# Table of contents

[Table of contents 1](#_Toc430522705)

[List of Table 2](#_Toc430522706)

[List of Figures 3](#_Toc430522707)

[Definitions, Acronyms, and Abbreviations 4](#_Toc430522708)

[A. Introduction 5](#_Toc430522709)

[1. Project Information 5](#_Toc430522710)

[2. Introduction 5](#_Toc430522711)

[3. Current Situation 5](#_Toc430522712)

[4. Problem Definition 5](#_Toc430522713)

[5. Proposed Solution 6](#_Toc430522714)

[5.1. Feature functions 6](#_Toc430522715)

[5.2. Advantages and disadvantages 7](#_Toc430522716)

[6. Functional Requirements 7](#_Toc430522717)

[7. Role and Responsibility 8](#_Toc430522718)

[B. Software Project Management Plan 9](#_Toc430522719)

[1. Problem Definition 9](#_Toc430522720)

[1.1. Name of this Capstone Project 9](#_Toc430522721)

[1.2. Problem Abstract 9](#_Toc430522722)

[1.3. Project Overview 9](#_Toc430522723)

[2. Project organization 13](#_Toc430522724)

[2.1. Software Process Model 13](#_Toc430522725)

[2.2. Roles and responsibilities 13](#_Toc430522726)

[3. Project Management Plan 15](#_Toc430522727)

[3.1. Software development life cycle 15](#_Toc430522728)

[3.2. Phase Detail 18](#_Toc430522729)

[3.3. Task sheet 19](#_Toc430522730)

[3.4. All Meeting Minutes 19](#_Toc430522731)

[4. Coding Convention 19](#_Toc430522732)

[4.1. Java Coding Convention 19](#_Toc430522733)

[4.2. Android Coding Convention 20](#_Toc430522734)

[G. Appendix 22](#_Toc430522735)

# List of Table

[Table 1: Definitions, Acronyms, and Abbreviations 6](#_Toc431549797)

[Table 2: Roles and Responsibilities 10](#_Toc431549798)

[Table 3: Hardware requirement for continuous integrating server 13](#_Toc431549799)

[Table 4: Hardware requirement for web development 14](#_Toc431549800)

[Table 5: Hardware requirement for mobile development 14](#_Toc431549801)

[Table 6: Hardware requirement for wear development 14](#_Toc431549802)

[Table 7: Software requirement 15](#_Toc431549803)

[Table 8: Roles and responsibilities 18](#_Toc431549804)

[Table 9: Tools and Techniques 18](#_Toc431549805)

[Table 10:Software development life cycle 20](#_Toc431549806)

[Table 11: Requirements definition 21](#_Toc431549807)

[Table 12: System and software design 21](#_Toc431549808)

[Table 13: Implementation and unit test 22](#_Toc431549809)

[Table 14: Integration and system testing 22](#_Toc431549810)

[Table 15: Operation and maintenance 23](#_Toc431549811)

[Table 16:Naming conventions for drawables 24](#_Toc431549812)

[Table 17: Naming conventions for icons 25](#_Toc431549813)

[Table 18: Use case WG01 - <Guest> Login 35](#_Toc431549814)

[Table 19: Use case WS01 - <Staff> Logout 37](#_Toc431549815)

[Table 20: Use case WS02 - <Staff> Approve or reject all current system notifications. 39](#_Toc431549816)

[Table 21: Use case WS03 - <Staff> Approve or reject detail notification. 41](#_Toc431549817)

[Table 22: Use case WS04 - <Staff> Unlock notification. 43](#_Toc431549818)

[Table 23: Use case WS05 - <Staff> Configure parsing source. 45](#_Toc431549819)

[Table 24: Use case WS06 - <Staff> Choose parsing time. 47](#_Toc431549820)

[Table 25: Use case SC01 - <Staff> Configure parsing source. 50](#_Toc431549821)

[Table 26: Use case SC02 - <Staff> Parse bus timetable. 52](#_Toc431549822)

[Table 27: Use case SC03 - <Staff> Configure parsing source. 54](#_Toc431549823)

[Table 28: Use case WS07 - <Staff> Add staff. 57](#_Toc431549824)

[Table 29: Use case WS08 - <Staff> Add staff. 59](#_Toc431549825)

[Table 30: Use case MB01 - <Customer> Search Arbitrary Location on map 64](#_Toc431549826)

[Table 31: Use case UC02 - <Customer> Search Bus Route 68](#_Toc431549827)

[Table 32: Use case MB03 - <Customer> Navigate Current Location On Map 71](#_Toc431549828)

[Table 33: Use case MB04 - <Customer> Search Motorbike Route 75](#_Toc431549829)

[Table 34: Use case MB05 - <Customer> Reference Bus Route Detail 77](#_Toc431549830)

[Table 35: Use case MB06 - <Customer> Reference Bus Timetable Detail 79](#_Toc431549831)

[Table 36: Use case MB07 - <Customer> Configuration Wear Setting 80](#_Toc431549832)

[Table 37: Use case MB08 - <Customer> Synchronize with Wear 82](#_Toc431549833)

[Table 38: Use case MB09 - <Customer> Synchronize Data with Server 84](#_Toc431549834)

[Table 39: Use case UR01 - <Customer> Show information of a bus station on map 87](#_Toc431549835)

[Table 40: Use case WR02 - <Customer> Show information of a bus station on map 89](#_Toc431549836)

[Table 41: Use case WR03 - <Customer> Navigate current place map 90](#_Toc431549837)

[Table 42: Use case WR04 - <Customer> Navigate current place map 92](#_Toc431549838)

[Table 43: Use case WR05 - <Customer> Notify at motorbike turn 94](#_Toc431549839)

[Table 44: Use case WR06 - <Customer> Show list bus station 95](#_Toc431549840)

[Table 45: Use case WR07 - <Customer> Show list motorbike turn 96](#_Toc431549841)

[Table 46: Use case WR08 - <Customer> Show bus station on map 98](#_Toc431549842)

[Table 47: Use case WR09 - <Customer> Show motorbike on map 100](#_Toc431549843)

[Table 48: Conceptual Diagram Data Dictionary 102](#_Toc431549844)

# List of Figures

[Figure 1: Waterfall model 17](#_Toc431549845)

[Figure 2: Web application use case 32](#_Toc431549846)

[Figure 3: Android mobile application use case 32](#_Toc431549847)

[Figure 4: Android wear application use case 33](#_Toc431549848)

[Figure 5: <Guest> Overview Use Case 34](#_Toc431549849)

[Figure 6: <Guest> Login 34](#_Toc431549850)

[Figure 7 <Staff> Overview Use Case 36](#_Toc431549851)

[Figure 8 <Staff> Logout 37](#_Toc431549852)

[Figure 9: <Staff> Approve or reject detail notification 40](#_Toc431549853)

[Figure 10: <Staff>Block and Unblock notification 43](#_Toc431549854)

[Figure 11: <Staff> Configure parsing source 44](#_Toc431549855)

[Figure 12: <Staff> Configure parsing time 46](#_Toc431549856)

[Figure 13: <Scheduler> Overview Use Case 48](#_Toc431549857)

[Figure 14: <Staff> Parse bus route 49](#_Toc431549858)

[Figure 15: <Staff> Parse bus timetable 52](#_Toc431549859)

[Figure 16: <Staff> Parse data periodically 54](#_Toc431549860)

[Figure 17: <Admin> Admin Overview Use Case 55](#_Toc431549861)

[Figure 18: <Admin> Add staff 56](#_Toc431549862)

[Figure 19: <Admin> Edit staff 58](#_Toc431549863)

[Figure 20: <Guest> Search Location 61](#_Toc431549864)

[Figure 21: <Customer> Search Arbitrary Location on map 62](#_Toc431549865)

[Figure 22: <Customer> Search Bus Route 65](#_Toc431549866)

[Figure 23: <Customer> Navigate Current Location On Map 69](#_Toc431549867)

[Figure 24: <Customer> Search Motorbike Route 72](#_Toc431549868)

[Figure 25: <Customer> Reference Bus Route Detail 76](#_Toc431549869)

[Figure 26: <Customer> Reference Bus Timetable Detail 78](#_Toc431549870)

[Figure 27: <Customer> Configuration Wear Setting 80](#_Toc431549871)

[Figure 28: <Customer> Synchronize with Wear 82](#_Toc431549872)

[Figure 29: <Customer> Synchronize Data with Server 84](#_Toc431549873)

[Figure 30: <Customer> Overview use case 86](#_Toc431549874)

[Figure 31: <Customer> Show information of a bus station on map 87](#_Toc431549875)

[Figure 32: <Customer> Show information of a bus station on map 88](#_Toc431549876)

[Figure 33: <Customer> Navigate current place map 90](#_Toc431549877)

[Figure 34: <Customer> Navigate current place map 92](#_Toc431549878)

[Figure 35: <Customer> Notify at motorbike turn 93](#_Toc431549879)

[Figure 36: <Customer> Show list bus station 95](#_Toc431549880)

[Figure 37: <Customer> Show list motorbike turn 96](#_Toc431549881)

[Figure 38: <Customer> Show bus station on map 98](#_Toc431549882)

[Figure 39: <Customer> Show motorbike on map 99](#_Toc431549883)

[Figure 40: Conceptual diagram 102](#_Toc431549884)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| SWR | Smart Wear on your Route |
| API | Application Programming Interface |
| Wear device | Smart watch that uses Android Wear OS 4.4 or above |
| Staff | Administrator of website |
| BusMap | Official mobile application developed by Ho Chi Minh Ministry of Communications and Transport -http://www.buyttphcm.com.vn/Detail\_News.aspx?sl=717 |
| RAPTOR | Round-based Public Transit Optimized Router |
| mcRAPTOR | More criteria RAPTOR |
| Station | Distinct location in the network where one can board or get off a vehicle (bus, train) |
| Round | Represents a sequence of stations a specific vehicle (train, bus, subway …) |
| Connection | A connection models a vehicle departing at one start station to end station of one trip without intermediate halt. |
| Trip | Trip is a round with time arrival information at each station. One trip often has many routes. |
| Footpath | Model walking connection between stations. |
| Pareto Set | A subset of the set of feasible points of solutions that contains all points that have at least one objective optimized while holding all other objectives constants. |

Table 1: Definitions, Acronyms, and Abbreviations

# Introduction

## Project Information

Project name: Smart Wear on Your Route

Project Code: SWR

Product Type: Website, Android and Android Wear application

Start Date: September 7th, 2015

End Date: December 20th, 2015

## Introduction

Nowadays, within the strong development of presently economy, time is always one of the priorities in all areas. In particularly, when participating in traffic, how to know fastest route in your journey is the critical condition for user.

Presently, most of an application on market is not support searching through more than two points. For example, Google Map and BusMap just supports on searching through two points at most so that they cannot help user if user has more than one place to go. Moreover, no mobile application supports routing when participating traffic, for example notify message when customer near a station, or need to turn on specific street. Last but not least, no mobile application supports wear devices, so user must look up their mobile phone when participating in traffic and this behavior makes some inconveniences such as thief, accident …

Facing above problems, our team build the application named is Smart Wear on Your Route. In our application, we allow user find route through more than two points. We also support user choose their departure time so they can choose suitable route that they can come to place on time. We support user search by using voice command. We provide notify system function allowing customer know when participating traffic. We also provide wrong route detection system when user go to wrong route. We support map offline, so that customer doesn’t need network while participating traffic. Moreover, our application supports wear devices so user can look up on their wear device when they participating traffic avoiding some above problems.

In additional, we also provide system software on website for staff to manage bus route, bus time information and approve the change from background handler.

## Current Situation

Nowadays, when participating in traffic, user often wants to find route through some locations. This situation becomes more important especially participating by bus or motorbike. Currently, mobile market has some applications that support routing such as Google map or BusMap.

Google Map and BusMap allow user enters starting location and ending location into their cell phones (with already networked). After that, Google Map and BusMap will suggest some optimal paths. Finally, user will choose the best route suitable for their need. When user finish selected their choice, mobile application will render route on mobile screen so user can follow the route.

With Google Map, user can optional enter arrival time and departure time. By this constraint, Google Map will find suitable routes that user can start and come to place on time.

## Problem Definition

Below are disadvantages of current situation:

* BusMap doesn't support motorbike route.
* BusMap doesn’t support time constraint (arrival time, departure time) when finding route.
* Google Map and BusMap don't support searching through more than two points.
* Google Map and BusMap don’t support routing when participating traffic.
* Google Map supports map offline with some limitations: Download area is limit with 50km x 50km and will be expired after 30 days. After 30 days, user must download that area again.
* Google Map voice search has some limitations such as just can only search one place.
* BusMap and Google Map don’t support using smart watch for routing when participating traffic, just for phone. There are some disadvantages of using only smartphone to find route such as theft, inconvenience, no safety in motorbike control as well as the bus.

## Proposed Solution

Our proposed solution is to build and mobile application and android wear application named “Smart Wear on Your Route” to resolve the current situations. We also design the system to be scalable so we can extend our system for more platforms (iOS, Windows Phone) in the future and can be used for more transit protocols (train, high-speed train)

SWR system includes a web application, background process, mobile application and wear application with following functions:

### Feature functions

Web application: For Staff only.

* Manage routing: staff edits information for bus route and bus timetable.
* Notify new update data from server to staff: if official website (http://www.buyttphcm.com.vn/) has new data, background process will notify to staff and staff will decide approve this update or not.

Background process:

* Check new data periodically: Background process will check new data at 0 AM each day. If background process detects that data has been changed, background process will write new data to temporary database and notify messages for staff.

Mobile application:

* Searching bus route through from two points to four points: user inputs start point, two optional middle points and end point and optional departure time. Application will find the best bus route from start point through middle points to end point which optimize condition (shortest time, least number change route)
* Find bus route through from two points to four points with optimize: user inputs start point, two optional middle points and end point and optional departure time, then choose “optimize” option. Application will find the best bus route from start point through three points which optimize condition (shortest time, least number change route), no matter order last three points.
* Find motorcycle route through from two points to four points: user inputs start point, two optional middle points and end point and optional departure time. Application will find the best motorcycle route from start point through middle points to end point which optimize condition shortest time.
* Find motorcycle route through from two points to four points with optimize: user inputs start point, two optional middle points and end point and optional departure time, then choose “optimize” option. Application will find the best motorcycle route from start point through three points which optimize condition shortest time, no matter order last three points.
* Assist user searching using typing or voice command: User has two ways for input data: Using keyboard and typing text or using voice command.
* Routing bus route: when user chooses one bus route for participating traffic, application will start for tracking user’s location, and notify message (by vibrate, notification, sound) when user is near a bus station that need to get off station.
* Routing motorbike route: when user chooses one motorbike route for participating traffic, application will start for tracking user’s location, and notify message (by vibrate, notification, sound) when user is near a turn. Application also notify message when user goes to wrong route and recommmend suitable message when come to searched route again.
* Detect wrong motorbike route: When users go to wrong route, application will detect and notify message periodically for users know that they have gone wrong route.
* Recommend suitable motorbike direction for user when user is near searched route: when user is on wrong route and go near again searched route, application will recommend suitable direction for user.
* Map offline: By using map offline, user can use routing function without network connectivity.

Wear app:

* Bus: Application will notify for user when bus nears the station that user should to leave:  if bus in circular range of station of the route's plan, application will show the message name of the next station.
* Motorcycle:
* App will notify when user has to turn route: If user drives in circular range of next turn, application will automatically show message which should to do next and vibrate until user out of this range.
* App will notify when user goes to wrong route: If user goes to wrong route compare to original searched route, application will automatically show message that user has gone to wrong route.
* Map:
* Show your current location: show current user location on map with route user should to go (including bus or motorbike).

### Advantages and disadvantages

Advantages:

* Support optimization passes through multi points.
* Support bus route timetable so that user can easily find suitable round for their time.
* Support routing on street both on motorbike and bus.
* Support map offline so user doesn’t need network connectivity when participating traffic.
* Support on smart wear. User can look up information easier than using mobile phone. Also using smart wear decreases ability for mobile phone to be stolen.

Disadvantages:

* + Smart wear has higher price than smart phone and these screen is slightly small and hard to use for newbie.
* Application must parse data from third website so cannot update data intermediately.

## Functional Requirements

Web Component: (for staff only):

* Edit bus route and bus time information.
* Approve bus route and bus time change from background handler and write to official database.

Parser Component:

* Parse bus route information.
* Parse bus timetable information.
* Periodically, detect the change from official bus website in order to write to temporary database.

Mobile Component:

* Find the path’s optimization from two points to four points when using bus.
* Find the path’s optimization from two points to four points when using motorbike.
* Voice search for bus and motorbike.
* Routing when participating traffic both on bus and motorbike vehicle.
* Detecting wrong route and recommend again true direction.
* Sync data from mobile to wear.

Wear Component:

* Receive data from mobile.
* Notify message when user near the bus station that should to left.
* Notify next turns when user drive by motorbike.
* Auto scroll to current user’s location on map.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Huỳnh Quang Thảo | Developer | Leader | huynhquangthao@gmail.com |
| 3 | Trần Thanh Ngoan | Developer | Member | ngoanttse61125@fpt.edu.vn |
| 4 | Nguyễn Trung Nam | Developer | Member | namntse61132@fpt.edu.vn |
| 5 | Ngô Tiến Đạt | Developer | Member | datntse60980@fpt.edu.vn |

Table 2: Roles and Responsibilities

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

Official name: Smart Wear on Your Route

Vietnamese name: Hỗ trợ đi đường với thiết bị đeo tay thông minh.

Abbreviation: SWR

### Problem Abstract

The transport system has a lot of roads and bus routes. Even local people hardly know well. With this system, users must spend a lot of time to know how to get from one location to others. With the worst case scenario, user can lose their phone or cause danger to themselves because look up mobile phone frequently when participating traffic.

We provide application, which helps users find bus routes or motorcycle routes from two points to four points with optimized conditions (shortest time, walking distance, number transfers and departure time). Application supports finding route on android phone and real-time navigation on mobile and smart wear. Application also supports detect and recommend suitable direction when user is on wrong way. Application supports map offline so user doesn’t need network connectivity when participating traffic. Application also supports voice speech command.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* **Depend on other system**: all the crawl data is gotten from other system. So if data has been changed, our system will be out of update.
* **Network Connection:** mobile application must be connected to 3G for getting GPS and map information periodically. Moreover, wear device must be in range of Bluetooth because pairing with mobile device.
* **Testing:** hardly to test GPS function when switching between bus stations due to physical limitation.
* **Absent of team members:** team members get sick or unexpected problems. Working time and learning time are different among members.
* **Violence Google terms of Service:** Google API Terms of Service doesn’t allow real-time navigation or route guidance, including but not limited to turn-by-turn route guidance that is synchronized to the position of a user's sensor-enabled device.

#### The Proposed System

Based on current issues, combined with research results in HCM city traffic system, we propose a system allow user searches route on mobile phone and real-time navigation by using smart wear.

We also developed algorithm name RAPTOR algorithm[[1]](#endnote-1) for finding shortest distance between two points on graph.

We also store a local database on mobile device to allow users search bus route information offline.

Moreover, we have built a background handler for checking third-party server periodically every 0AM to always get the latest data.

Our system includes three main subsystems: an online website for staffs, a mobile application as well as watch application for Participants traffic.

##### Website

Website provided following features:

* For staffs:
* Staff edits information for bus route and bus timetable.
* Notify new update data from server to staff.
* Beside above, website system also provides an API interface for mobile, wear applications to retrieve data.

##### Background Handler

Check new data periodically at 0 AM each day.

Parse data from website.

##### Bus Routing Mobile Application

This application is used by user and does followings:

* Find bus route from two points to four points.
* Find bus route from two points to four points with optimize.
* Find motorcycle route from two points to four points.
* Find motorcycle route from two points to four points with optimize.
* Routing for motorbike route.
* Routing for bus route.
* Detect wrong way and recommend true direction for user.

##### Wear Application

Application for wear device must be paired with android phone and do following function:

* Bus: Application will notify for user when bus nears the station that user should to leave.
* Motorcycle: Application will notify when user has to turn route.
* Map: Show current user location on map with route user should to go (including bus or motorbike.

#### Boundaries of the System

Our system is working on Ho Chi Minh city transportation system.

The language for mobile application and wear application is Vietnamese. The language for staff management site is English.

Mobile devices run android 4.3 or above. Smart wears run android wear API 20 or above.

The complete product includes:

* Website application for staffs.
* Background handler.
* Mobile and wear application for users.

#### Future plans

Our current system only supports bus and motorbike route in Ho Chi Minh city. We also just optimize time travel, walking distance, number transfers and departure time. We design the system to make it easier for further development including:

* **Support more public transportation**: system will support more public transits such as train andhigh-speed train.
* **Support more conditions**: system will provide more options for user choice such as: minimum ticket price, ticket price (student, old people). Moreover, we will design system support user give both departure time and arrival time.
* **Support other cities in Viet Nam**: system will support user find routes in other big cities such as Ha Noi or Da Nang.

#### Development Environment

##### Hardware requirement

For continuous integrating server:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 8 Mbps |
| Operating System | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 1GB RAM | 3GB or more |

Table 3: Hardware requirement for continuous integrating server

For web development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 8 Mbps |
| Operating System | Window Vista, 7, 8 | Window 7, 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 4GB RAM | 6GB or more |

Table 4: Hardware requirement for web development

For mobile development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | Wi-Fi Connection 512Kbps  Bluetooth Connection 4.0 | Wi-Fi Connection 8Mbps  Bluetooth Connection 4.0 |
| Operating System | Android 4.3 | Android 5.0 |
| Hardware | GPS supported | GPS supported |
| Memory | 1 GB RAM | 2 GB or more |

Table 5: Hardware requirement for mobile development

For wear development:

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | Bluetooth Connection 4.0 | Bluetooth Connection 4.0 |
| Operating System | Android Wear API 20 | Android Wear API 22 |
| Hardware |  | GPS supported |
| Memory | 512MB RAM | 512 MB or more |

Table 6: Hardware requirement for wear development

##### Software requirement

|  |  |
| --- | --- |
| Software | Name / Version |
| Operating system | Windows 7 or above, MacOS 10.10 or above |
| Environment | JDK 1.6, Java EE 6, Android SDK minimum API 20 |
| Modeling tool | StartUML 5.0 |
| IDE | Intellij IDEA 14.1, Android Studio 1.3.1 |
| DBMS | MySQL 5.6 |
| Source control | Git 2.3.2, Source Tree 1.6.20.0 |
| Web browser | Chrome 42 or above |
| Team Collaboration | Slack 1.1.3 |
| Issues and Task Management | GitHub and ZenHub |
| Others | Microsoft Word, Microsoft Excel, Adobe Photoshop |

Table 7: Software requirement

## Project organization

### Software Process Model

This project is developed under waterfall model. We apply customized waterfall model to capable with current situation in our team. We choose this model because the following reasons:

* This project is 4 months long due to the FPT University Capstone Project timeline, which can be consider a short project.
* Based on discussing carefully before with our supervisor, the requirements of this project are stable, clear, fixed and well understood by all team members.
* This project use android and wear technology, which we have strong background knowledge and well practice skills. Moreover, google has enough tutorials, supporting for further research.



