In addition, the system is heavily tested and verified to ensure all the data are stored and retrieved correctly. Furthermore, the traffic information is based on previous vehicle location history and there might be some inaccuracy in some cases. This problem must be explained in user terms and conditions. In addition, the vehicle location is regularly update 4 time in every minute. Therefore, there is less chances on vehicle location accuracy.

However, the estimated time of arrival of the vehicle is based on past location history and therefore, in some cases result can be inaccurate. But on the other hand, the user can also view real time vehicle location on map and which help user to minimize the inaccurate ETA problem.

5.2. Legal Issues

5.2.1. Intellectual Property and Software Licensing

“Intellectual Property is personal ownership of the creative ideas that develop from an individual’s mind or intellect or ideas and knowledge in which a business has obtained an interest” (purplezeus.com, n.d.). There are high risk that one’s software property can be used/forged on unauthorized any other organization or person. Therefore, software product must be kept secured.

Solution: In order to minimize intellectual property issue, the product must be licensed and end-user license agreement need to be produced. In addition, various lawsuits and rules need to be produced in case the product is used violating the rule, appropriate action can be taken.

5.2.2. Patent and Copyright infringement

A copyright protects works of authorship that have been tangibly expressed in a physical form (uspto.gov, n.d.). On the contrary to securing intellectual property of a software product, it is also need to be ensured that the product did not violate and infringe any patent and copyrights as well. For example, it is need to be ensured that the software product have not used deliberately any other proprietary third party software or peace of code.

Solution: All the essential system codes are developed from the scratch and do not use any third party proprietary codes. In addition, this system uses various open source software framework for various functionalities. For example, for offline map, OpenStreetMap, which is one of the popular opensource map service is used. In addition, other open source tools used in this system are OSMOSIS, tilecache, mapsforge, mapnik, Graphhopper etc.

5.3. Social Issues

5.3.1. Software usage, easy accessibility and convenience

One of the social issue is that a software product must be accessible to end user easily. In addition, the software need to consume as much less data in order to maintain end-user convenience and easy accessibility.

Solution: The passenger app is a portable system and is available for smartphone. Therefore, this system can be easily accessible to end-user. In addition, this system is freeware and tracking service provided by this system is available to everyone as long as they use the android app. Furthermore, the login credential is only required for payment and therefore for other use, users are not required to go through any extra activity in order to use this system. In addition, end-user application is embedded with offline map and therefore, it helps to minimize volume of data download from the server and provide greater accessibility.

6. Conclusion and Recommendations

Sajha Yatayat Vehicle Tracking and Management System is an online web system for tracking vehicle and managing various vehicle related information. In addition, this system has a sub-system intended to the end-user or passenger in order to monitor and track vehicle of Sajha Yatayat and realtime vehicle location information.

For the development of this system, Enhanced waterfall methodology is used since all the requirement are clearly known and project need to deliver in the end as complete system. The project was initiated after studying the feasibility of the proposed system. After feasibility study, it is found that the project is feasible in the current scenario of Nepal. The project length is 20 week and a Gantt chart has been produced a project planning in order to conduct this project.

After requirement analysis, 17 use cases has been derived as an essential requirement of this system and the system has 3 sub-system; web system, Passenger android app and Driver android app. The web system is mainly focused on vehicle data management and tracking and monitoring of the vehicles. Similarly, the Passenger's app is intended to track vehicles and get various information such as estimated time of arrival, traffic situation etc.

The web system is web based and runs on browsers. On the other hand, passenger App and driver app work on android mobile operating system. An internet connection is required in order to user various services of this system.

This system is modelled using object oriented paradigm. Therefore, Unified Modelling language (UML) has been used. Similarly, this system is developed using object oriented Java programming language. In addition, the system is developed using spring MVC framework. In web system application layer, JQuery and Bootstrap has been used. Similarly, for map utility, OpenstreetMap has been used and local map service has been produced using Mapnik, Leaflet, Tilecache and OpenstreetMap data. In addition, for database PostgresSQL and PostGIS is used. During testing phase, all the possible system scenario has been tested and all the result are success.

Developing this system was the most difficult and challenging project of my entire 4 year semester. But on the other hand, I have learned various technical and project manager skills and abilities and industry level experience.

6.1. Learning from this project

After the successful completion of the project I have gained various technical and non-technical skills. The major learning is software engineering approach for developing a software system in correct and discipline manner. Most of the learned skills were already gained in previous semesters modules and this project helped me to sharpen these skills.

First, I have learned how to properly produce a project planning form project initiation to implementation. In addition, I have learned how to correctly gather requirements from the problem domain and conduct the feasibility study of the project in order to identify whether the project can be developed or not. The one of the important skills gained to produce SRS document. SRS document is one of the important and essential document which helps to identify and describe the system that we are going to develop. Another important learning is to system design, from system architecture design to various process design using UML paradigm and database designs.

In addition to these software engineering and project management learning, I have gained many new technical skills. One of the continuous learning is Java, since the system developed is completely coded on java programming language. Following are the list of important skills that I have learned throughout this system development.