Figure 1: Waterfall model

Reference: Page 30, chapter 2, Software process model, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

We customize the waterfall model from the reference to make the process more capable with current situation of our team.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Supervisor / Project Manager | - Clarify user requirement.  - Technical support and business analysis.  - Tracking development process.  - Review document and product. |
| 2 | Huỳnh Quang Thảo | Team leader, BA, Developer, Tester | - Tracking process.  - Planning project, distribute tasks.  - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing.  - Deploy product. |
| 3 | Trần Thanh Ngoan | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |
| 4 | Nguyễn Trung Nam | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |
| 5 | Ngô Tiến Đạt | BA, Developer, Tester | - Requirement analysis.  - Database design.  - Documentation.  - GUI Design.  - Coding.  - Testing. |

Table 8: Roles and responsibilities

* 1. **Tools and Techniques**

|  |  |
| --- | --- |
| Tool / Technique | Name / version |
| Frontend | HTML, CSS, JavaScript, jQuery, Bootstrap |
| Backend | JavaEE, Servlet, JSP, Hibernate |
| Web server | Apache Tomcat 7 |
| Mobile | Android Development. |
| Wear | Android Wear Development |
| Development tool | IntelliJ IDEA 14, Android Studio 1.3.1 |
| DBMS | MySQL 5.6 |
| Source control | Git 2.3.2, Source Tree 1.6.20.0 |
| Modeling tool | StarUML 5.0 |
| Document tool | Microsoft Word 2013, Microsoft Excel 2013 |

Table 9: Tools and Techniques

## Project Management Plan

### Software development life cycle

Below are all the major tasks that need to be performed sequentially during the development of the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constrains | Risk |
| Requirements Definition | Identify and clarify system requirements. | Report No.1 Introduction. | 20 man-days | N/A | - Missing requirement.  - Project’s scope can be unclear.  - Lack of member share and understand. |
| System and Software Design | - Identify hardware and software requirements.  - Decide software architect and clarify software detail design.  - Design database. | Report No.2 Software Project Management Plan, Report No. 3 Software Requirement Specification and  Report No. 4 Software Design Description. | 50 man-days | Depend on Requirements Definition. | - Misunderstood or unclear system’s requirement.  - Lack of practical experience leading to unreasonable design. |
| Implementation and Unit Testing | - Implements all functions of system.  - Create test plan.  - Perform Unit testing. | Software package. | 120 man-days | - Base on Software Requirement Specification and Software Design Description.  - Coding try to follow coding convention. | - Member does not performs unit test.  - Lack of practical experience. |
| Integration and System Testing | - Perform integration test and system test. | Report No. 5 System Implementation & Test | 35 man-days | Implementation and Unit Testing are finished. | - Lack of testing experience leading to lack of test cases.  - Not enough time for performing test. |
| Operation and Maintenance | - Deploy the system  - Create the user’s manuals.  - Do routine maintenance activities. | Report No.6 Software User’s Manual | 15 man-days | Integration and System Testing are finished. | User’s manual may be difficult for user to understand and confuse. |

Table 10:Software development life cycle

### Phase Detail

#### Phase 1: Requirements Definition

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Identify and clarify system requirements. | Research current systems to collect requirements.  Define main and needed functions the system must include. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |

Table 11: Requirements definition

#### Phase 2: System and Software Design

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Identify hardware and software requirements. | Find out the suitable hardware and software for the system, as well as its minimum and recommended requirements. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Decide software architect and clarify software detail design. | - Define the major software components and interfaces.  - Draw core flow diagram, use case diagram, prototype …  - Group meeting to review and modify. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Design database. | - Design database for the system. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |

Table 12: System and software design

#### Phase 3: Implementation and Unit Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Implements all functions of system. | Coding all the components. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Create test plan. | Planning for testing. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Perform Unit testing | - Write Unit test cases.  - Implement Unit tests. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |

Table 13: Implementation and unit test

#### Phase 4: Integration and System Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Perform integration test and system test. | - Test groups of modules and test whole the system. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |

Table 14: Integration and system testing

#### Phase 5: Operation and Maintenance

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| Deploy the system | Deploy the system in client environment. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Create the user’s manuals. | Create a guideline to instruct users using system. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |
| Do routine maintenance activities. | Do routine maintenance activities for client system. | Huỳnh Quang Thảo  Trần Thanh Ngoan  Nguyễn Trung Nam  Ngô Tiến Đạt |

Table 15: Operation and maintenance

### Task sheet

Refer to “Task sheet” folder.

### All Meeting Minutes

Refer to “Meeting Minutes” folder.

## Coding Convention

### Java Coding Convention

This project follows “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

We use followings naming convention from the reference to capable with current situation in our team:

* Naming:
* Class names must be in Pascal case.
* Variable names must be in Camel case.
* Each Java class belongs to a single file.
* Intentions:
  + - * Use four spaces intentions.
      * Avoid lines with more than 80 characters
      * Declaration:
      * One declaration per line is recommended since it encourages commenting.
      * In absolutely no case should variables and functions be declared on the same line.
      * Do not put different types on the same line.

### Android Coding Convention

On Android Development and Wear Development, we follow guideline on:

<https://github.com/ribot/android-guidelines/blob/master/project_and_code_guidelines.md>

We use followings naming convention from the reference to capable with current situation in our team:

* Resources file names are written in lowercase\_underscore.
* Naming conventions for drawables:

|  |  |  |
| --- | --- | --- |
| Asset Type | Prefix | Example |
| Action bar | ab\_ | ab\_stacked.9.png |
| Button | btn\_ | btn\_send\_pressed.9.png |
| Dialog | dialog\_ | dialog\_top.9.png |
| Divider | divider\_ | divider\_horizontal.9.png |
| Icon | ic\_ | ic\_star.png |
| Menu | menu\_ | menu\_submenu\_bg.9.png |
| Notification | notification\_ | notification\_bg.9.png |
| Tabs | tab\_ | tab\_pressed.9.png |

Table 16:Naming conventions for drawables

* Naming conventions for icons:

|  |  |  |
| --- | --- | --- |
| Asset Type | Prefix | Example |
|  |  |  |
|  |  |  |
| Icons | ic\_ | ic\_star.png |
| Launcher icons | ic\_launcher | ic\_launcher\_calendar.png |
| Menu icons and Action Bar icons | ic\_menu | ic\_menu\_archive.png |
| Status bar icons | ic\_stat\_notify | ic\_stat\_notify\_msg.png |
| Tab icons | ic\_tab | ic\_tab\_recent.png |
| Dialog icons | ic\_dialog | ic\_dialog\_info.png |

Table 17: Naming conventions for icons

# **Software Requirement Specification**

## User Requirement Specification

### Customer requirement

Customer is user who uses mobile application and wear application and use web services for searching bus route or motorbike route. The customer can use some following functions:

* Mobile application includes:
* Search function includes:
  + - Search arbitrary location map.
    - Search bus route go through two points to four points.
    - Search bus route go through two points to four points with optimization.
    - Search motorbike route go through two points to four points.
    - Search motorbike route go through two points to four points with optimization.
  + View current location on map.
  + Routing function includes:
    - Routing for bus route.
    - Routing for motorbike route.
    - Detect wrong route and recommend true direction when participating traffic.
* Configuration
  + Download offline map.
  + Change notify distance for bus and motorbike.
  + Choose priority search option.
* Wear application includes:
* Map function includes:
  + View current location on map.
  + Navigate a bus station on map.
  + Navigate a motorbike route on map.
* Notification function includes:
  + Show notification when near the bus station user should to leave.
  + Show notification when come to other motorbike’s turn in motorbike route.
  + Show warning message when user goes to wrong route.
  + Show recommend message when user from wrong route near again original searched route.

### Staff requirement

Staff is people who works directly with system and can change information of bus route or bus timetable. Staff can use some following functions:

* Bus management function includes:
  + View bus route information.
  + View bus timetable information.
  + Edit bus route information.
  + Edit bus timetable information.
* A notification shows a new bus route or bus timetable information when data change. Notification management function includes:
  + View all system notifications.
  + View a detail notification.
  + Approve or reject all current system notifications. So all bus routes or bus timetable information will be updated or stay same respectively.
  + Approve or reject a detail notification. So that bus route or bus timetable information will be updated or stay same respectively.

## System Requirement Specification

### External Interface Requirement

#### User interface

* The user interface for mobile application and wear application uses Vietnamese language.
* The user interface for staff uses English language.
* Use consistent palette of colors between the text and the background.
* The user interface for web application displays best on 1024x768-screen size.
* The user interface for mobile application displays best on screen size larger than 4’’.
* The user interface for wear application displays best on screen size from 1.65’’.

#### Hardware Interface

* Smartphone with Wifi or 3G, GPS and Bluetooth.

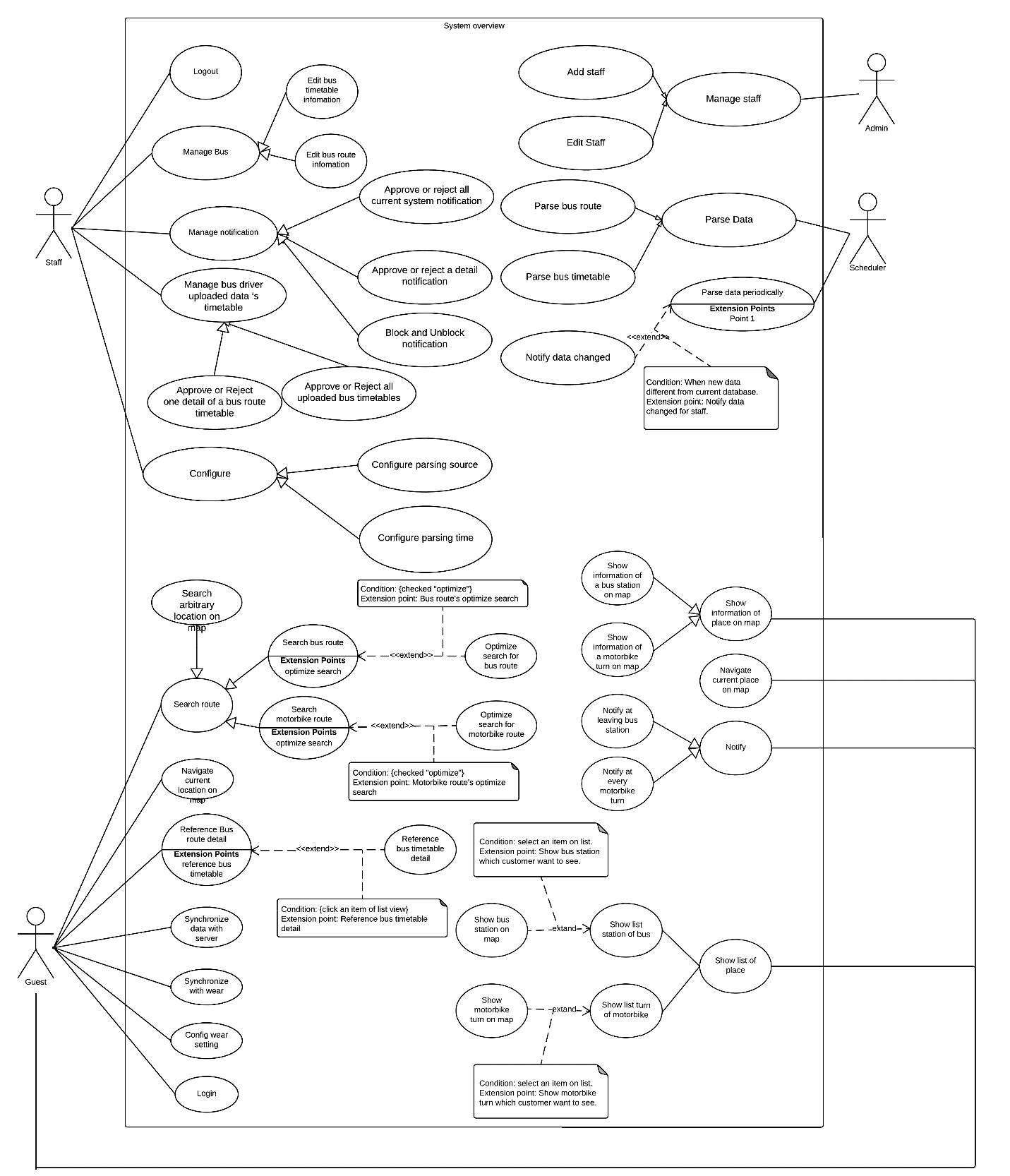
#### Software Interface

* Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet Explorer (v10 or above) browse.
* Mobile application: Android operating system (v 4.3 or above).
* Wear application: Android Wear operating system (API 20 or above)

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.
* Use HTTP protocol 1.1 for communication between the mobile application and the web service.

### System Overview Use Case



#### Web Application

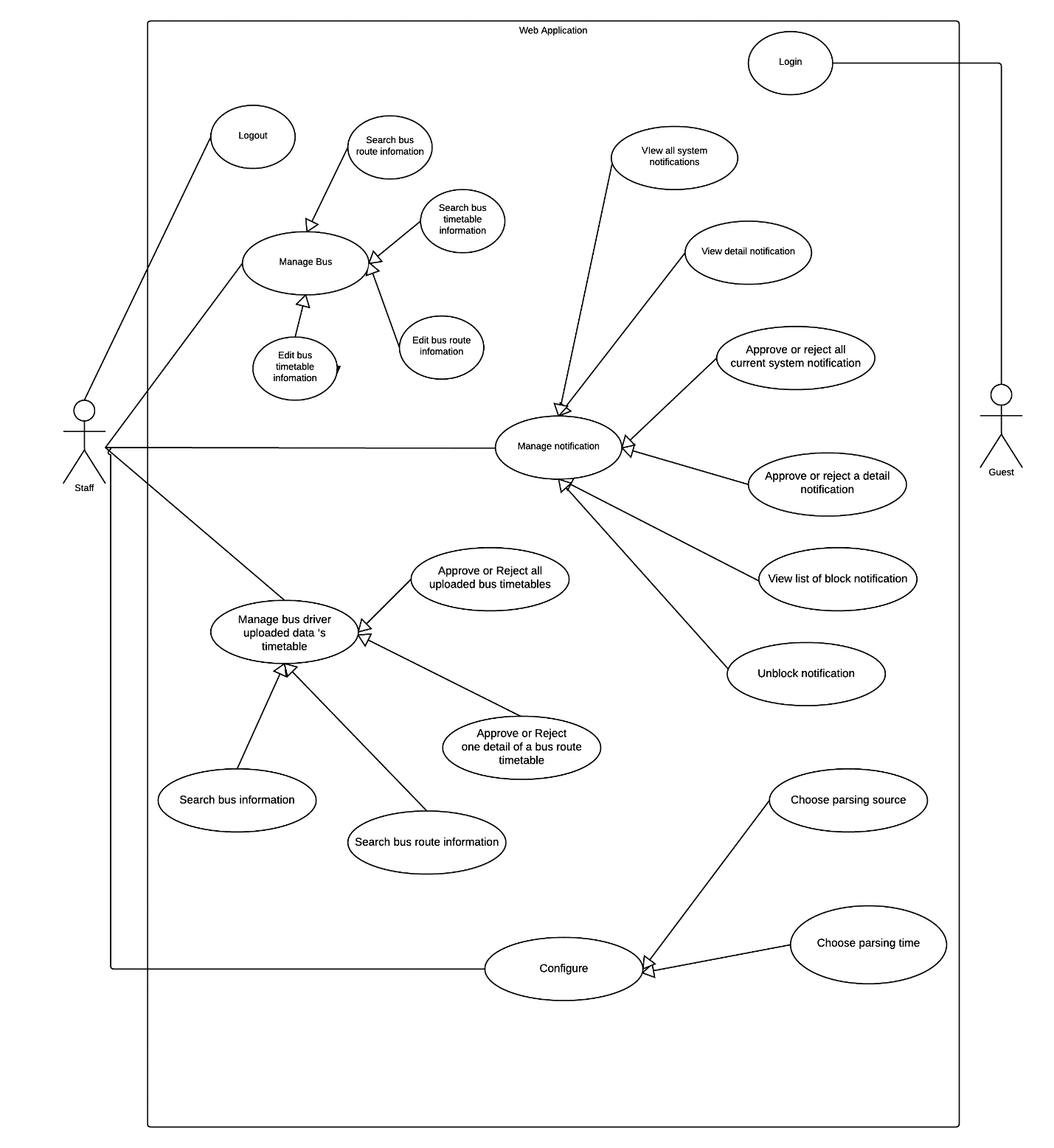


Figure 2: Web application use case

#### Android Mobile Application

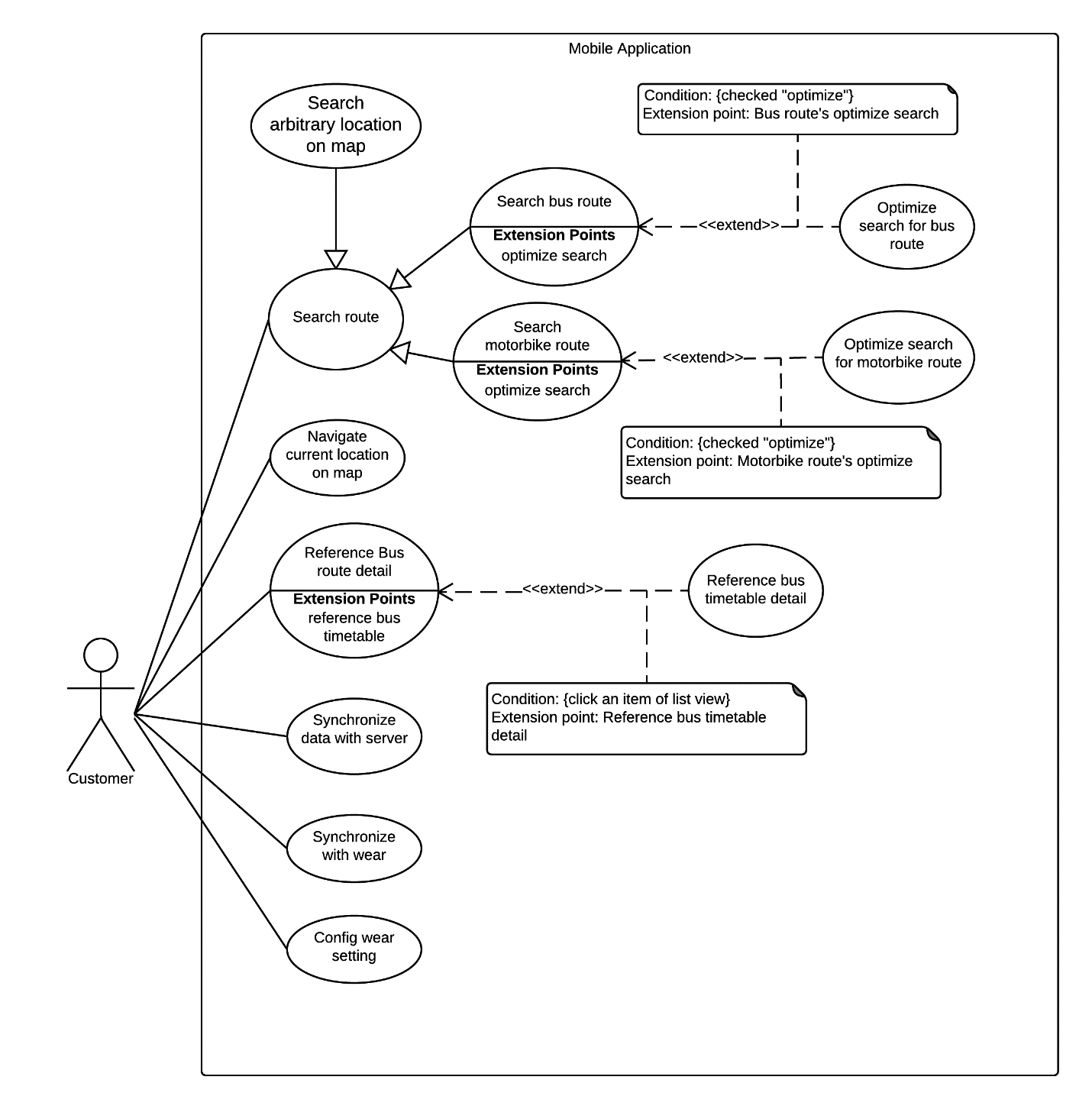


Figure 3: Android mobile application use case

#### Android Wear Application

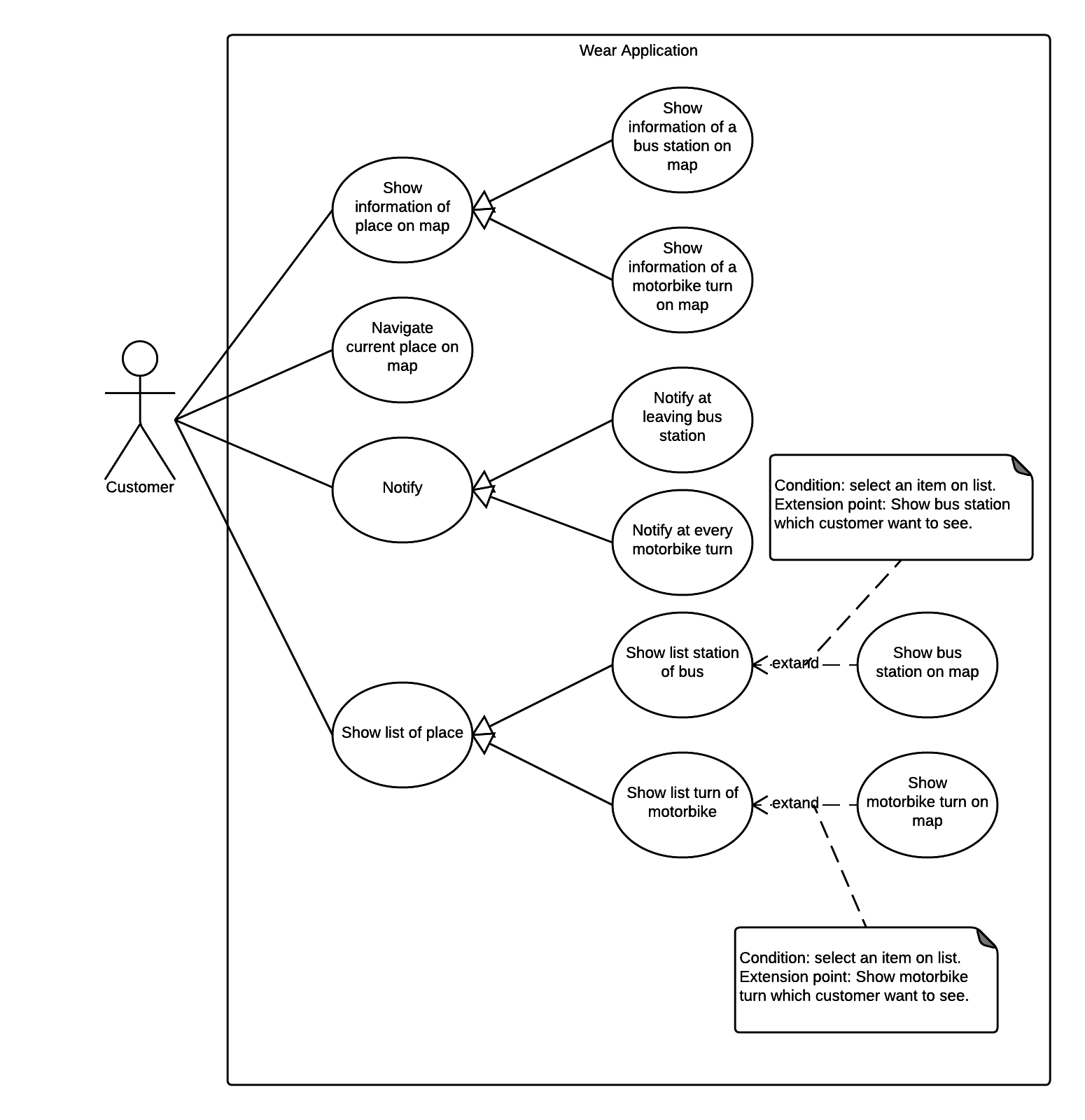


Figure 4: Android wear application use case

#### Bus Driver Mobile Application

### List of Use Case

#### Web Application

##### <Guest> Overview Use Case



Figure 5: <Guest> Overview Use Case

###### <Guest> Login



Figure 6: <Guest> Login

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WG01 | | | |
| Use Case No. | WG01 | **Use Case Version** | 2.0 |
| Use Case Name | Login | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Guest   Summary:   * This use case allows guest to log in the system and this function will be implemented on web application.   Goal:   * Guest can log in the system. Therefore, they can use some additional functions of staff.   Triggers:   * Guest sends the login command.   Preconditions:   * **N/A**   Post Conditions:   * Success**:** Guest login the system. Guest become a Staff on system. * Fail**: N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Guest goes to login view. | System requires identity information from Guest:   * Guest code: free text input. * Password: free text input. | | **2** | Guest inputs information. |  | | **3** | Guest sends login command to system. | If guest login successfully, guest will login system with login account role.  If login fail, system show error message “Invalid username or password”. |   Alternative Scenario: **N/A**  Exceptions: **N/A**  Relationships: **N/A**  Business Rules:   * Password field displayed “\*”. * After guest login to system successfully, guest will have redirected to specified view based on their role on the system: staff or admin. * If role is “Admin”, the system will display to Admin view. * If role is “Staff”, the system will display to Staff Dashboard view | | | |

Table 18: Use case WG01 - <Guest> Login

##### <Staff> Overview Use Case

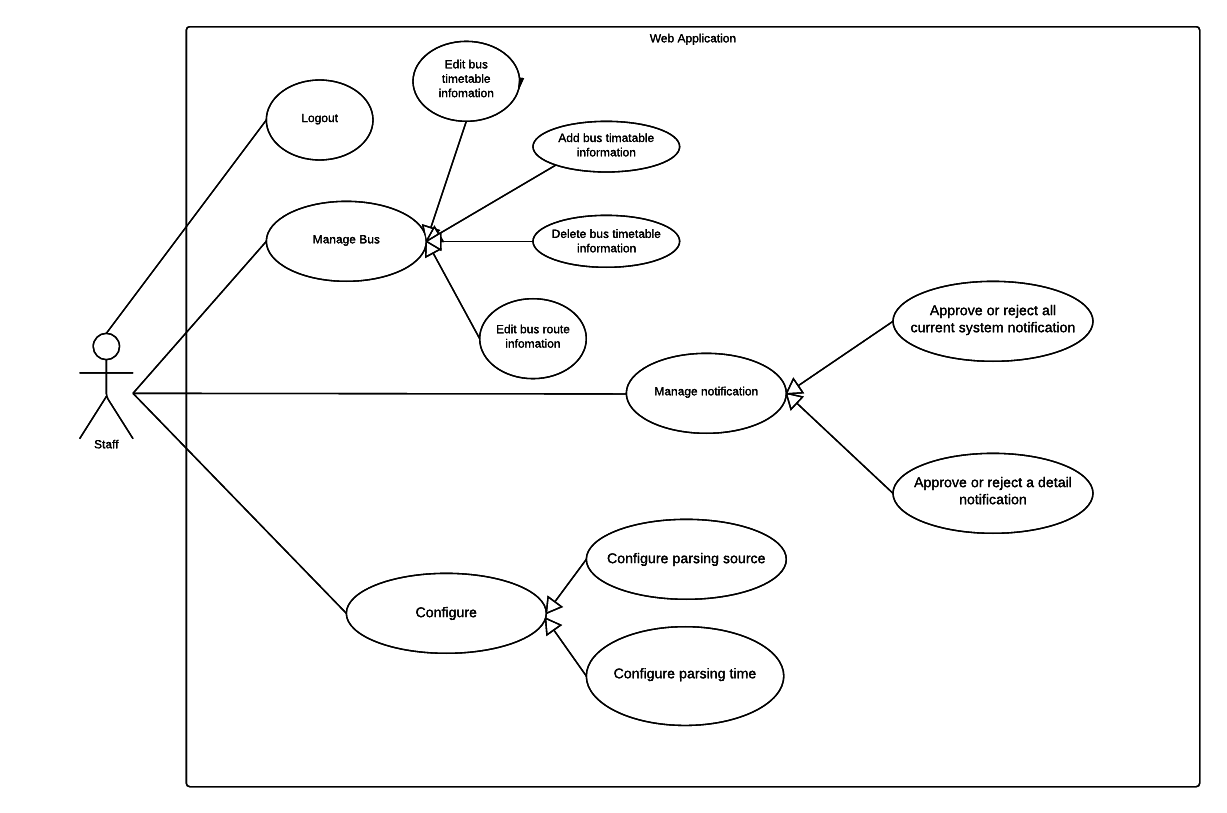


Figure 7 <Staff> Overview Use Case

###### <Staff> Logout

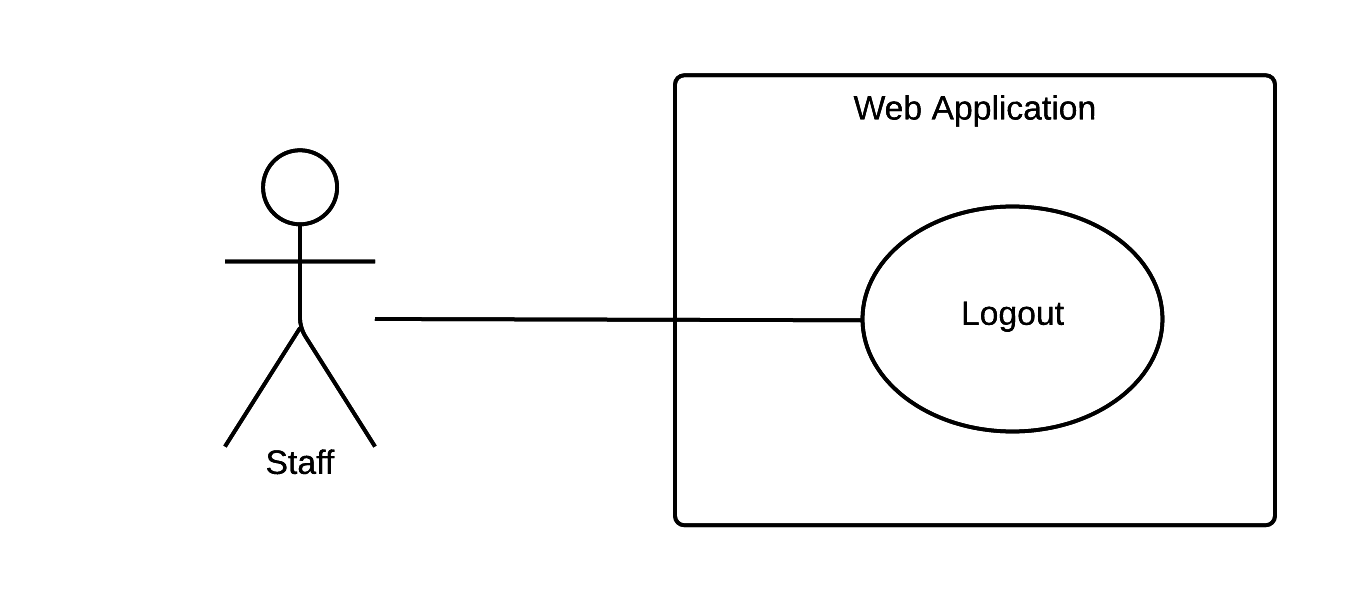


Figure 8 <Staff> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS01 | | | |
| Use Case No. | WS01 | **Use Case Version** | 2.0 |
| Use Case Name | Logout | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Staff   Summary:   * This use case allows staff logout the system and this function will implement on web application.   Goal:   * System provides logout for support staff exits from system and redirect staff to login view. After that, not all functions of staff will validate on this machine.   Triggers:   * Staff sends logout command.   Preconditions:   * Staff must login to system. * Session must be existed.   Post Conditions:   * Success**:** Staff logs out the system**.** * Fail**:** Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff sends logout command. | Staff exits from system.  [Exception 1] |   Alternative Scenario: **N/A**  Exceptions:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | Session timeout. | System signs out automatically then navigate to login page because of long inactivity. |   Relationships: **N/A**  Business Rules:   * If user is inactive in 30 minutes, session will be expired and system will log staff out automatically. * When staff logged out from system, system will redirect staff to login view**.** | | | |

Table 19: Use case WS01 - <Staff> Logout

###### <Staff> Reject all current system notifications

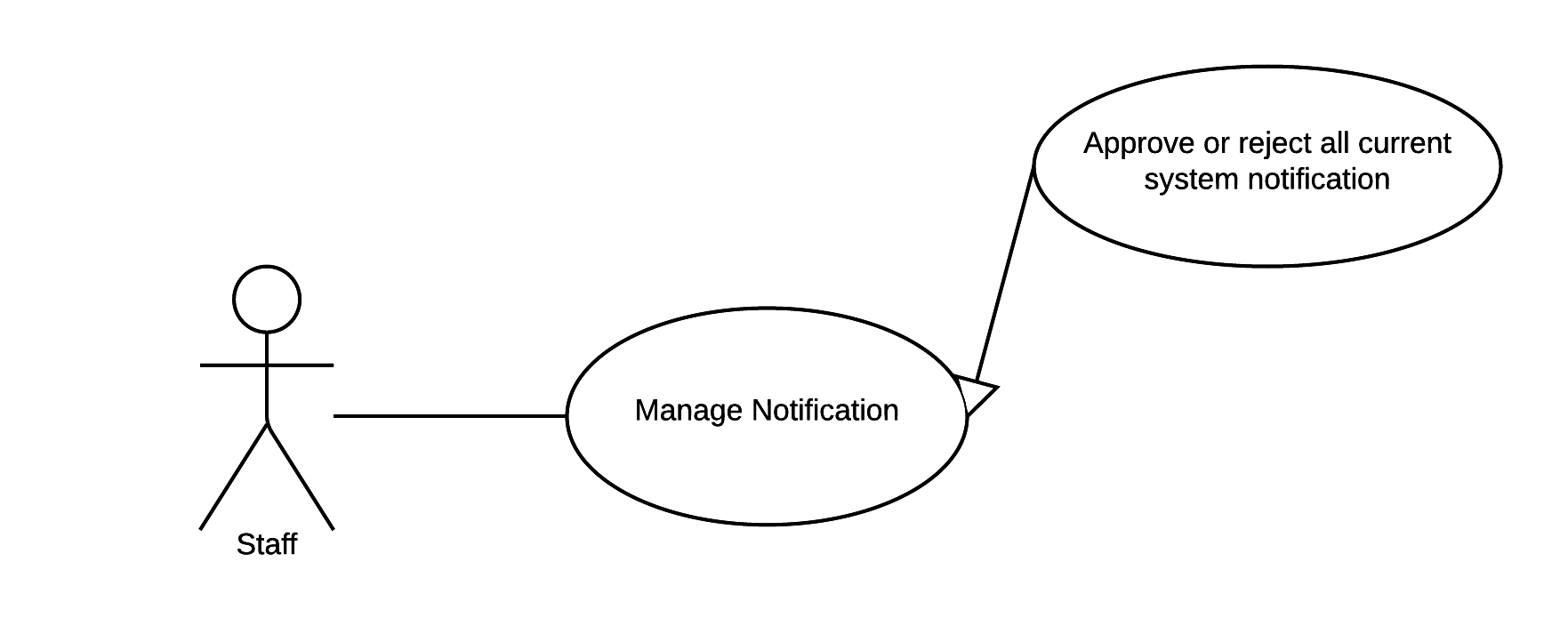


Figure 25 <Staff> Approve or reject all current system notifications

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS02 | | | |
| Use Case No. | WS02 | **Use Case Version** | 2.0 |
| Use Case Name | Approve or reject all current system notifications | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff**.**   Summary:   * This use case allows staff to approve or reject all current system notifications, and this function will implement on web application.   Goal:   * System provides approve or reject all current system notifications for supporting staff manages system notifications.   Triggers:   * Staff sends approve or reject all current system notifications command.   Preconditions:   * Staff must login to the system with role Staff.   Post Conditions:   * Success**:** Data updated. Show successful message to staff. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to all system notifications view. | System will display all current system notifications view following information:   * List of notifications will show on block and each notification includes:   + Content message – link to view detail notification.   + Action:     - Approve notification.     - Reject notification.   + Time of arrived notification. | | **2** | Staff sends approve all commands.  [Alternative 1] | * System will approve all system notifications. * System updates new data to storage. * System show successful message to staff. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff sends reject all command. | * All notifications will be removed, and current data in system doesn’t change. |   Exceptions: **N/A.**  Relationships: **N/A.**  Business Rules:   * When staff approves all system notifications, system will update data from each notification:   + If notification notify changes of timetable, system will update data of timetable.   + If notification notify changes of bus route, system will update data of bus route. * Staff will receive a successful message when system done update:   + Content: “New data updated”.   + This message will automatically close in two seconds. * Notification structure includes: * Date created. * Bus route or bus time change. | | | |

Table 20: <Staff> Approve or reject all current system notifications.

###### <Staff> Approve or reject notification



Figure 25 <Staff> Approve or reject all current system notifications

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS03 | **Use Case Version** | 2.0 |
| Use Case Name | Approve or reject notification | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to approve or reject notification, and this function will implement on web application.   Goal:   * System provides approve or reject notification for supporting staff manage notification.   Triggers:   * Staff sends approve or reject command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success**:** Detail of notification will have approved or rejected. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to detail notification view. | Display detail notification view with following information:   * Route no. * Route name. * Route type: depart or return. * List of detail notifications on table and each row includes**:**   + Index.   + Notification detail.   + Time of arrived notification.   + Action:     - Approve notification.     - Reject notification. | | **2** | Staff sends approve command.  [Alternative 1] | * System will approve notification. * System updates new data to storage. * System shows successful message to staff. |   Alternative Scenario:  **[Alternative 1]**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff sends reject command. | This notification will be removed. After that, system will show message “Reject notification” to customer. |   Exceptions: **N/A.**  Relationships: **N/A.**  Business Rules:   * When staff approves detail notification, system will update data from notification:   + If notification notify changes of timetable, system will update data of timetable.   + If notification notify changes of bus route, system will update data of bus route. * Staff will receive a successful message when system done update:   + Message places on left of top of list system notification.   + Content of message is “New data updated”   + Message will automatically close in two seconds. * Notification structure includes: * Date created. * Bus route or bus time change. | | | |

Table 20: <Staff> Approve or reject all current system notifications.

###### <Staff> Configure parsing source

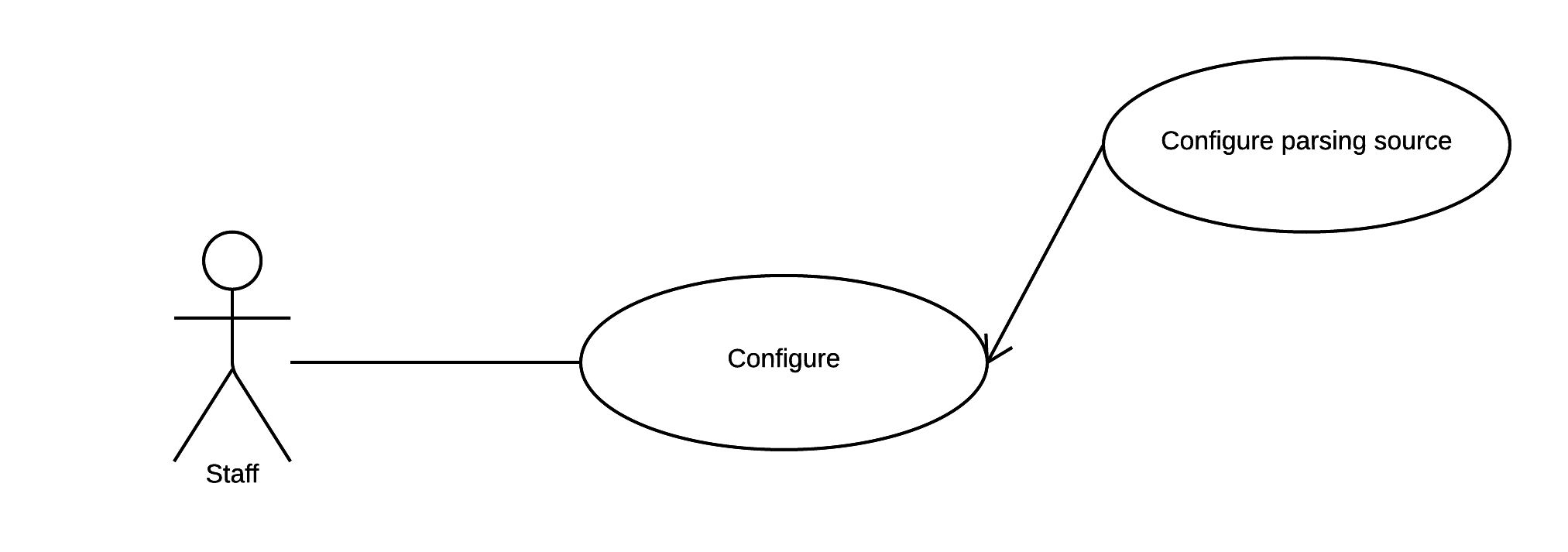


Figure 11: <Staff> Configure parsing source

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS05 | **Use Case Version** | 2.0 |
| Use Case Name | Configure parsing source. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to choose parsing source and this function will implement on web application.   Goal:   * The system provides configure parsing source for parser knows which source to parse.   Triggers:   * Staff sends configure parsing source command.   Preconditions:   * Staff must login into the system with role Staff. * Staff must specific source for parsing.   Post Conditions:   * Success: System call scheduler for parsing data. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends configure parsing source command. | System have two option for parsing:   * If staff chooses parsing from file:   + Staff selects file from local computer.   + If selected files for parsing bus route: reference use case: <Scheduler> Parse bus route   + If selected file for parsing bus timetable: reference use case: <Scheduler> Parse bus timetable.   [Alternative 1]   * If staff chooses parsing from web:   + System request parsed link: <http://www.buyttphcm.com.vn/TTLT.aspx> (link 1) and   <http://mapbus.ebms.vn/> (link 2).   * + Within parsed link, system fetches data by using JSoup and HTMLUnit.   + System call scheduler:     - Within link 1, system call parse bus timetable (reference use case <Scheduler> Parse bus timetable)     - With link 2, system call parse bus route (reference use case <Scheduler> Parse bus route) |   Alternative Scenario:  [Alternative 1]   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | No file for parsing. | Show message to notify staff must select file before configure parsing source. | | 2 | File is inappropriate with required file from system. | Show message to notify staff select incorrect file for parsing. |   Exceptions: N/A  Relationships: N/A.  Business Rules:   * In presently, system just supports for map of Ho Chi Minh bus. * Local file for parsing must correct following template file:   C:\Users\datnt\Desktop\excelTemplate.JPG   * + System currently just support excel file format.   + File name must follow format: C:\Users\datnt\Desktop\fileName.JPG (9: route no, depart: route type).   + Successful message Content: “Parse bus timetable successful”. * Error message: Content: “Parse bus timetable failed. Please make sure excel file is correct with template Template.xls”. | | | |

Table 23: <Staff> Configure parsing source.