- PostGIS (For location data storage) and PostgreSQL DBMS.
- Bootstrap Framework
- JavaScript and JQuery framework
- Spring framework and Spring MVC
- Jackson library for binding and converting Java class into JSON.
- Create offline map using OSM data, OSM2Pgsq, Mapnik, tilecache and leaflet library.
- Restful webservice using JAX-RS and Jersey
- Android application development using Material design

- MVP design pattern for android
- Mapforge and Graphhoper for android offline map.
- Eclipse, Toad Data modeler, Astah Design tool, Android studio, Maven etc.

Overall, I have gained a complete beginner experience of how to initiate a small IT project from project initialization to testing and implementation. And, Now I am able to make a web system, an restful web service and android app and implement an offline map in web and android application. Furthermore, I have gained much knowledge of spatial computing and location service.

6.2. Problems and challenges

I have chosen enhance waterfall methodology for this project where all the tools and technique must be known beforehand but despite of that prior to starting this project, most of the tools and framework used during this project was unknown to me and I have to go through learning of each and every tools on my own and as a result I had to really work hard during the development phase. I have taken help of various online sites and resources and one of the best site were stackoverflow.com, stackexchange.com etc. which had become my essential part throughout this project.

Some of the major challenges and problems are following:

- Problem on how to implement offline map and which took me around a month to learn and implement.
- Android app development and material design is another challenging task since I have no experience on android development before this project.
- Implementing MVP design pattern on Android is very tedious and took me a week to learn.
- Initially, prior to this project, I have knowledge of PostgresSQL database. But in order to store various location and geometry/geography data and perform various spatial calculation such as distance, length etc. PostGIS (Extension of PostgreSQL) database is required. This also took some time to learn.
- Other Various challenging algorithm problems: Traffic finder, ETA calculation, finding vehicle nearest to user destination, finding nearest next bus stop of vehicle in nearest position

of the user location, identifying and checking if the vehicle is in the right direction of the user's location or not in order to find nearest vehicles etc.

Things I cannot do but which can be done in future escalation:

- Combination of Openstreetmap and google map in order to provide accurate information. For example, most of the destination and places name is not available in OSM, and which is available in google map. Therefore, I wanted to combine both in order to give precise result. Currently, I have only used OSM data.
- Implement live location change feature in map in android app due to the limited documentation and less knowledge of Graphhopper Map API for vehicle tracking. Therefore, currently user need to manually refresh map in order to see changed location on map. On the other hand, ETA and vehicle current road name information are updated in interval time.

Things you could have done differently:

- Use facebook/twitter login validation instead of registration enabled login, much easier and helps to avoid registration and signups and provide more flexibility to the user.
- Cache various data such as destinations, route, vehicle, and Bus stop etc. data on android app's local SQLite database, So that the user request and calculation become fast. For example, in current scenario, the suggestion for destination when user type in search bar is retrieved from the server and it take some 5 seconds and more in order to display suggestion in order to provide greater accessibility to user. However, it was implemented in SQLite database, the suggestion have appeared in milliseconds.

6.3. Recommendations and future Escalation

Following are some recommendation for various features that can be implemented in this system in near future.

Bus Stop Information Board

This function enable user to get various information related to Bus including estimated arrival time, route name and other Bus details. The main benefit of this feature is that passenger can get the information about the bus when they are waiting in the Bus stop. Not all people may use smartphone therefore this feature might be very useful to them. For this system, only the software part of this feature will be developed in the form of system API i.e. no hardware related work will be performed.

Bus Stop Information Board

This function enable user to get various information related to Bus including estimated arrival time, route name and other Bus details. The main benefit of this feature is that passenger can get the information about the bus when they are waiting in the Bus stop. Not all people may use smartphone therefore this feature might be very useful to them. For this system, only the software part of this feature will be developed in the form of system API i.e. no hardware related work will be performed.

Web version of android application

Currently, user can access vehicle tracking service via only Android application. Therefore, only limited user can get benefit of this system. Therefore, a web version of the android application can be produced so that any user can easily track vehicle using web browser on any device.

7. Reference

Arthur, C. (2012) *The history of smartphones: timeline* [Online] Available at: <http://www.theguardian.com/technology/2012/jan/24/smartphonestimeline> [Accessed on: 12th April 2015]

Balaji, S. (2012) WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *International Journal of Information Technology and Business Management* [Online] 2(1/June) p.26-30 Available at: <http://jitbm.com/Volume2No1/waterfall.pdf> [Accessed on: 12th April 2015]

CA Technologies (2011) *Agile: An Introduction* [Online] Available at: <https://www.youtube.com/watch?v=OJfIDE6OaSc> [Accessed on: 5th April 2015]

Chivers, T. (2013) The Story of Google Maps [Online] Available at: <http://www.telegraph.co.uk/technology/google/10090014/ThestoryofGoogleMaps.html> [Accessed on: 6th April 2015]

Cohn, M. (2012) *Difference Between Scrum and Extreme Programming* [Online] Available at: <http://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming> [Accessed on: 5th April 2015]