###### <Staff> Edit bus timetable route information

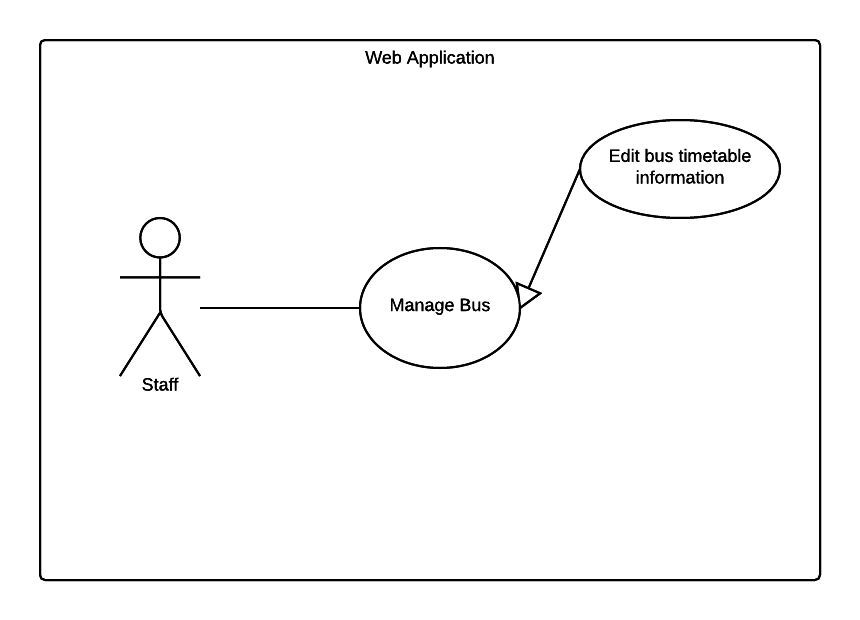


Figure 12: <Staff> Edit bus timetable information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS06 | **Use Case Version** | 2.0 |
| Use Case Name | Edit bus timetable information. | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to edit bus timetable information and this function will implement on web application.   Goal:   * System provides edit bus timetable information for supporting staff can edit bus timetable.   Triggers:   * Staff sends edit bus timetable information command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success**:** Bus timetable information will be updated. * Fail: **Show error message.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to bus timetable edit view. | System displays view with following information:   * Route no. * Route name. * Route type. * Table with following information:   + Field start time.   + Field end time.   + Link edit bus timetable. | | **2** | Staffs fill information. |  | | **3** | Staff sends edit bus timetable command. | Validate start time and end time.  [Exception 1]  Update bus timetable’s new time to system.  Reload table bus timetable information. |   Alternative Scenario: **N/A.**  Exceptions:  **[Exception 1]**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields. |   Relationships: **N/A.**  Business Rules:   * Start time and end time in format 24h. | | | |

Table 24: <Staff> Edit bus timetable route information.

###### <Staff> Add bus timetable route information

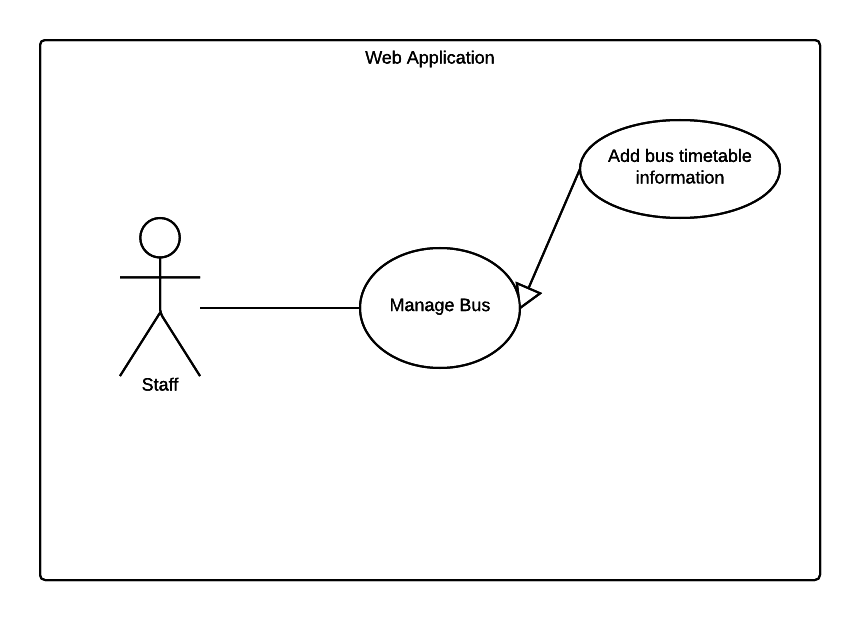


Figure 12: <Staff> Add bus timetable information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS06 | **Use Case Version** | 2.0 |
| Use Case Name | Add bus timetable information. | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to add bus timetable information and this function will implement on web application.   Goal:   * System provides add bus timetable information for supporting staff can add bus timetable.   Triggers:   * Staff sends add bus timetable information command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: New bus timetable is added into storage. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to bus timetable add view. | System display view with following information:   * Route no. * Route name. * Route type. * Button add bus timetable. * Table of bus timetable information with following information:   + Start time field.   + End time field. | | **2** | Staff fills information. | Information will be displayed on free text input field. | | **3** | Staff sends add bus timetable command. | Validate start time and end time.  [Exception 1]  Add bus timetable’s new time to storage.  Show success message. |   Alternative Scenario: **N/A.**  Exceptions:  **[Exception 1]**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Missing of required fields | Show message notify staff input missed fields. |   Relationships: **N/A.**  Business Rules:   * Start time and end time in format 24h. | | | |

Table 24: <Staff> Add bus timetable route information.

###### <Staff> Delete bus timetable route information

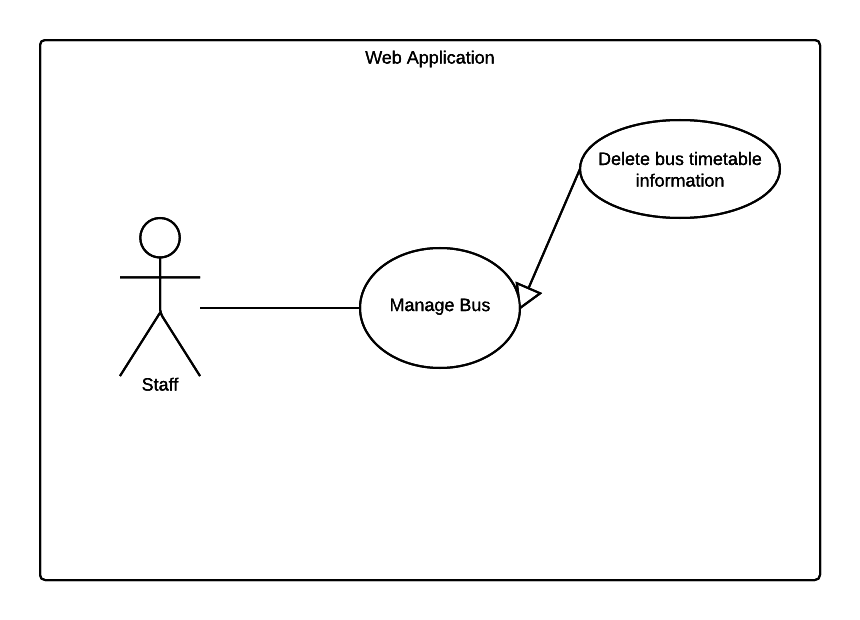


Figure 12: <Staff> Delete bus timetable information

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Delete bus timetable information. | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to delete bus timetable information and this function will implement on web application.   Goal:   * System provides delete bus timetable information for support staff can delete bus timetable information.   Triggers:   * Staff send delete bus timetable information command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: Delete selected bus timetable information. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to bus timetable delete view. | System display view with following information:   * Route no. * Route name. * Route type. * Table with following information:   + Field start time.   + Field end time.   + Link delete bus timetable. | | **2** | Staff sends edit bus timetable command. | System delete selected bus timetable information.  Show success message. |   Alternative Scenario: **N/A.**  Exceptions: **N/A.**  Relationships: **N/A.**  Business Rules:   * Bus timetable must have at least one in the system. | | | |

*Table 24: <Staff> Delete bus timetable route information*

###### <Staff> Edit bus route information

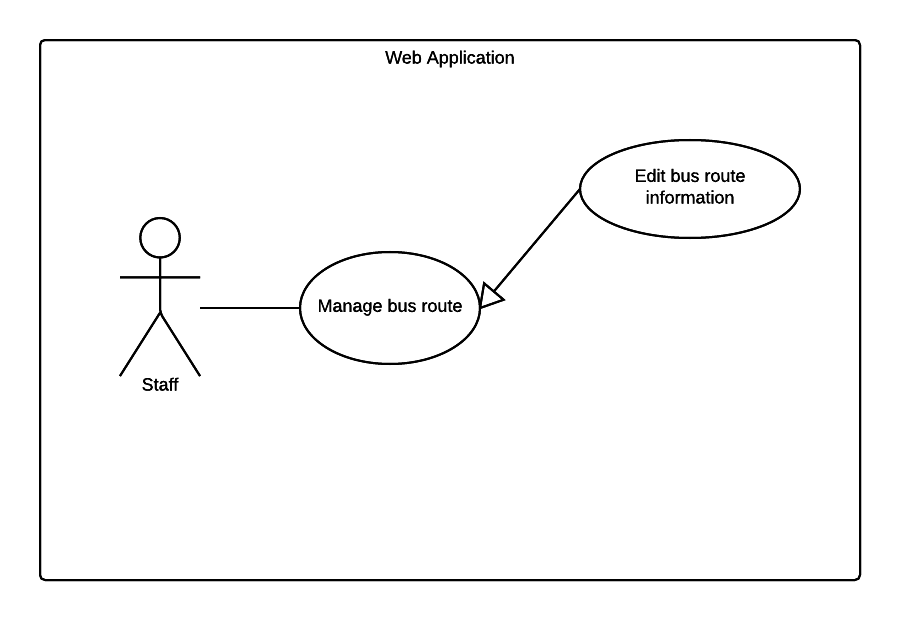


Figure 12: <Staff>Edit bus route information.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Edit bus route information | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allow staff to edit bus route information and this function will implement on web application.   Goal:   * System support edit bus route information for support staff can edit route information.   Triggers:   * Staff send edit bus route information command.   Preconditions:   * Staff must login with role staff.   Post Conditions:   * Success: Edited bus route information will be updated into storage. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to route management view. | System display table view with following information:   * Route no * Route name. * Link edit route * Link delete route. * Search box. * Navigate component. | | **2** | Staff sends edit bus route information command. | System update edited bus route information into storage.  Show successful message. |   Alternative Scenario: **N/A.**  Exceptions: **N/A.**  Relationships: **N/A.**  Business Rules:   * Bus timetable must have at least one in the system. | | | |

*Table 24: <Staff> Edit bus route information.*

###### 2.3.1.2.9. <Staff> Delete bus route information

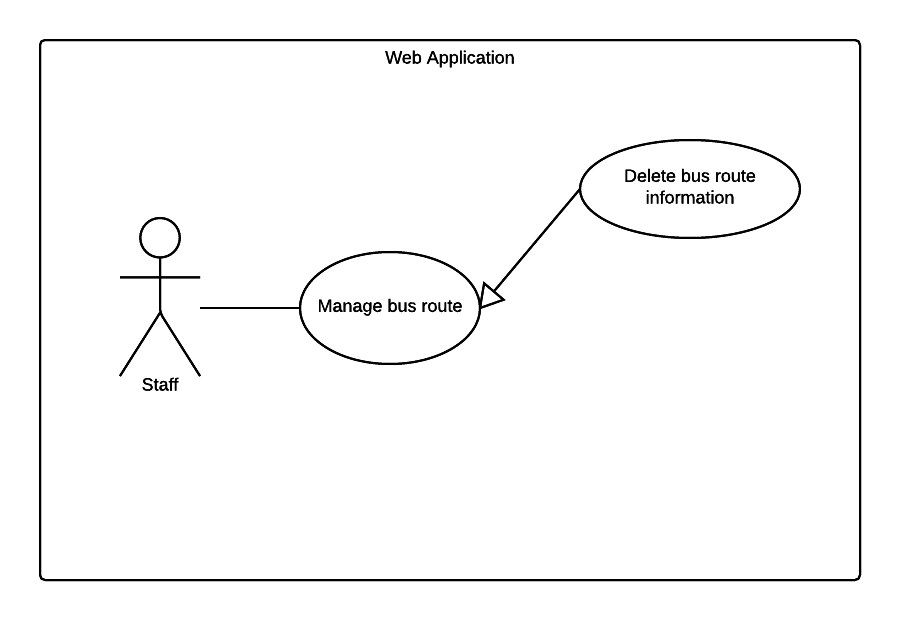


Figure 12: <Staff> Delete bus route information.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Delete bus route information | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case will allow staff to delete bus route information and this function will implement on web application.   Goal:   * System provides delete bus route information for support staff delete can delete bus route information.   Triggers:   * Staff sends delete bus route information command.   Preconditions:   * Staff must login with role staff.   Post Conditions:   * Success: Selected route will be removed in storage. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to route management view. | System display table view with following information:   * Route no. * Route name. * Link edit route * Link delete route. * Search box. * Navigate component. | | **2** | Staff sends delete bus route information. | System will delete selected route in storage.  Show successful message. |   Alternative Scenario: **N/A.**  Exceptions: **N/A.**  Relationships: **N/A.**  Business Rules:   * Route for deleting must be existed in storage. | | | |

*Table 24: <Staff> Delete bus route information.*

###### <Staff> Add bus route information

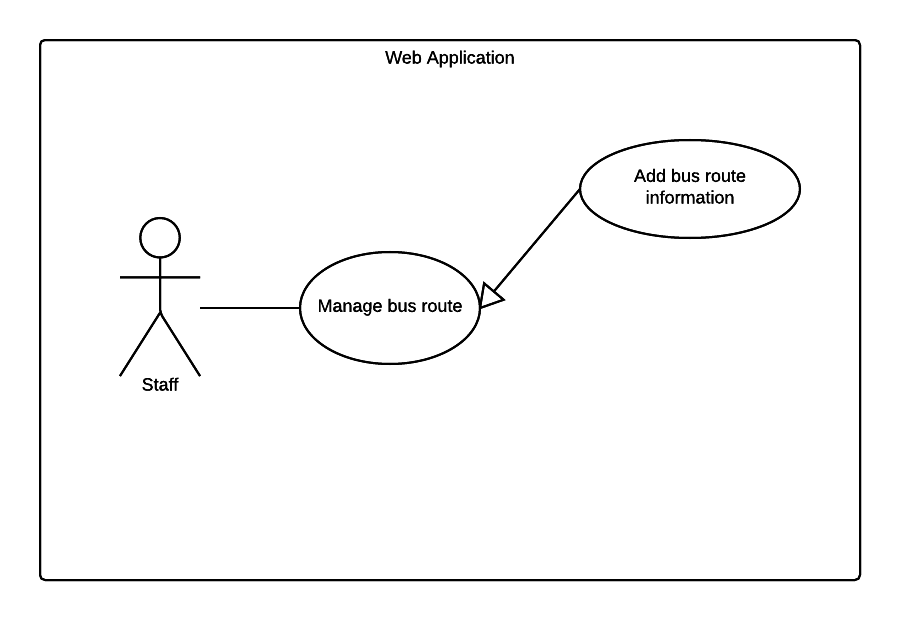


Figure 12: <Staff> Add bus route information.

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Add bus route information. | | |
| Author | DatNT | | |
| Date | 22/11/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case will allow staff add bus route information and this function will implement on web application.   Goal:   * System provides add bus route information for support staff can add route information.   Triggers:   * Staff sends add route information command.   Preconditions:   * Staff must login with role staff.   Post Conditions:   * Success: Selected route will be removed in storage. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Staff goes to route management view. | System display table view with following information:   * Route no. * Route name. * Link edit route * Link delete route. * Button add route. * Search box. * Navigate component. | | **2** | Staff sends add route information command. | System will redirect staff to add route view. | | **3** | Staff fill information and choose file for adding. | System will display typing information on free text input field. | | **4** | Staff sends add command. | With selected json file, system calls scheduler for parse json file (reference use case: <Scheduler> Parse bus route).  With selected excel file, system call scheduler for parse excel file (reference use case: <Scheduler> Parse bus timetable>) [Alternative 1], [Exception 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | No file for parsing. | Show message to notify staff must select file before configure parsing source. | | 2 | File is inappropriate with required file from system. | Show message to notify staff select incorrect file for parsing. |   Exceptions:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | Format file doesn’t compatible with current system file template. | * System terminates process parsing. * System sends fail message to staff and notify “incompatible file format” |   Relationships: **N/A.**  Business Rules:   * System just support add route from json file and excel file. | | | |

*Table 24: <Staff> Add bus route information.*

##### <Admin> Overview Use Case

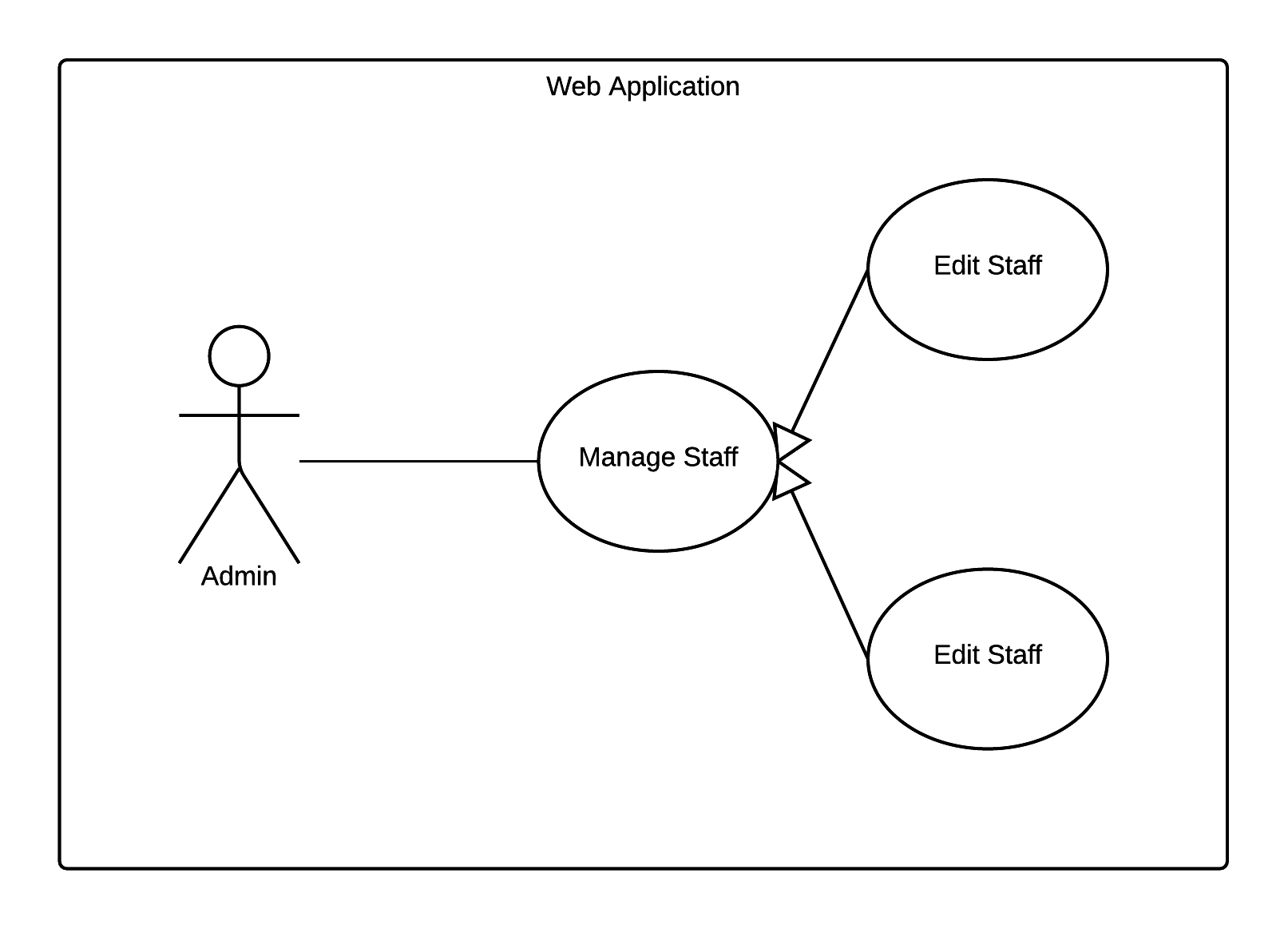
. 

Figure 13: <Admin> Admin Overview Use Case

###### <Admin> Add staff

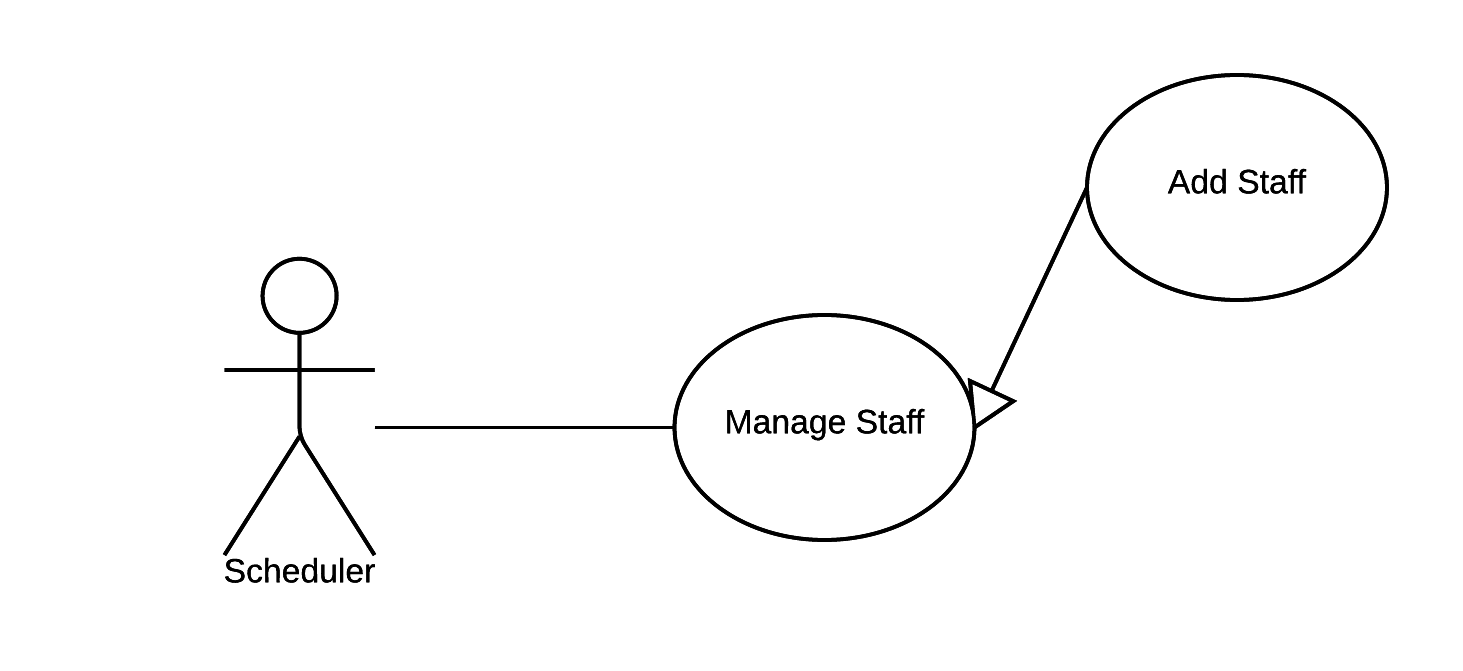
**

Figure 14: <Admin> Add staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS07 | | | |
| Use Case No. | WS07 | **Use Case Version** | 2.0 |
| Use Case Name | Add staff. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Admin.   Summary:   * This use case allows admin add staff and this function will implement on web application.   Goal:   * The system provides add staff for creating new staff for manage bus system.   Triggers:   * Admin sends add staff command.   Preconditions:   * Admin must login to system.   Post Conditions:   * Success: New staff created. Show success message. * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin goes to add staff view. | System displays view with following information:   * Staff’s full name: free text input, required. * Staff’s username: free text input, required. * Staff’s password: password input, required. * Staff’s email address: free text input, required. | | **2** | Admin fills out the form. | * After customer finish one field, system will show error message (if need) next to that field. Therefore, admin knows which field has typed wrong and need to input again. | | **3** | Admin sends create staff command. | * Validate data.   [Exception 1]   * If data valid, system creates new staff. * System shows successful message to admin.   [Alternative 1] |   Alternatives Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | Staff cannot be inserted. | System will show message that this new staff cannot be inserted. |   Exceptions:   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Missing required field. | Show message to notify admin, which required fields are missing. | | 2 | Field’s value have length out of specified range. | Show message to notify staff which field’s value is out of range. | | 3 | Email address is incorrect with format email. | Show message to notify staff an email is not valid. | | 4 | The inputted email address has existed on storage. | Show message to notify staff an inputted email address has existed. |   Relationships: **N/A.**  Business Rules:   * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/   * An email must be unique among staff. No two staffs share same email address. * Password must be encrypted before save to database. | | | |

Table 25: <Staff> Add staff.

###### <Admin> Edit staff

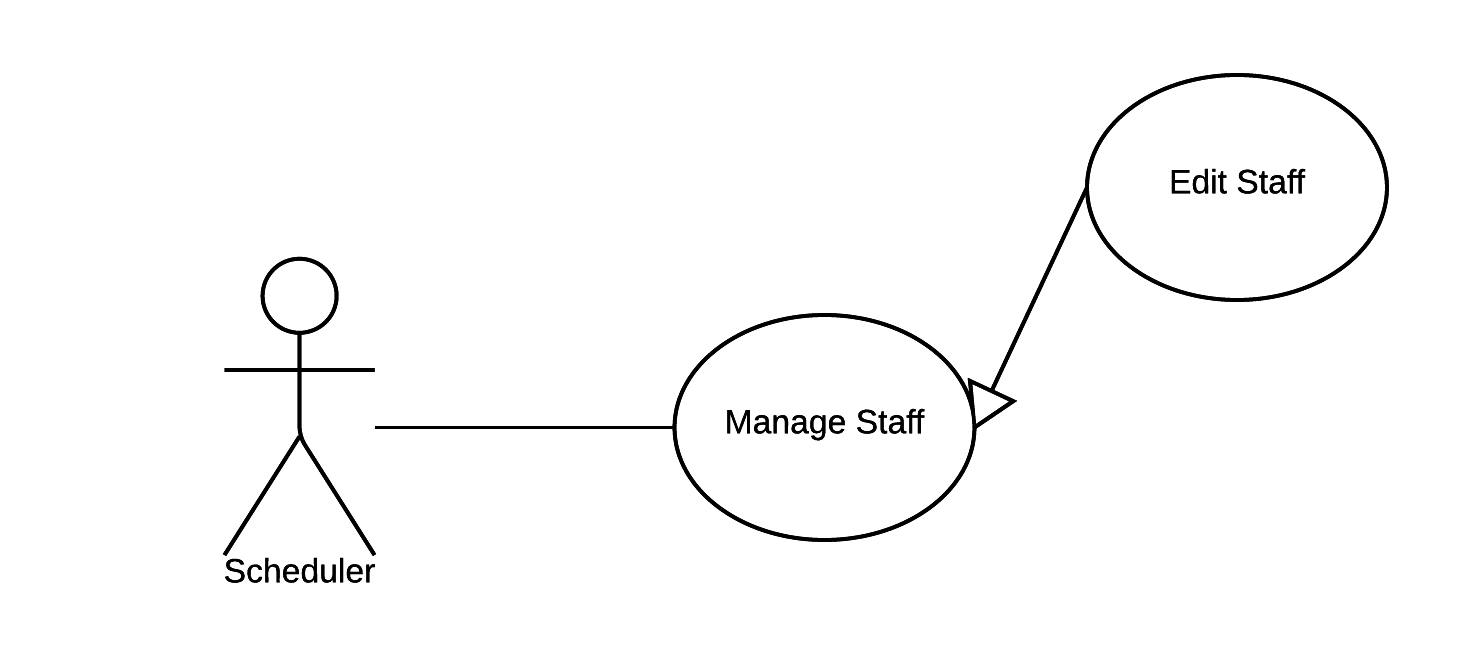


Figure 15: <Admin> Edit staff

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS08 | | | |
| Use Case No. | WS08 | **Use Case Version** | 2.0 |
| Use Case Name | Add staff. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Admin.   Summary:   * This use case allows admin adds staff and this function will implement on web application.   Goal:   * The system provides edit staff to help admin can edit staff information such as change password when staff forget their password or other related information.   Triggers:   * Admin sends edit staff command.   Preconditions:   * Admin must login to system.   Post Conditions:   * Success**:** Staff’s information updated. Show success message**.** * Fail: **N/A.**   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin goes to edit customer information view. | System display view with following information:   * Staff’s full name. * Staff’s username. * Staff’s email address. * Staff’s phone number. * Staff’s role. | | **2** | Admin selects free text field. | This field will be changed to editable mode. | | **3** | Admin inputs value. | After customer finish one field, system will show error message (if need) next to that field. Therefore, admin knows which field has typed wrong and need to input again. | | **3** | Admin sends edit staff command. | * Validate data.   [Exception 1]   * If data valid, system update new staff’s information to storage. * System show successful message to admin.   [Alternative 1] |   Alternatives Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | Staff cannot be updated. | System will show message that this new staff cannot be updated. |   Exceptions:  **[Exception 1]**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Field’s value have length out of specified range. | Show message to notify staff which field’s value is out of range | | 2 | Email address is incorrect with format email. | Show message to notify staff an email is not valid. | | 3 | The inputted email address has existed on storage. | Show message to notify staff an inputted email address has existed. |   Relationships: **N/A.**  Business Rules:   * An email address must be validated by this regular expression:   /^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/   * Password must be encrypted before save to database. | | | |

Table 26: <Staff> Add staff.

###### <Admin> Logout

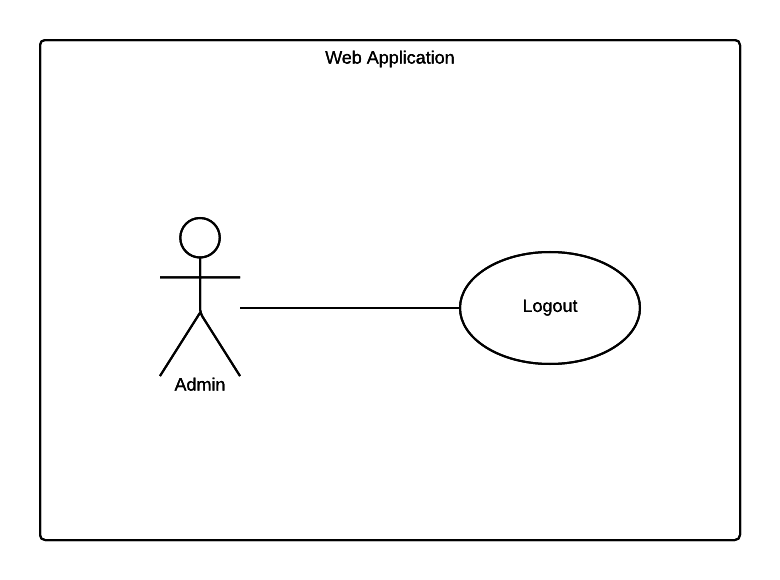


Figure 8 <Admin> Logout

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS01 | | | |
| Use Case No. | WS01 | **Use Case Version** | 2.0 |
| Use Case Name | Logout | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Admin   Summary:   * This use case allows admin logout the system and this function will implement on web application.   Goal:   * System provides logout for support admin exits from system and redirect admin to login view. After that, all functions of admin will not available on this machine.   Triggers:   * Admin sends logout command.   Preconditions:   * Admin must login to system. * Session must be existed.   Post Conditions:   * Success**:** Admin logs out the system**.** * Fail**:** Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Admin sends logout command. | Admin exits from system.  [Exception 1] |   Alternative Scenario: **N/A**  Exceptions:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | Session timeout. | System signs out automatically then navigate to login page because of long inactivity. |   Relationships: **N/A**  Business Rules:   * If user is inactive in 30 minutes, session will be expired and system will log admin out automatically. * When admin logged out from system, system will redirect admin to login view**.** | | | |

Table 19: <Admin> Logout

#### Scheduler

##### <Scheduler> Overview Use Case

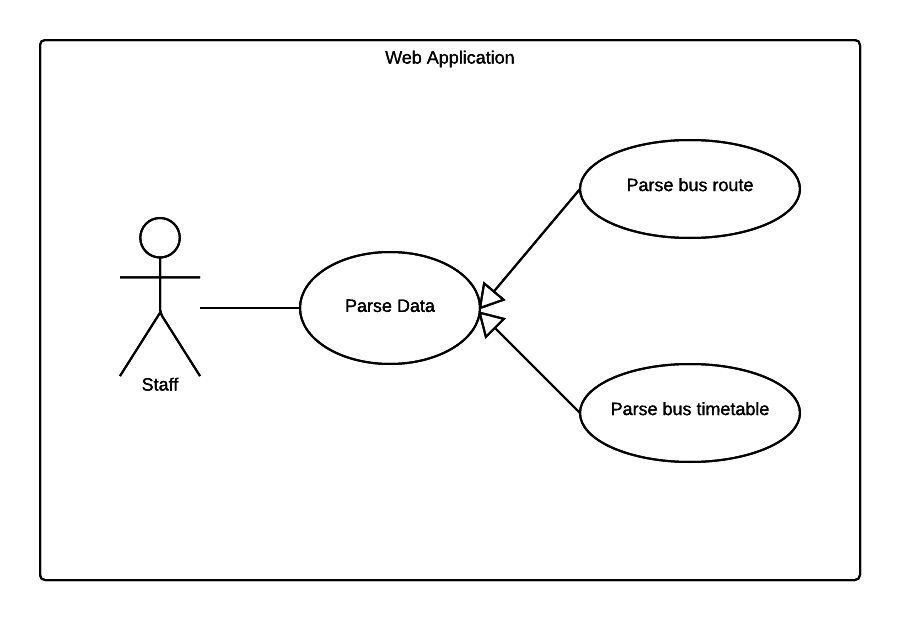


Figure 16: <Scheduler> Overview Use Case

###### <Scheduler> Parse bus route

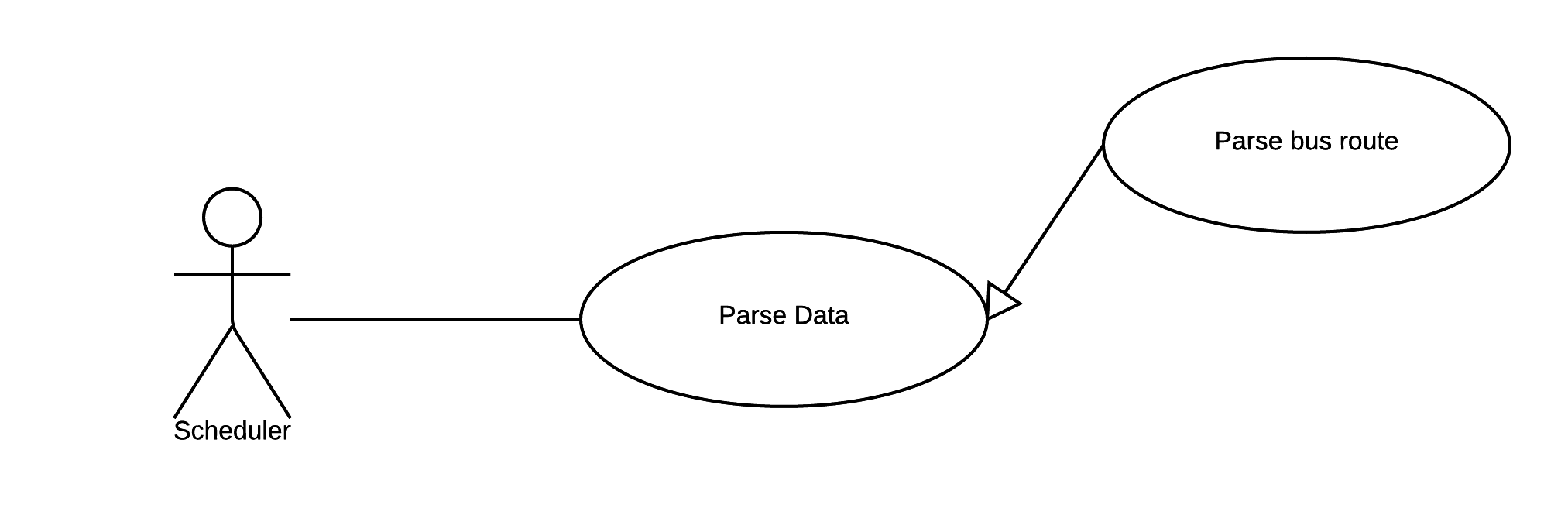


Figure 17: <Staff> Parse bus route

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – SC01 | | | |
| Use Case No. | SC01 | **Use Case Version** | 2.0 |
| Use Case Name | Parse bus route. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Scheduler.   Summary:   * This use case allows scheduler to parse bus route.   Goal:   * The system provides parses bus route for converting online source to own database for reuse later when third-party server is disconnected.   Triggers:   * Scheduler sends parsing source command.   Preconditions:   * Staff must configure parsing source command.   Post Conditions:   * Success: New data inserted to storage. * Fail: Nothing changed in storage and show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Scheduler sends parse bus route command. | * Scheduler has two ways for getting data: from local file or from third-party service. Base on configure setting (reference use case <Staff> configure parsing source), scheduler will choose method for getting data. * Validate Data [Exception 1] * If data is valid, system will insert data into storage. [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | System insert data to storage unsuccessfully. | System sends unsuccessful message to staff. |   Exceptions:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | format file doesn’t compatible with current system file template. | * System terminates process parsing. * System sends fail message to staff and notify “incompatible file format” |   Relationships: **N/A.**  Business Rules:   * The format file template file for suggestion can download from URL:   [http://mapbus.ebms.vn/ajax.aspx?action=listRouteStations&id={busId}&isgo={true|false}](http://mapbus.ebms.vn/ajax.aspx?action=listRouteStations&id=%7bbusId%7d&isgo=%7btrue|false%7d)   * busId parameter: id of bus route. Can get from link <http://map.ebms.vn/> * isGo parameter: decide to get depart or return of one bus route. If isGo is true, server will return “depart route”. Otherwise, serve will return “return route”. * Currently system supports JSON format for template file. * Local file for parsing must correct following template file:      * Field 1 is route no. * Field 2 currently is not use. * Field 3 currently is not use. * Field 4 is middle point in path. * If field 5 is false, “return route”. If field 5 is true, “depart route”. * Field 7 is order in path. * Field 8 is not use. * Field 9 is name of station. * Field 10 is a latitude. * Field 11 is a longitude. * Field 12 is not use. * Field 13 is name of route. * Field 14 is id of station. | | | |

Table 27: <Staff> Configure parsing source.

###### <Scheduler> Parse bus timetable.

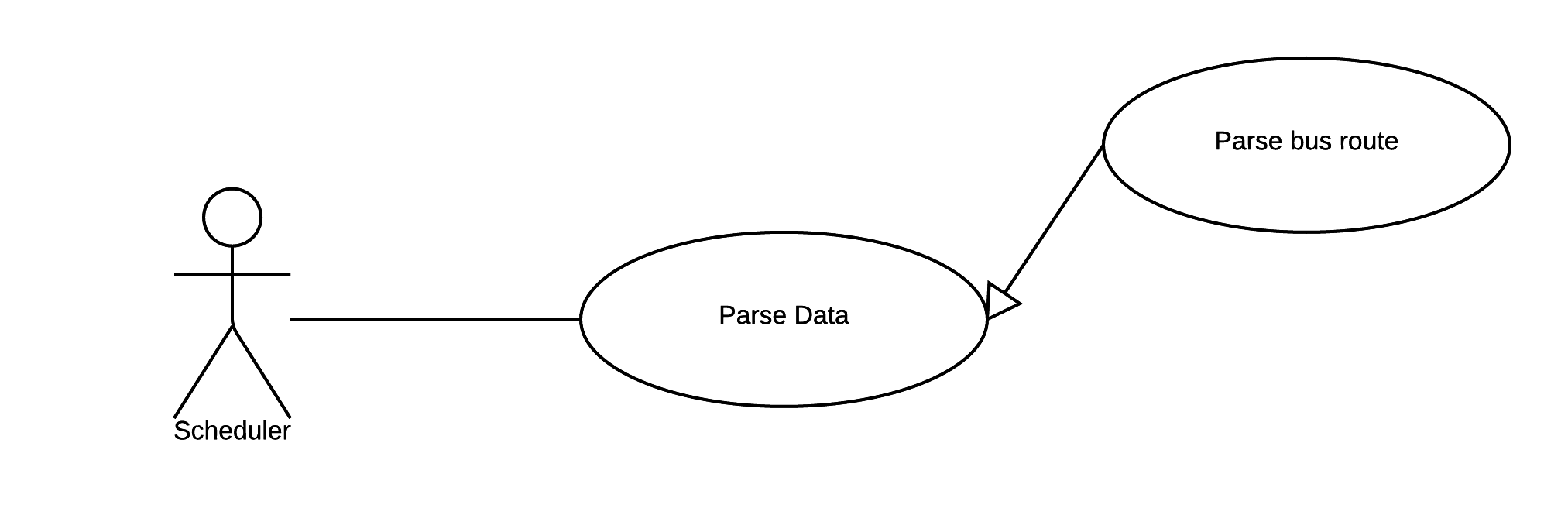


Figure 18: <Staff> Parse bus timetable

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – SC02 | | | |
| Use Case No. | SC02 | **Use Case Version** | 2.0 |
| Use Case Name | Parse bus timetable. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff**.**   Summary:   * This use case allows scheduler to parse bus timetable.   Goal:   * The system provides parse bus timetable for getting all bus timetable of each route from server to internal database for reuse later when third-party server is disconnected.   Triggers:   * Scheduler sends parsing source command.   Preconditions:   * Source has configured.   Post Conditions:   * Success**:** New data inserted to storage. * Fail: Nothing changed in storage.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | **1** | Scheduler sends parse bus time command. | * Scheduler has two ways for getting data: from local file or from third-party service. Base on configure setting (reference use case <Staff> configure parsing source), scheduler will choose method for getting data. * Fetch data based on Apache POI library. * Validate Data [Exception 1] * If data valid, scheduler will insert data into storage. * System show successful message to staff. * [Alternative 1] |   Alternatives Scenario:   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | System insert data to storage unsuccessfully. | System sends unsuccessful message to staff. |   Exceptions:  **[Exception 1]**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | **1** | File format is incorrect with default system file template. | * System terminates process parsing. * System show failed message to staff and suggests a template file. |   Relationships: **N/A.**  Business Rules:   * File for parsing must correct following template file:   C:\Users\datnt\Desktop\excelTemplate.JPG   * + File name must follow format: C:\Users\datnt\Desktop\fileName.JPG (9: route no, depart: route type)   + System currently just support excel file format.   + Successful message Content: “Parse bus timetable successful”. | | | |

Table 28: <Staff> Parse bus timetable.