Croslow, A. (2013) *A beginner's Guide to Different Satellite Navigation Systems* [Online] Available at: <http://linxtechnologies.com/blog/beginners-guide-satellite-navigation-systems/> [Accessed on: 12th April 2015]

CTA Bus Tracker (n.d.) *CTA Bus Tracker* [Online] Available at: <http://www.ctabustracker.com/bustime/home.jsp> [Accessed on: 19th Feb. 2015]

Favre, L. (2015) *Best GPS and navigation apps for Android* [Online] Available at: <http://www.androidpit.com/best-offline-gps-apps-for-android> [Accessed: 8th April 2015]

Gautam, A. (2011) *History of Navigation* [Online] Available at: www.slideshare.net/gautamanurag/history-of-navigation-8152806 [Accessed on: 10th April 2015]

Google (n.d.) About – Google Maps [Online] Available at: <http://maps.google.com/maps/about> [Accessed on: 6th April 2015]

Johnson, P. (2012) *Can a lone developer use Agile* [Online] Available at: <http://www.itworld.com/article/2714022/it-management/can-a-lone-developer-use-agile-.html?page=2> [Accessed on: 5th April 2015]

Khan, M. & Khan, F. (2014) Importance of Software Testing in Software Development Life Cycle, *International Journal of Computer Science Issues* vol 11, No. 2, pp.120-121, Available at: <http://www.ijcsi.org/papers/IJCSI-11-2-2-120-123.pdf> [Accessed on: 25th Dec. 2015]

Lameguy (2013.) *The “Big-Bang” Software development lifecycle model* [Online] Available at: <https://lameguy.wordpress.com/2013/05/29/the-big-bang-software-development-lifecycle-model/> [Accessed on: 5th April 2015]

Maheshwari, S. & Jain, D. (2012) A Comparative Analysis of Different types of Models in Software Development Life Cycle.*International Journal of Advanced Research in Computer Science and Software Engineering* [Online] www.ijarcsse.com 2(5/May). p.285-290 Available at: http://www.ijarcsse.com/docs/papers/May2012/Volum2_issue5/V2I500405.pdf [Accessed on: 28th March 2015]

Moovit (n.d.) *Ride Public Transit Smarter: Get the fastest, least crowded routes in real time* [Online] Available at: <http://www.moovitapp.com> [Accessed on: 19th Feb. 2015]

Munassar, N. & Govardhan, A. (2010) A Comparison Between Five Models Of Software Engineering.*International Journal of Computer Science Issues* [Online] www.IJCSI.org 7 (5/September) p.94-100 Available at: <http://www.ijcsi.org/papers/7-5-94-101.pdf> [Accessed on: 5th April 2015]

NextBus (n.d.) *NextBus Real-Time Passenger Solution* [Online] Available at: <http://cts.cubic.com/solutions/real-timepassengerinformation/nextbus.inc.aspx> [Accessed on: 19th Feb. 2015]

purplezeus.com, (n.d.) Social and Ethical issues [Online] Available at: <http://www.purplezeus.com/social-and-ethical-issues.html> [Accessed on: 2nd Jan 2016]

Rastogi, V. (2015) Software Development Life Cycle Models-Comparison, Consequences. *International Journal of Computer Science and Information Technologies* [Online] 6 (1). p.168-172. Available at: <http://www.ijcsit.com/docs/Volume%206/vol6issue01/ijcsit2015060137.pdf> [Accessed on: 28th March 2015]

Schutz, Y. & Chambaz, A. (n.d.) Short Communication: Could a satellite-based navigation system (GPS) be used to assess the physical activity of individuals on earth?.*European Journal of Clinical Nutrition* [Online] 51.p338-339. Available at: <http://onlinewww.jpier.org/PIER/pier.php?paper=06090504> [Accessed on: 6th April 2015]

Sullivan, M. (2012) *A brief history of GPS* [Online] Available at: <http://www.techhive.com/article/2000276/a-brief-history-of-gps.html> [Accessed on: 10th April 2015]

Techrepublic.com (2006) *Understanding the pros and cons of the Waterfall Model of software development* [Online] Available at: <http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development> [Accessed on: 6th April 2015]

uspto.gov(n.d.) Trademark, Patent and Copyright? [Online] Available at: <http://www.uspto.gov/trademarks-getting-started/trademark-basics/trademark-patent-or-copyright> [Accessed on: 29th Dec. 2015]

WiseGeek (n.d.) *What Is a Navigation System?* [Online] Available at: <http://www.wisegeek.com/whatisanavigationsystem.htm#didyouknowout> [Accessed on: 12th April 2015]

Zafar, R. (2012) What is Software testing? What are different types of testing? [Online] www.codeproject.com, Available at: <http://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty> [Accessed on: 25th Dec. 2015]

Zeil, S. (n.d.) *Software Development: The Waterfall Model* [Online] Available at: <http://www.cs.odu.edu/~cs333/website-s11/Lectures/waterfall/page/waterfall.html> [Accessed on: 6th April 2015]

Zheng, E. (2013) Ethical and Social Issue in Information System. [Online] Available at: <http://ocmis.blogspot.com/2013/03/ethical-and-social-issues-in.html> [Acessed on: 28th Dec. 2015]