#### Android Mobile Application

##### <Customer> Overview Use Case

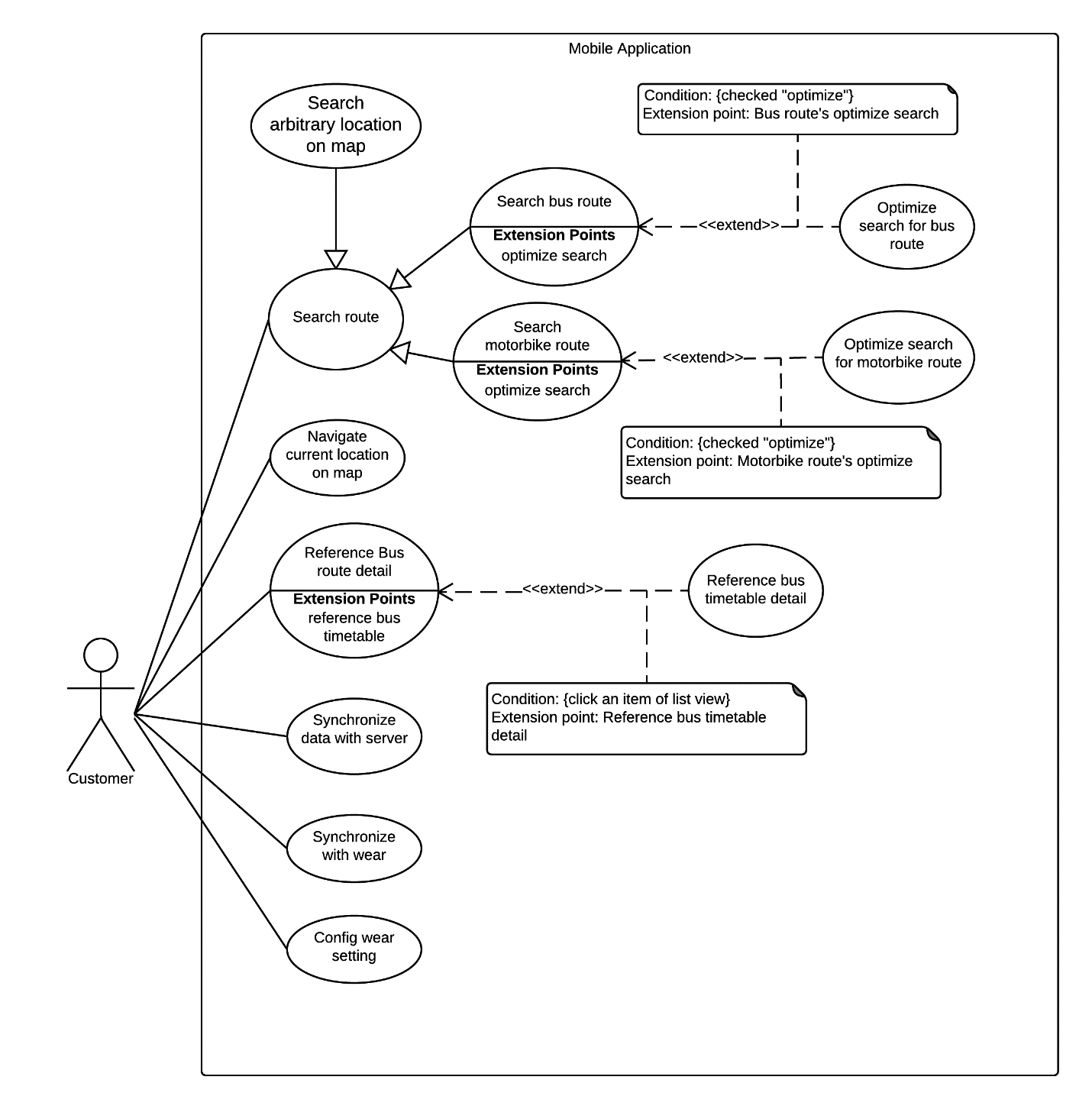


Figure 20: <Guest> Search Location

###### <Customer> Search Arbitrary location on map

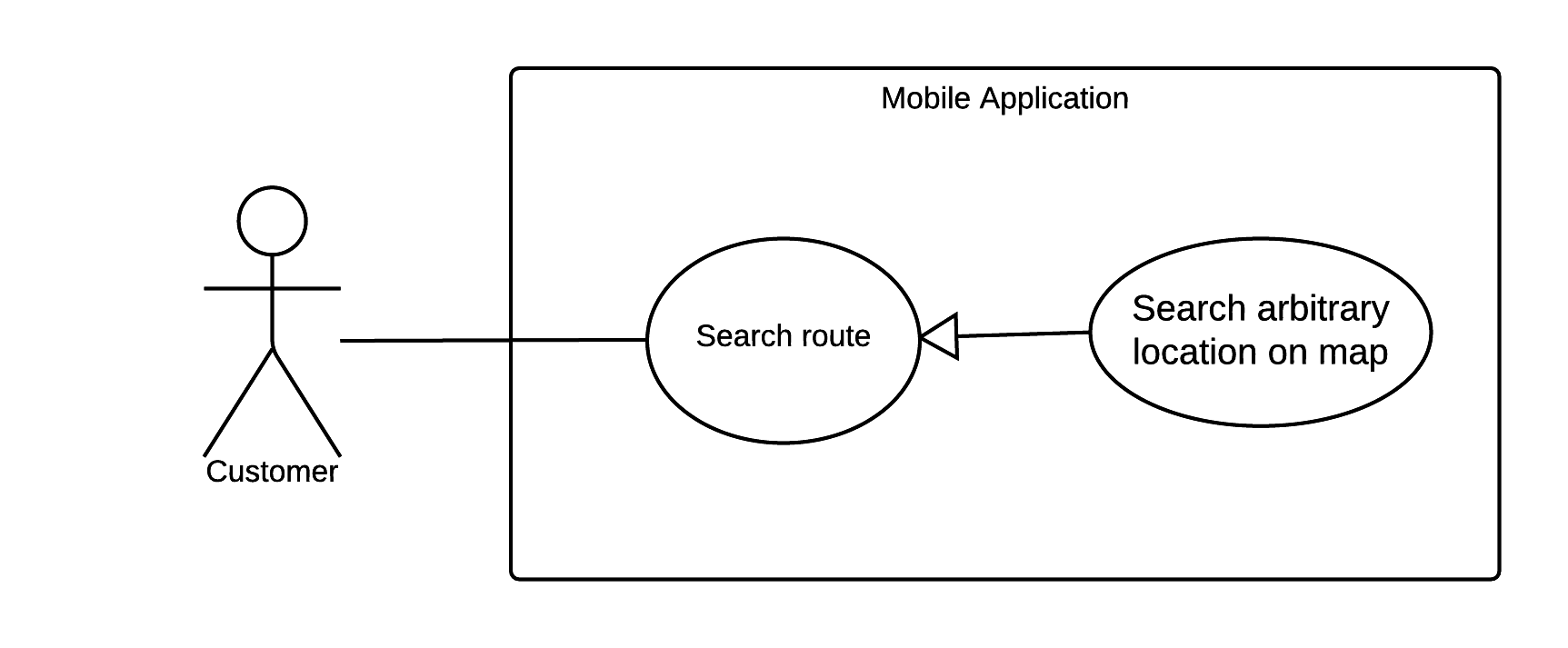


Figure 21: <Customer> Search Arbitrary Location on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB01 | | | |
| Use Case No. | MB01 | **Use Case Version** | 2.0 |
| Use Case Name | Search arbitrary location on map | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer search an arbitrary location on the map for the mobile application.   Goal:   * The system shows the location on the map. By using this function, customer will get more information before choose place for finding route.   Triggers:   * Customer sends the location command.   Preconditions:   * Mobile must be connected to the internet.   Post Conditions:   * Success: Show a location on map. * Fail: Show an error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The customer goes to search view. | System displays search view including:   * Location search: free text input. * Show list of search history. | | 2 | If customer inputs arbitrary location.  [Alternative 1] | System suggests location while typing. | | 3 | Customer sends search command search. | System processes and shows the location on the map.  [Alternative 2]  [Exception 1,2] |   Alternative Scenario:  [Alternative 1]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | If customer chooses a location from history. | System will show that location on free text input. |   [Alternative 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | If system cannot find a location on the map. | System will show a message notify to customer that location does not found on the map. |   Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t have the internet connection. | Show an error message that this device hasn’t connected to the internet yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Server responses error with some error code. | If error code is:   * UNKNOW\_ERROR: system shows customer message that request could not be processed due to a server error. The request may succeed if customer tries again. * OVER\_QUERY\_LIMIT: system shows customer message that server currently cannot support more queries. |   Relationships: N/A  Business Rules:   * System uses Nutiteq map to display the map. * System uses Google Map API to search a place. * System uses Google Place API for suggest place while typing on search field. * System uses Google Maps Geocoding API for converting latitude and longitude to normal string address. * Location will be stored in history database: * Searching place data. * Searching date data. * Detail information includes: * Street name. * Address. * History is limited in 20 nearest results. * Search history will have sorted by time from latest. * Currently system doesn’t solve OVER\_QUERY\_LIMIT error. This error means user has requested Google service more than 2500 times in that day. Customer should to wait next 24 hours for continue use this function. In the future, system will extend license to “Google Map API for Work” so our application can serve more than 100 000 requests per day. | | | |

Table 30: Use case MB01 - <Customer> Search Arbitrary Location on map

###### <Customer> Search Bus Route

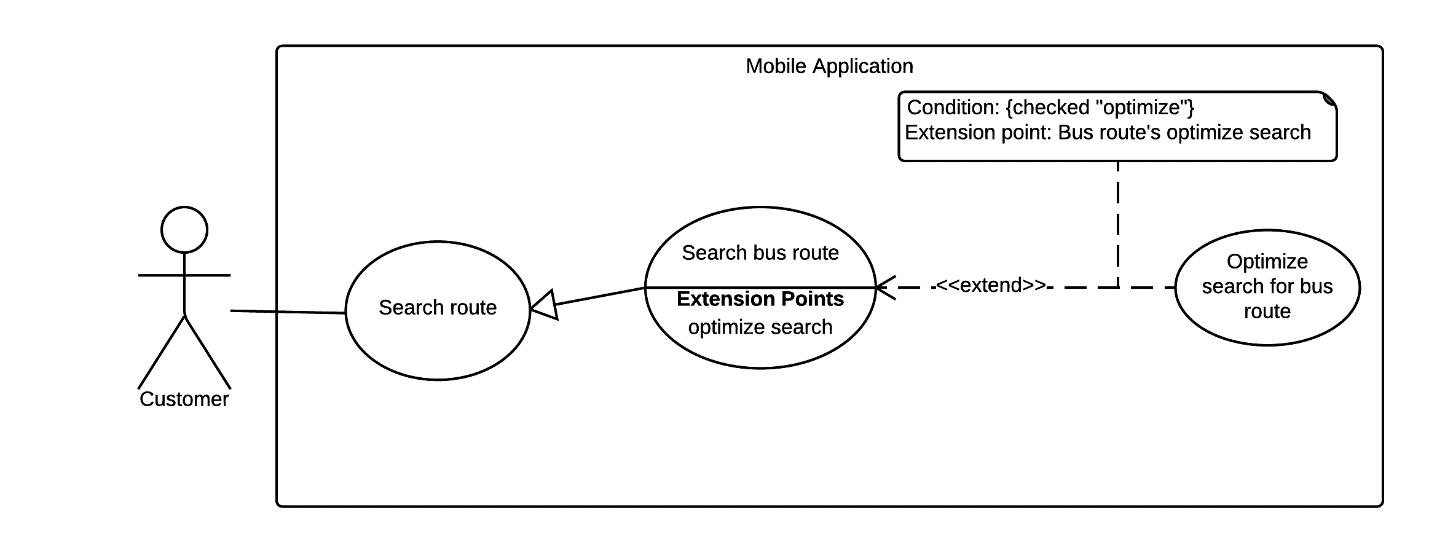


Figure 22: <Customer> Search Bus Route

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB02 | | | |
| Use Case No. | MB02 | **Use Case Version** | 2.0 |
| Use Case Name | Search Bus Route | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer search bus route through some locations (from two to four points) on the mobile application.   Goal:   * The system suggests some bus optimal paths to the customer. Customer can use this result to navigate a route when participating traffic.   Triggers:   * Customer sends a search bus route command.   Preconditions:   * Mobile must be connected to the internet.   Post Conditions:   * Success: The system draws route from the start location to the end location and show on the map. * Fail: Show an error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The customer goes to search route view. | Display search route view including:   * Start location: free input text. * End location: free input text. * Send a search command: button. * Send a set time command: button. * Send an optional command: button. * Send a voice command: button. | | 2 | When customer selects start location input text or end location input text. | Display new view for selecting place with following information:   * Location: free text input. * Location lists from history. * Send a voice command: button. | | 3 | If customer inputs start or end location.  [Alternative 1] | Input field shows location that customer has selected. | | 4 | The customer goes to choose the departure time view.  [Alternative 2] | System shows date time picker for customer selecting time. | | 5 | The customer can optionally choose departure time. | System update date on date time picker. | | 6 | The customer goes to choose the middle locations view. | Display new view with following information:   * First middle location: free input text. * Second middle location: free input text. * Optimal: checkbox * Optimal route: can be optional selected from these values: * “Đi bộ tối đa giữa hai trạm (mét)”. * “Số lần chuyển tuyến tối đa” | | 7 | The customer can input two middle locations.  [Alternative 1] | Input field shows location that customer has selected. | | 8 | Customer can choose the optimal path through multi points option. | Optimal path option dialog will show checked symbol. | | 9 | Customer can choose those option:   * “Đi bộ tối đa giữa hai trạm (mét)” * “Số lần chuyển tuyến tối đa” | System will show checkbox and free text field for the option user has selected. | | 10 | Customer can goes to search location with voice view. | Display new view with following information:   * Search voice command: button. | | 11 | When customer talk location in order to search. | Input field shows location that customer has talk. | | 12 | Customer sends search bus route command. | System shows a list with some optimal paths.  [Exception 1,2,3] | | 13 | Customer chooses one optimal path. | System draws the path on map and shows the detailed path. |   Alternative Scenario:  [Alternative 1]   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | If customer sends a location from history. | System will show the location that customer has selected on free input text. |   [Alternative 2]   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer can choose current system time. | System will update current system time on date picker. |   Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t connect the internet. | Show an error message that this device hasn’t connected to the internet yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | If the start location or the end location text field is empty. | System shows missing a field message to the customer. |   [Exception 3]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Customer choose optimal path between multi points option but only two locations has inputted. | System shows message that must has at least three locations for using this function. |   [Exception 4]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | System not found nearby station bus. | System shows message that “Không có trạm xe bus nào gần vị trí bạn tìm kiếm”. |   Relationships: Has an extension optimize search for bus route.  Business Rules:   * System uses Nutiteq map to display the map. * Location must be in HCM city. * The algorithm for finding bus route is called from SWR web service. * The optimal path for multi points option: Normally, first inputted point and last inputted point will be kept and system just rearrange all middle points for finding optimal path. By choosing this option, first inputted point will be start location, and all rest points will be rearranging for finding optimal path. So in this case, last inputted point doesn’t necessary a last location user will visited. * The system gives some options for finding optimal path: * Số lần chuyển tuyến tối đa: system will find fasted path that number of transfer doesn’t exceed certain number (input by user). System set default 2 transfers. * Khoảng cách đi bộ tối đa giữa hai trạm: Default is 300 m. So every walking between two stations, or from first location to first station doesn’t exceed this number. If “Khoảng cách đi bộ tối đa giữa hai trạm” of customer input not found route path but distance less than 1000 m found route, system will notify message ask customer want choose walking 1000m. If customer accept, system show route path else system show message “Không tìm thấy trạm nào gần”. | | | |

Table 31: Use case UC02 - <Customer> Search Bus Route

###### <Customer> Navigate Current Location On Map

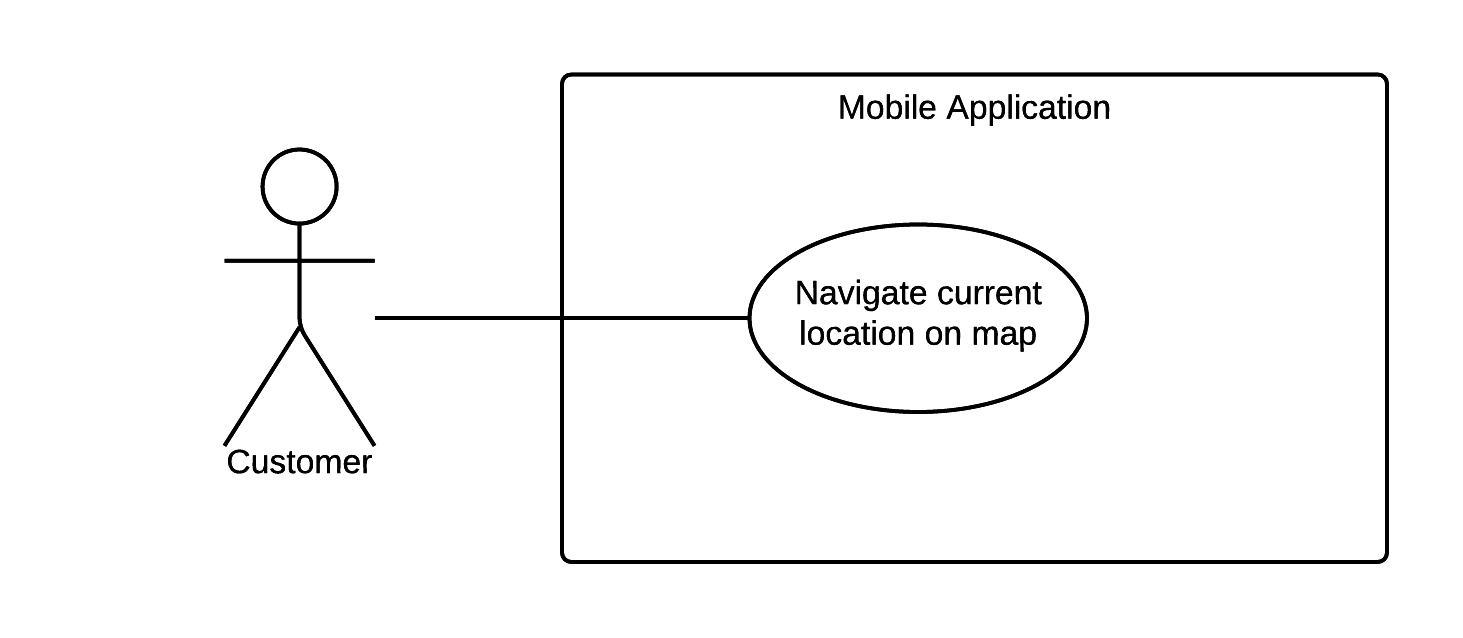


Figure 23: <Customer> Navigate Current Location On Map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB03 | | | |
| Use Case No. | MB03 | **Use Case Version** | 2.0 |
| Use Case Name | Navigate Current Location On Map | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer search current location on the map with GPS for mobile application.   Goal:   * The system shows the current location on the. This function will help customer know where they are on map, before choosing place for searching route.   Triggers:   * Customer sends current GPS command.   Preconditions:   * GPS must be enabled. * Mobile must be connected to the internet.   Post Conditions:   * Success: Show a location on map. * Fail: Show an error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to map view. | System will display Google map. | | 2 | Customer sends current location command. | The system displays current location on the.  [Exception 1,2] |   Alternative Scenario: N/A  Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t have the internet. | Show an error message that this device hasn’t connected to the internet yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | GPS doesn’t enable. | Show an error message that this device hasn’t enable GPS yet. |   Relationships: N/A  Business Rules:   * System uses Nutiteq map to display the map. * Using Google Maps Geocoding API for converting latitude and longitude to normal string address. | | | |

Table 32: Use case MB03 - <Customer> Navigate Current Location On Map

###### <Customer> Search Motorbike Route

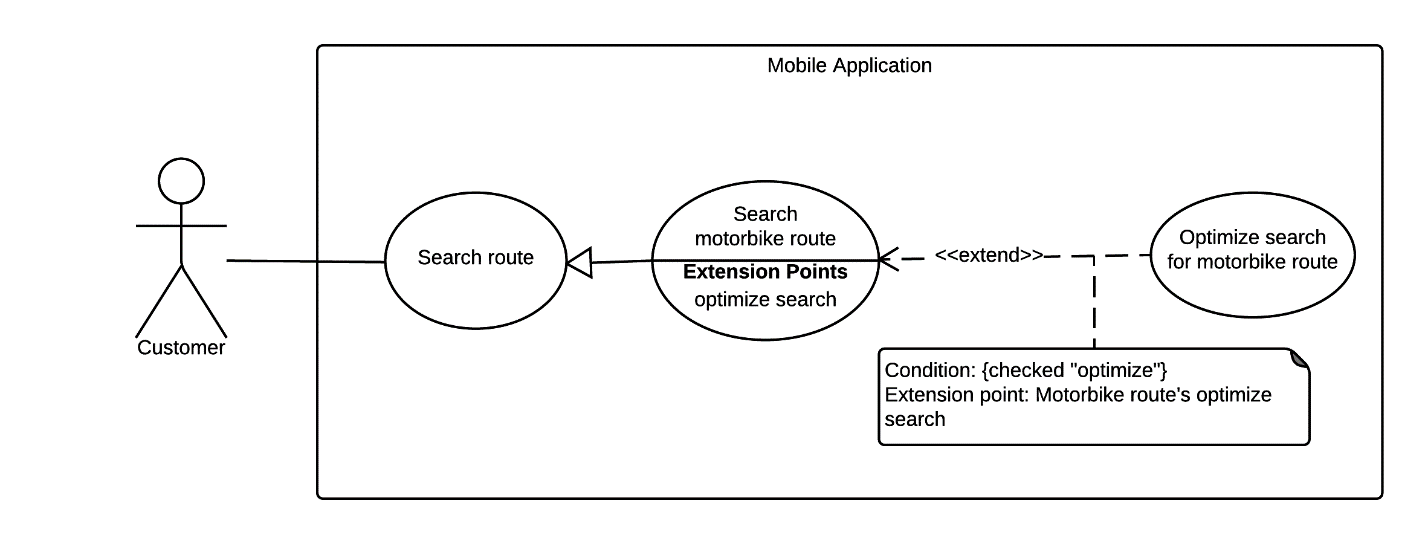


Figure 24: <Customer> Search Motorbike Route

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB04 | | | |
| Use Case No. | MB04 | **Use Case Version** | 2.0 |
| Use Case Name | Search Motorbike Route | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer search motorbike route thought some locations (from two to four) on the mobile application.   Goal:   * The system suggests some motorbike optimal paths to the customer. Customer can use this result to navigate a route when participating traffic.   Triggers:   * Customer sends search motorbike route command.   Preconditions:   * Mobile must be connected to the internet.   Post Conditions:   * Success: System draws route from the start location to the end location and shows on the map. * Fail: Show an error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The customer goes to search route view. | Display search route view including:   * Start location: free input text. * End location: free input text. * Send a search command: button * Send a set time command: button. * Send an optional command: button. * Send a voice command: button. | | 2 | When customer select start location input text or end location input text. | Display new view with following information:   * Location: free text input. * Location from history. | | 3 | If customer inputs start and end location.  [Alternative 1] | Input field shows location that customer has selected. | | 4 | The customer goes to choose the departure time view. | System shows date time picker for customer selecting time. | | 5 | The customer can optionally choose departure time.  [Alternative 3] | System update date on date time picker. | | 6 | The customer goes to choose the middle locations view. | Display new view with following information:   * First middle location: free input text. * Second middle location: free input text. * Optimal: checkbox * Optimal route: can be optional selected from these values: * “Đi bộ tối đa giữa hai trạm (mét)” (disable). * “Số lần chuyển tuyến tối đa” (disable). | | 7 | The customer can input two middle locations.  [Alternative 1] | Input field shows location that customer has selected. | | 8 | Customer can choose the optimal path through multi points option. | Optimal path option dialog will show checked symbol. | | 9 | Customer can goes to search location with voice view. | Display new view with following information:  Search voice command: button. | | 10 | When customer talk location in order to search. | Input field shows location that customer has talk. | | 11 | Customer sends a command search. | System shows a list with some optimal paths.  [Exception 1, 2, 3] | | 12 | Customer chooses one path command. | System draws the path on map and shows the detailed path. |   Alternative Scenario:  [Alternative 1]   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | If customer sends the location from history. | System processes and input field shows the location that customer has selected. |   [Alternative 2]   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Customer can choose current system time. | System will update current system time on date picker. |   Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t connect the internet. | Show an error message that this device hasn’t connected to the internet yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 2 | If the start location or the end location text field not input. | System shows missing field message to customer. |   [Exception 3]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Customer choose optimal path between multi points option but only two locations has inputted. | System shows message that must has at least three locations for using this function. |   Relationships:   * Has an extension optimize search for motorbike route.   Business Rules:   * System uses Nutiteq map to display the map. * Using Google map API for searching route. * Location must be in HCM city. * The optimal path for multi points option: Normally, first inputted point and last inputted point will be kept and system just rearrange between points for finding optimal path. By choosing this option, first inputted point will be start location, and all rest points will be rearranging for finding optimal path. So in this case, last inputted point doesn’t necessary a last location user will visited. | | | |

Table 33: Use case MB04 - <Customer> Search Motorbike Route

###### <Customer> Setting

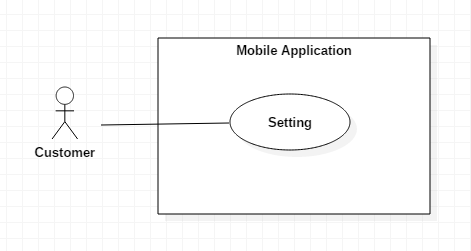


Figure 25: <Customer> Setting

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB05 | | | |
| Use Case No. | MB05 | **Use Case Version** | 2.0 |
| Use Case Name | Setting | | |
| Author | NgoanTT | | |
| Date | 21/11/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows the customer change setting from mobile application.   Goal:   * Customer can system configuration such as change port and IP address in order to bus can access to server, custom speed simulate, change distance to notify, download map offline and choose option display result.   Triggers:   * A customer sends change setting command.   Preconditions:   * Mobile must be connected with internet.   Post Conditions:   * Success: Change setting successful. * Fail: Show an error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to change setting view. | Display change setting view including:   * Bus server IP: free input text. * Bus server port: free input text. * Simulate speed: free input text. * Notify nearby bus station: free text input. * Notify nearby motorbike step: free text input. * Priority sort for bus: spinner. * Priority sort for motorbike: spinner. * Download map offline: checked. * Send a change setting command: button * Send a cancel command: button. | | 1 | A customer sends change setting command. | System processes.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t connect the internet. | Show an error message that this device hasn’t connected to the internet yet. |   Relationships: N/A  Business Rules: N/A. | | | |

Table 349: Use case MB10 - <Customer> Setting

###### <Customer> Download Audio File

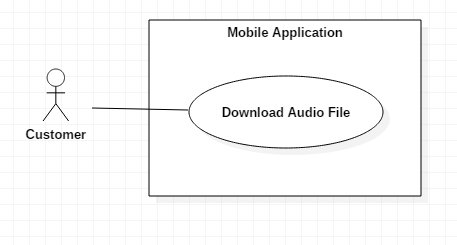


Figure 26: <Customer> Download audio file

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB06 | | | |
| Use Case No. | MB06 | **Use Case Version** | 2.0 |
| Use Case Name | Download Audio File | | |
| Author | NgoanTT | | |
| Date | 21/11/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows the customer download audio file for converting text to speech from FPT service for mobile application.   Goal:   * Customer can download audio file from FPT service in order to notify to customer when custom move on the path. File downloads saved in cache.   Triggers:   * Customer send download audio command.   Preconditions:   * Mobile must be connected with internet. * Customer has to search route and view map.   Post Conditions:   * Success: Notify message to customer. * Fail: Show an error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | A customer sends download audio file command. | System processes and downloads data successfully.  [Exception 1,2] |   Alternative Scenario: N/A  Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Mobile doesn’t connect the internet. | Show an error message that this device hasn’t connected to the internet yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Download audio file unsuccessful. | Show an error message that download fail. |   Relationships: N/A  Business Rules:   * System use FPT Service for Text-To-Speech API. This address is: http://118.69.135.22 * File download saved in cache and cache is set 100mb, if cache is full, it autoes delete information less use least. | | | |

###### <Customer> Voice search input

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB07 | | | |
| Use Case No. | MB07 | **Use Case Version** | 2.0 |
| Use Case Name | Voice search input | | |
| Author | NamNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customers to search location with their voice.   Goal:   * The application displays a list of locations base on user’s speech.   Triggers:   * Customer searches by voice.   Preconditions:   * Mobile must be connected to internet.   Post Conditions:   * Success: Application displays locations that user has said. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | Customer goes to voice search view. | Application show voice button. | | 2 | Customer selects voice button and says follow syntax. | System will use Google Speech To Text API for speech recognition.  After recognize, system will analysis text base on grammar to get location text.  System will send this location string to Google API for getting detail location (including latitude and longitude).  System displays that location for user to know.  Based on syntax, application detects his/her voice to keyword.  With that keyword, application will search location and save it, which closer with keyword. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * The grammar is: * Departure location must be said: “Tôi muốn đi từ” or “Từ ...” * First middle location must be said: “Điểm trung gian thứ nhất là ...” * Second middle location must be said: “Điểm trung gian thứ hai là ...” * Arrival location must be said: “Tôi muốn đi đến …” or “Đến …” * When user inputs new location of same field, old location will be overwrite. | | | |

Table 35: <Customer> Voice search input

###### <Customer> Autocomplete with keyword

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB08 | | | |
| Use Case No. | MB08 | **Use Case Version** | 2.0 |
| Use Case Name | Autocomplete with keyword | | |
| Author | NamNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer to search location with key word.   Goal:   * The application displays a list of locations have name closer with keyword.   Triggers:   * Customer input key word into textbox.   Preconditions:   * Mobile must be connected to internet.   Post Conditions:   * Success: Application displays a list of locations. * Fail: Show an error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | Customer goes to autocomplete view. | Application show autocomplete textbox and locations’ history, which customer has searched. | | 2 | Customer input keyword | Application show list of locations, which have name closer with keyword.  [Exception 1]  [Alternative 1] | | 3 | Customer choose location on list. And click back button. | Application will save that location to prepare for search route. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No. | Cause | Application Response | | 1 | If no location was founded. | Application returns empty list. |   Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | Application Response | | 1 | If mobile is not connected to internet. | Application will show message that mobile must be connect to internet. |   Relationships:  Business Rules:   * Location are loaded using Google AutoComplete API. * When system sends text to Google service for getting location, application will stop getting data from user input for saving queries. After getting data, system will again new user input to server. | | | |

Table 36: <Customer> Autocomplete with keyword

###### <Customer> Notify on right route when using motorbike

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB09 | | | |
| Use Case No. | MB09 | **Use Case Version** | 2.0 |
| Use Case Name | Notify on right route when customer using motorbike | | |
| Author | NamNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows application notifies to customers when they move on right route when using motorbike.   Goal:   * The application show notification to customers. And they will know what should do next.   Triggers:   * Customers moves on right route.   Preconditions:   * Customer moves on right route, which they have chosen and meet events on that route. * Tracking is enable.   Post Conditions:   * Success: Application notifies on mobile and wear. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | Customers move on right route and meet events on it | Application will detect their location and show message that customers should to do next.  Mobile will send message to wear, and it will show notification. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Customers will be notified by message on mobile, by sound and by message on wear. * Message will depend on what event they meet. These events include: * User is near a bus station that should to leave * User is near a motorbike turn that should to turn. * User is near the place (in both bus and motorbike) that they want to come. | | | |

###### <Customer> Notify when user is on wrong route when using motorbike

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB | | | |
| Use Case No. | MB10 | **Use Case Version** | 2.0 |
| Use Case Name | Notify when user is on wrong route when using motorbike | | |
| Author | NamNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows application notifies to users when they move on wrong route when using motorbike.   Goal:   * The application shows notification to users when they go to wrong way so they know can view again route before continue to participating traffic.   Triggers:   * Customers moves on wrong route.   Preconditions:   * Customers don’t move on right route, which they have chosen. * Tracking is enable.   Post Conditions:   * Success: Application notifies on mobile and wear. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | User doesn’t move on right route. | Application will detect their location and show message that customers are going to wrong way.  Mobile will send message to wear, and it will show notification. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * User will be notified by message on mobile, by sound and by message on wear. * User can change in setting, how far from true route so application will understand as wrong way and notify to user. | | | |

Table 37: Notify when user is on wrong route

###### <Customer> Notify when user is from wrong way near again searched route when using motorbike

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB | | | |
| Use Case No. | MB | **Use Case Version** | 2.0 |
| Use Case Name | Notify when user is from wrong way near again searched route when using motorbike | | |
| Author | NamNT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows application notifies to users when they from wrong way and go again near to original searched route when using motorbike.   Goal:   * The application shows notification to users when they go again searched route, so they can know direction of searched route and understand what should to do next.   Triggers:   * Customers moves on wrong route and come near again searched route.   Preconditions:   * Customers don’t move on right route, which they have chosen and come again searched route. * Tracking is enable.   Post Conditions:   * Success: Application notifies on mobile and wear. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | User is on wrong way and come near again searched route. | Application will detect their location and show message about direction for coming to true route.  Mobile will send message to wear, and it will show notification. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * User will be notified by message on mobile, by sound and by message on wear. * Recommend information includes: * Direction (left, right) for coming to true route. * Nearest turn that user should to come. * If user has network, system will: * Search again new route from current place to all of locations customer haven’t came yet. * This new route using Google Place API service. | | | |

Table 38: Notify when user is from wrong way near again searched route

#### Android Wear Application

##### <Customer> Overview Use Case

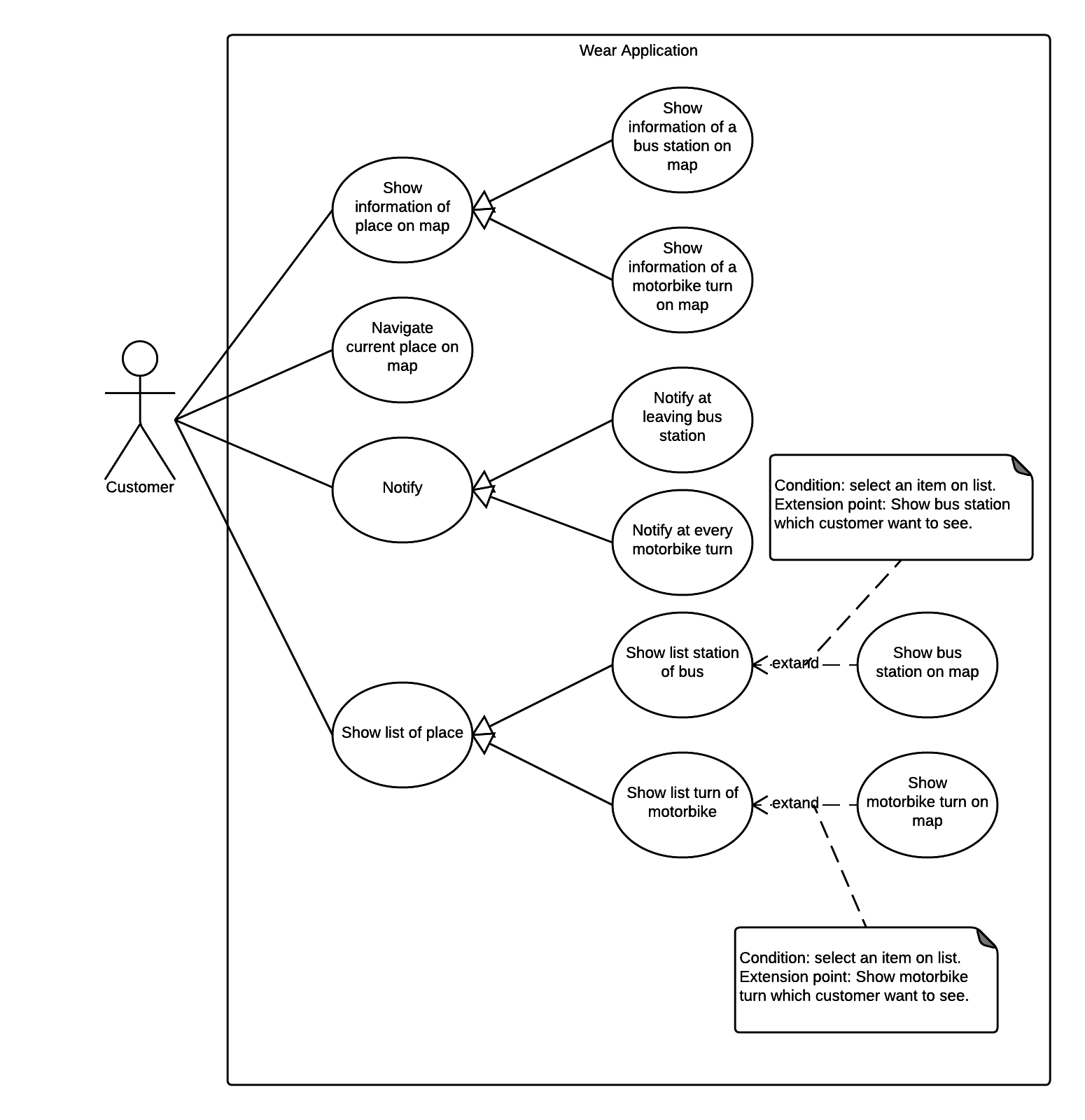


Figure 27: <Customer> Overview use case

###### <Customer> Show detail route on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR01 | | | |
| Use Case No. | WR01 | **Use Case Version** | 1.0 |
| Use Case Name | Show detail route on map. | | |
| Author | NamNT | | |
| Date | 27/11/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer see motor route or bus route on map, which customer has chosen.   Goal:   * Show detail of route on map which customer has chosen. Using this function, customer can locate which turn on map for easier navigate where they are on street.   Triggers:   * Customer sends show detail route command.   Preconditions:   * Wear is paired with mobile. * Mobile has sent data to wear. * Android and wear are in connecting range. * If wear device doesn’t have internet, mobile must be connected to internet.   Post Conditions:   * Success: Map moves to new place and shows information of route. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | Choose route on mobile | Wear turn on map, navigate to location of route and show detail of it. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Application uses Google Map for rendering route. * If android wear has own GPS, application will use android wear’s GPS. Otherwise, android wear application will use GPS from mobile device * Detail information includes: * Bus route or motorbike route. * Important location including: * Departure place and arrival place. * Middle bus stations that user should to leave. * Motorbike turn on searched route. | | | |

Table 39: Use case UR01 - <Customer> Show information of a bus station on map

###### <Customer> Notify when wear is received message from mobile

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR05 | | | |
| Use Case No. | WR02 | **Use Case Version** | 1.0 |
| Use Case Name | Notify when wear is received message from mobile | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows wear notifies to customer when has message from mobile.   Goal:   * Wear will show notification of mobile so customer can see easily on wear.   Triggers:   * Mobile send message to wear.   Preconditions:   * Wear is paired with mobile. * Android and wear are in connecting range.   Post Conditions:   * Success: Wear shows notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | Application Response | | 1 | Mobile send data to wear. | Wear will receive notification and display notification to user.. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Mobile sends data to wear when: * User is near a bus station that should to leave * User is near a motorbike turn that should to turn. * User goes to wrong route and need to notify * User comes again to true route and system will display a notify for user knows which should to do next. * A notification on wear often has two screen: * First screen: summarize information such as image of direction, short description of which action should to do next * Second screen: more detail information about next place. | | | |

Table 40: <Customer> Notify when wear is received message from mobile

## Software System Attribute

### Usability

#### Graphic User Interface

For mobile application and wear application, all the texts, labels and alerts will be written in Vietnamese.

For web application, all the texts, labels and alerts will be written in English.

#### Usability

* The system usability is easy to use that will need less than 3 days of training for company staffs to use the system management.
* Customers can use all mobile application’s functions by reading help manual inside mobile application.
* Bus drivers need less than 1 hours of training to use bus driver’s mobile application.

#### Installation

* User can follow installation and manual guide for installation. If there are any problems, user cans contacts developer for help.

### Reliability

* Scheduler task runs at OAM everyday with 100% execution rate.

### Availability

* N/A

### Security

* All data are validated before saving to database.
* Staff password must be encrypted in database.
* All data from background handler or bus driver’s uploaded data must be approved before saving to database.
* All privacy information such as search history is only stored at local database.

### Maintainability

* The system is separated into modules.

### Portability

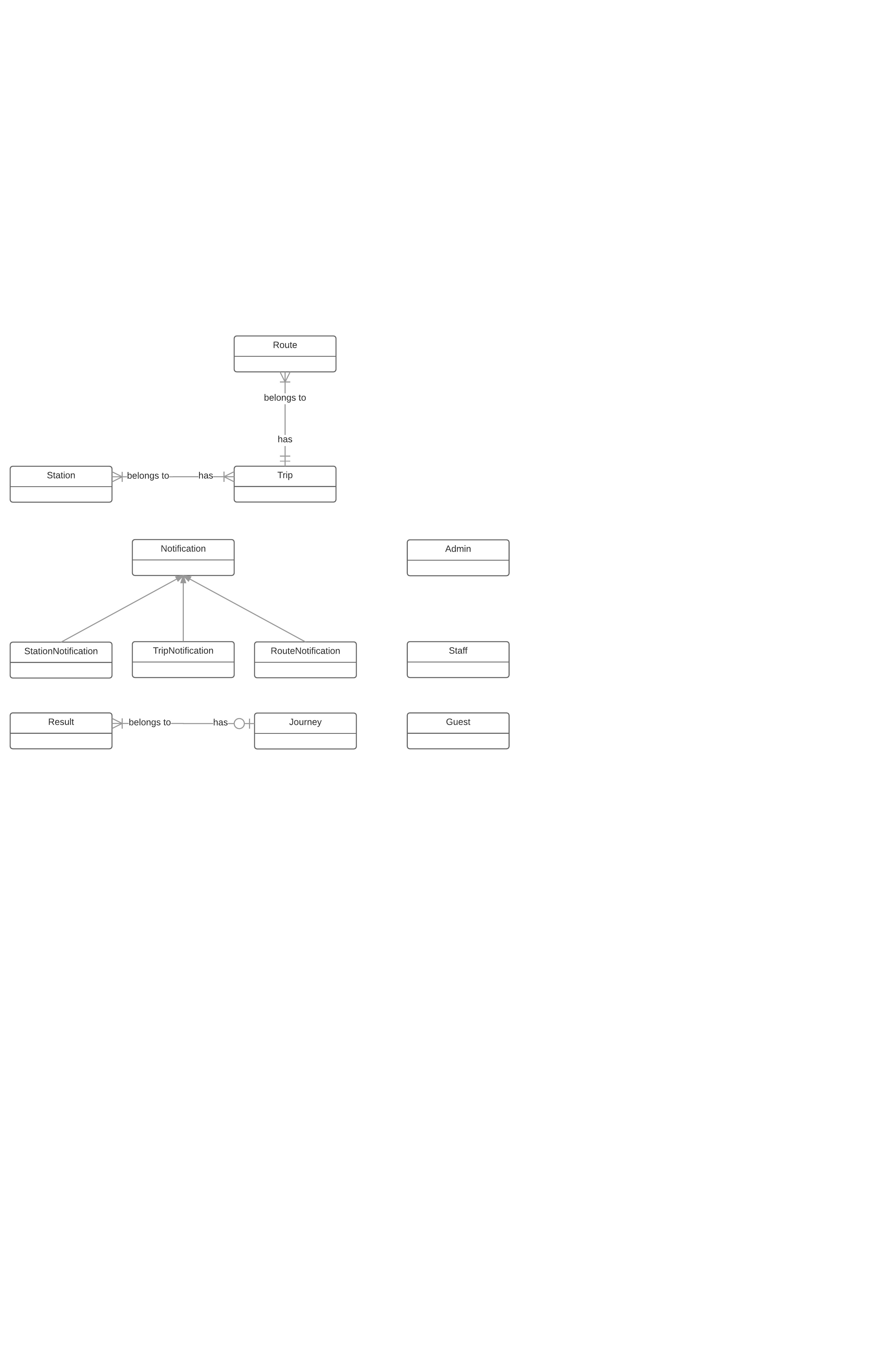
* Staff can use application on every OS supported web browser.
* Customer can use mobile application on every Android smartphone that have version greater than 4.3 and wear that have version from API 20.

### Performance

* Requests from mobile application to server for finding bus route are responded in less than 15 seconds at network connection 8 Mbps.
* Algorithm for finding bus route must run less than 10 seconds for Ho Chi Minh bus system.
* Mobile application synchronizes data with server in less than 1 minute at network connection 8 Mbps.

## Conceptual Diagram

**Web conceptual diagram:**

****

**Data Dictionary**

|  |  |
| --- | --- |
| Entity Data dictionary: describe all content of all entities | |
| **Entity Name** | **Description** |
| Admin | Abstract entity describes an admin in system. |
| Staff | Abstract entity describes a staff in system. |
| Guess | Abstract entity describes a guess in system. |
| Notification | Abstract entity describes a notification in system. |
| StationNotification | Notification for bus route. |
| TripNotification | Notification for bus time. |
| RouteNotification | Notification for route. |
| Station | Abstract entity represents distinct location in the network where one can board or get off a vehicle (bus, train) |
| Trip | Abstract entity represents a journey of one route with start time from first station and end time of last station. |
| Route | Abstract entity represents a sequence of stations a specific vehicle (train, bus, subway …) |
| Result | Abstract entity describes a bus result of two points. |
| Journey | Abstract entity describes a bus result of three or four points. |

*Table 48: Conceptual Diagram Data Dictionary*

**4.2. Mobile conceptual diagram:**

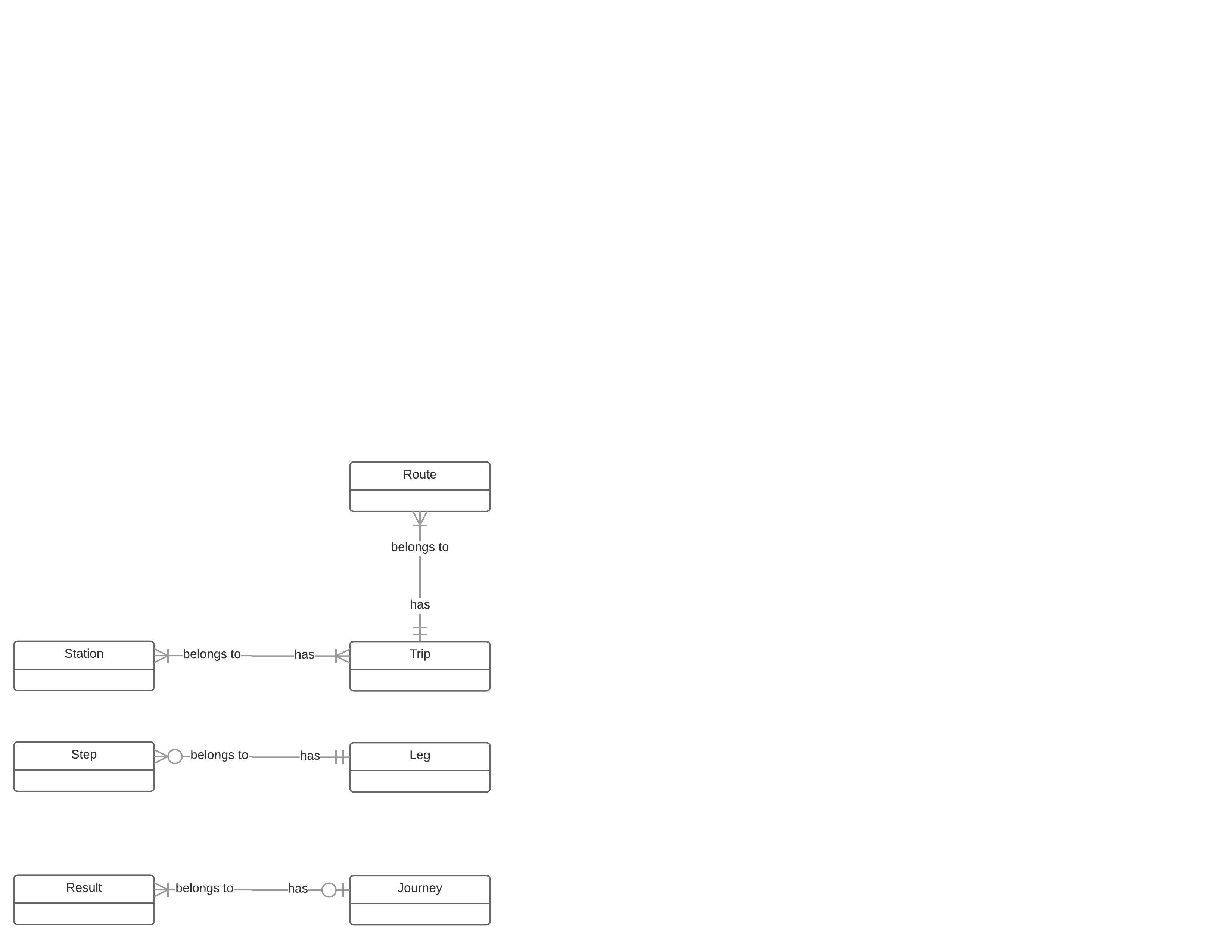
****

Figure 41: Conceptual diagram

**Data Dictionary**

|  |  |
| --- | --- |
| Entity Data dictionary: describe all content of all entities | |
| **Entity Name** | **Description** |
| Station | Abstract entity represents distinct location in the network where one can board or get off a vehicle (bus, train) |
| Trip | Abstract entity represents a journey of one route with start time from first station and end time of last station. |
| Route | Abstract entity represents a sequence of stations a specific vehicle (train, bus, subway …) |
| Result | Abstract entity describes a bus result of two points. |
| Journey | Abstract entity describes a bus result of three or four points. |
| Step | Abstract entity describes a detail instruction for motorbike route. |
| Leg | Abstract entity describes detail information for going from one place to another. |

*Table 49: Conceptual Diagram Data Dictionary*

# **Software Design Description**

## Design Overview

* This document describes the technical and user interface design of SWR system. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe a fully attributed Entity Relationship Diagram.

## System Architecture Design

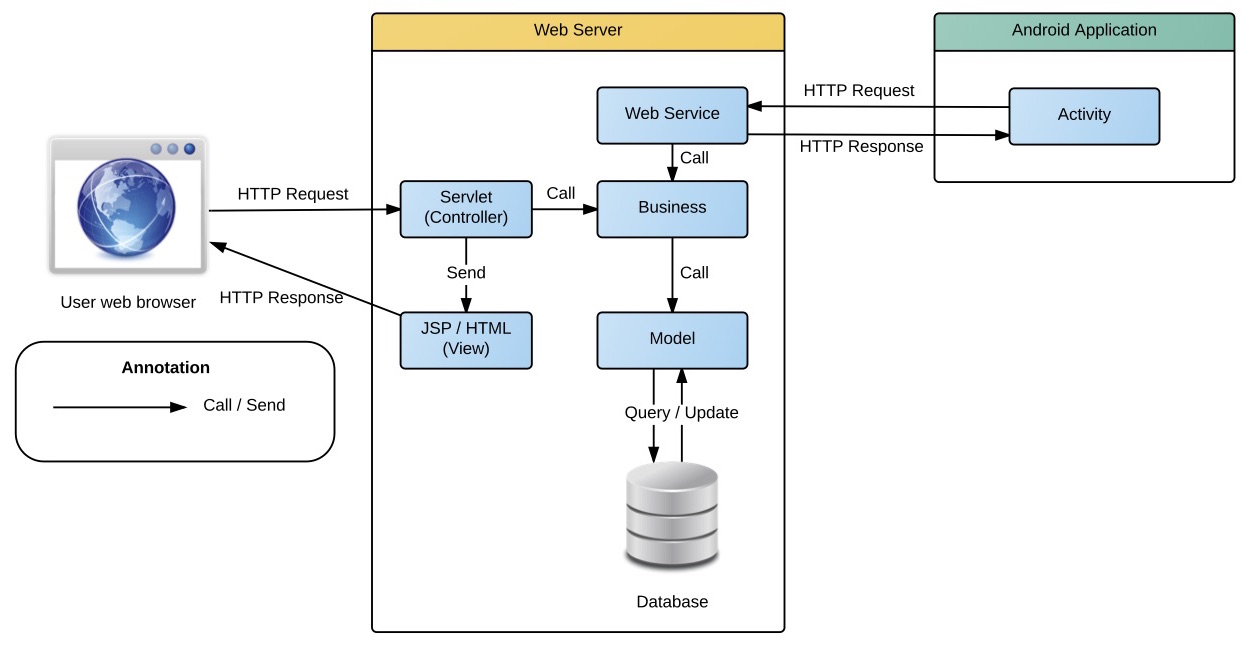


Figure 1 System architecture design

This diagram is referenced and modified from an original concept from: Chapter 6 Architecture Design, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

### Web Application architecture description

In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of following advantages:

* Web app contains a Web service (public API for mobile app), with MVC architecture, we can separate business code with Controller and View, so we can use the business code in web service without repeat the code.
* In scope of 4-member team, MVC architecture make it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

* **Servlet (Controller)** is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business’s method then selects view to return to user.
* **JSP/HTML (View)** is the parts of the application that handles the display of the data. The selection of View is under control of Controller.
* **Business** is the parts of the application that do business processing to solve domain problems.
* **Model** is the parts of the application that acts like a data transfer object between the system and database.
* **Web Service** is the parts of the application that acts like event handler for web and mobile communication via REST method.

### Mobile Application architecture description

The application is developed as an Android native application. In general, the application architecture conforms to Android architecture.



**Reference:** [Android Developer Guide - Application Fundamentals](http://developer.android.com/guide/components/fundamentals.html)

http://developer.android.com/guide/components/fundamentals.html

This project follows Android application architecture with following component:

* **Activity** is the basic core of an android application that handles user input,create thread to run asynchronous tasks, send request and receive data from server via web services ...

## Component Diagram

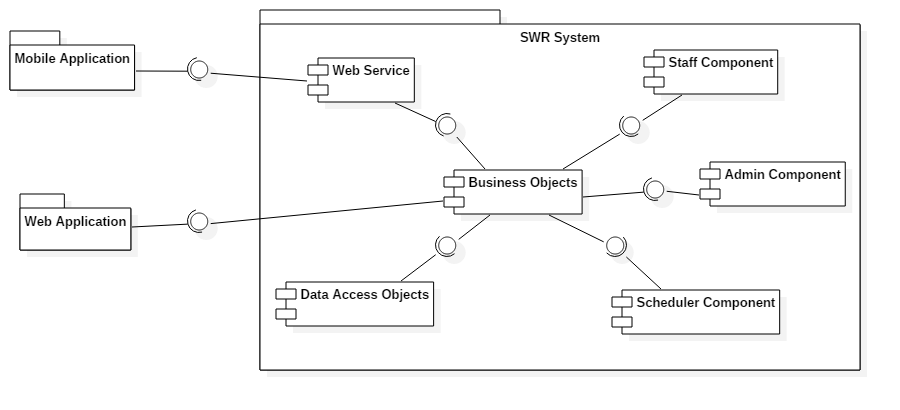


Figure 2 Component Diagram

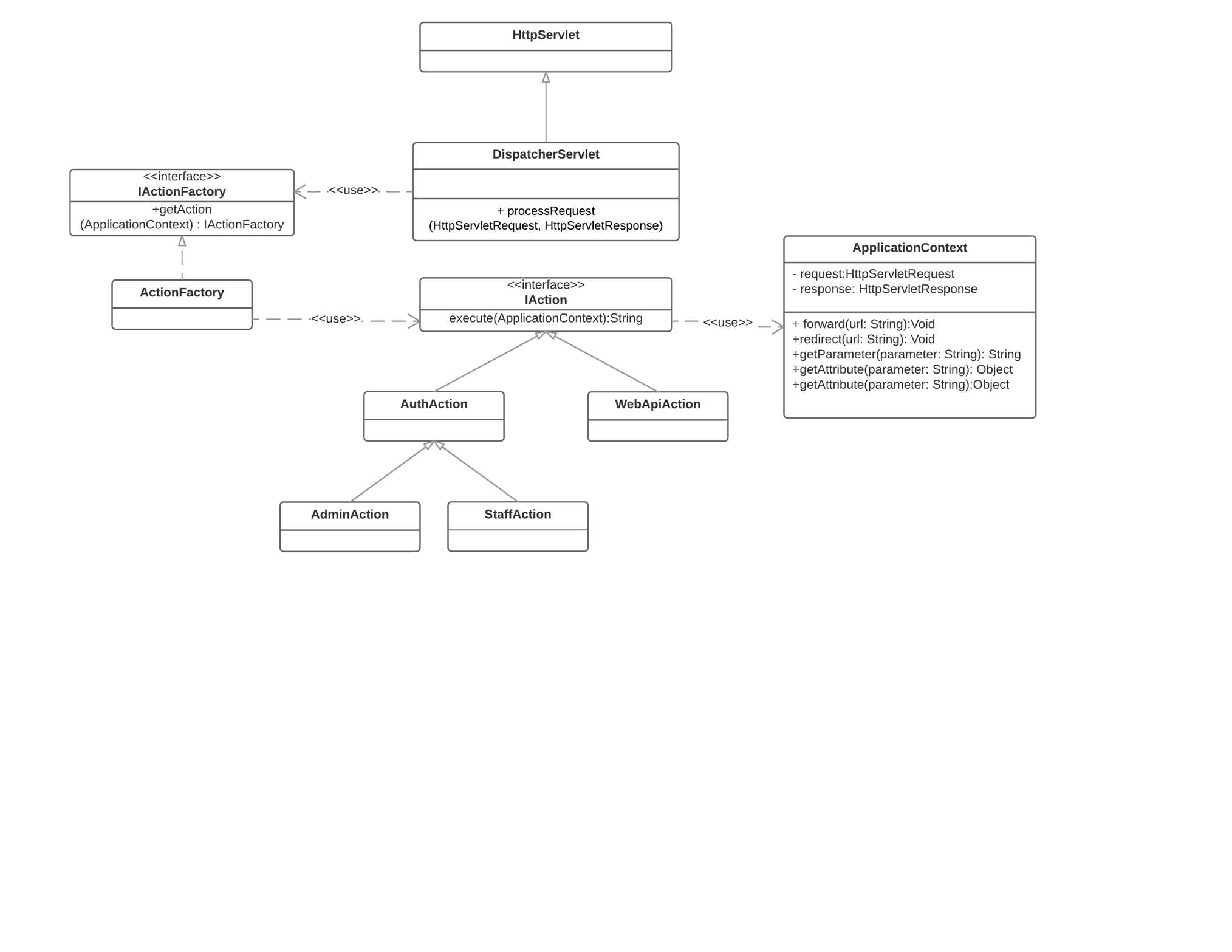
|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| Web Application | Web application package: View, Controller |
| Mobile Application | Mobile application package |
| Web Service | operations o handle interaction between the system and database. Provide API for mobile applications to interact with the system. |
| Staff Component | Component to handle staff activities in the system |
| Admin Component | Component to handle admin activities in the system |
| Scheduler Component | Component to handle scheduler in the system |
| Business Objects | Common objects to handle domain business operations for each components |
| Data Access Objects | Component to handle interaction between the system and database |

Table 1 Component Dictionary

### Detailed Descrip Class Diagram

Because web and mobile application need different entities, so conceptual diagram on web will be different to conceptual diagram on mobile application. We decide to design two separated conceptual diagrams.

**4.1.1. Web System class diagram:**



**4.1.1.1. HttpServlet:**

HttpServlet class is a system class of JavaEE framework. HttpServlet is an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.

**4.1.1.2. DispatcherServlet:**

DispatcherServlet is an application class, inherits from HttpServlet and get all requests from client. Base on action string from client, DispatcherServlet will call ActionFactory for getting suitable action for processing.

**4.1.1.3. ApplicationContext**

**4.1.1.4. IActionFactory:**

**4.1.1.5. ActionFactory**

**4.1.1.6. IAction**

**4.1.1.7. AuthAction**

**4.1.1.8. WebApiAction**

**4.1.1.9. AdminAction**

**4.1.1.10. StaffAction**

**4.1.2. Web mobile class diagram:**

**4.1.2. Web class diagram:**

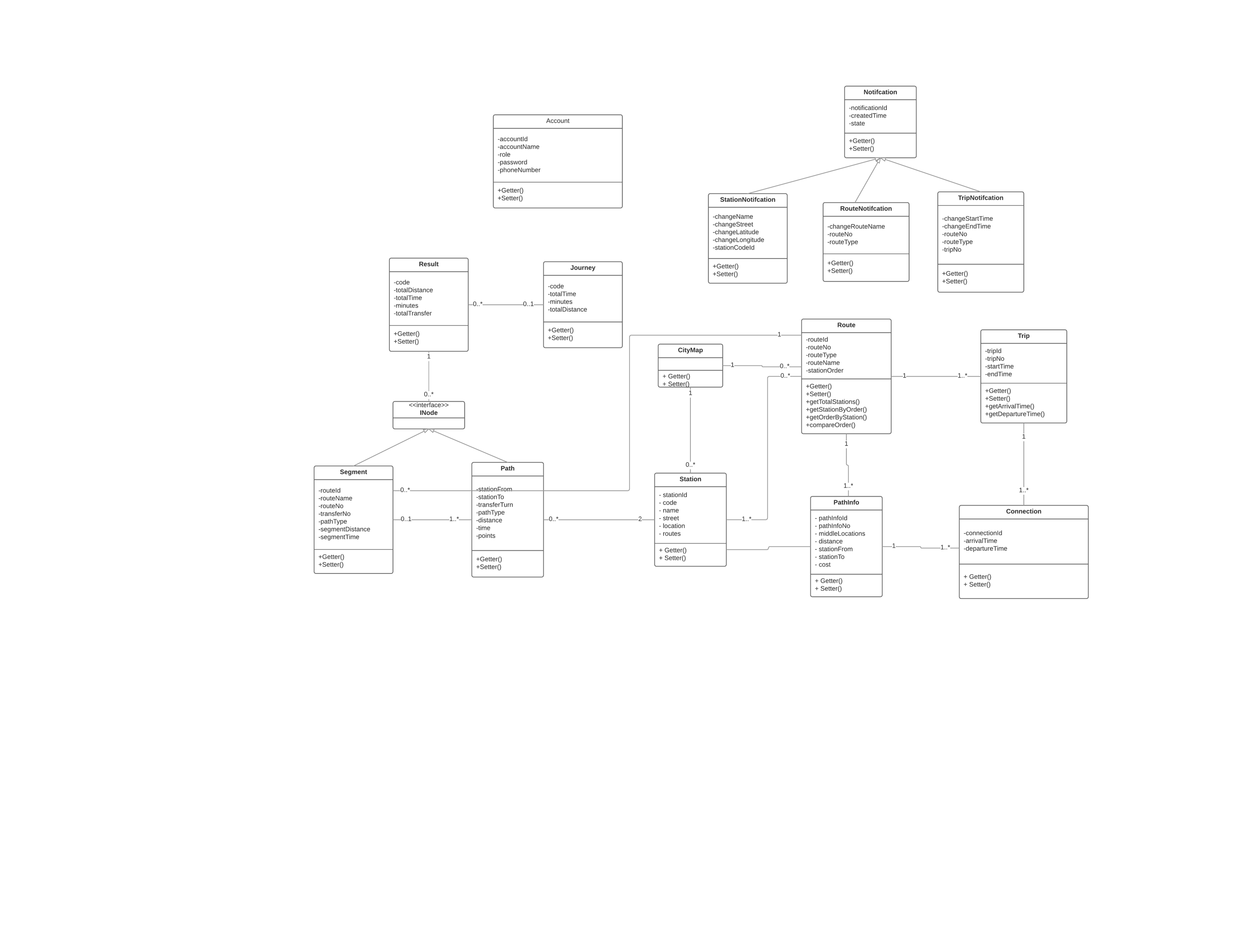


Table 2 Class dictionary

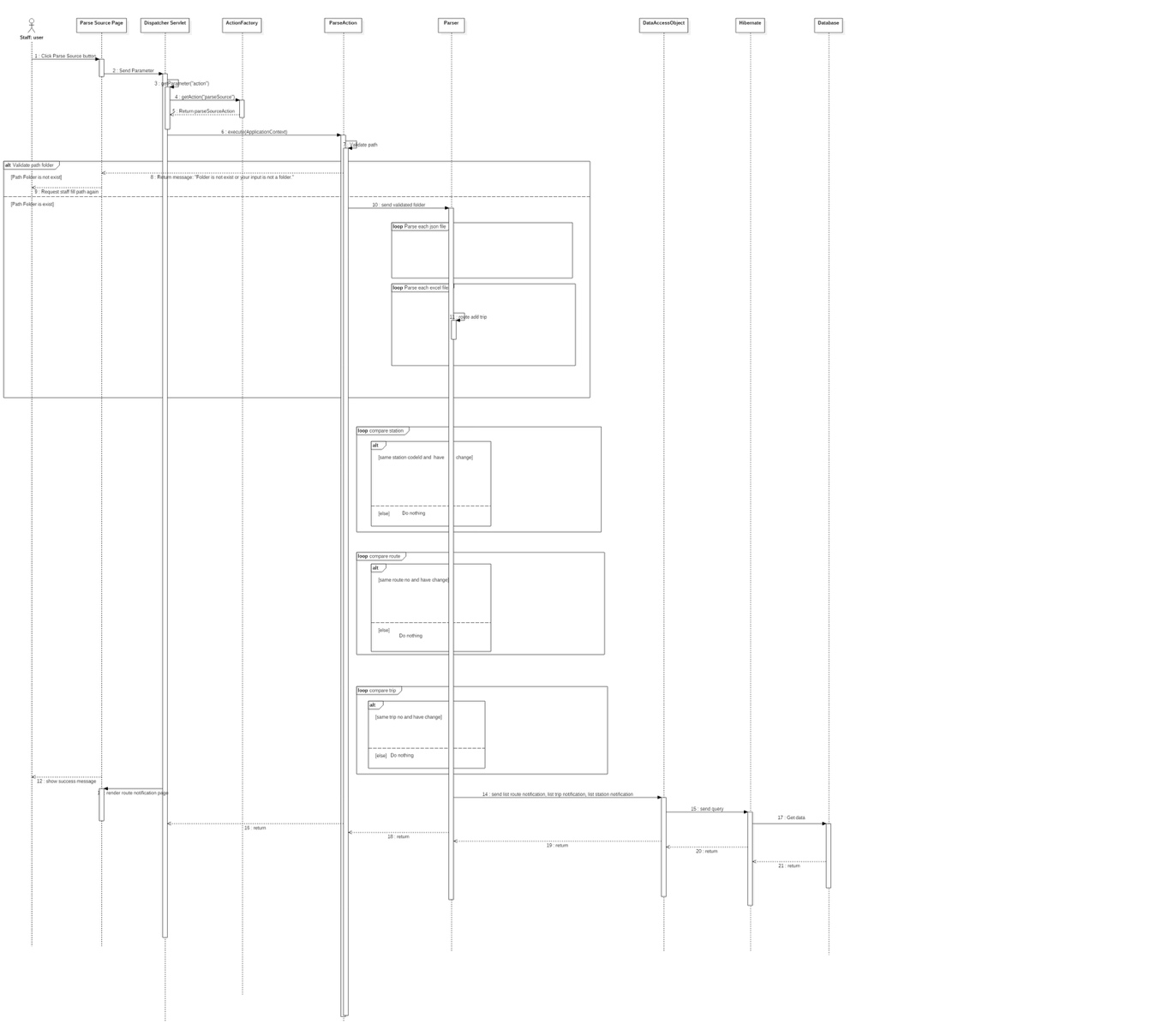
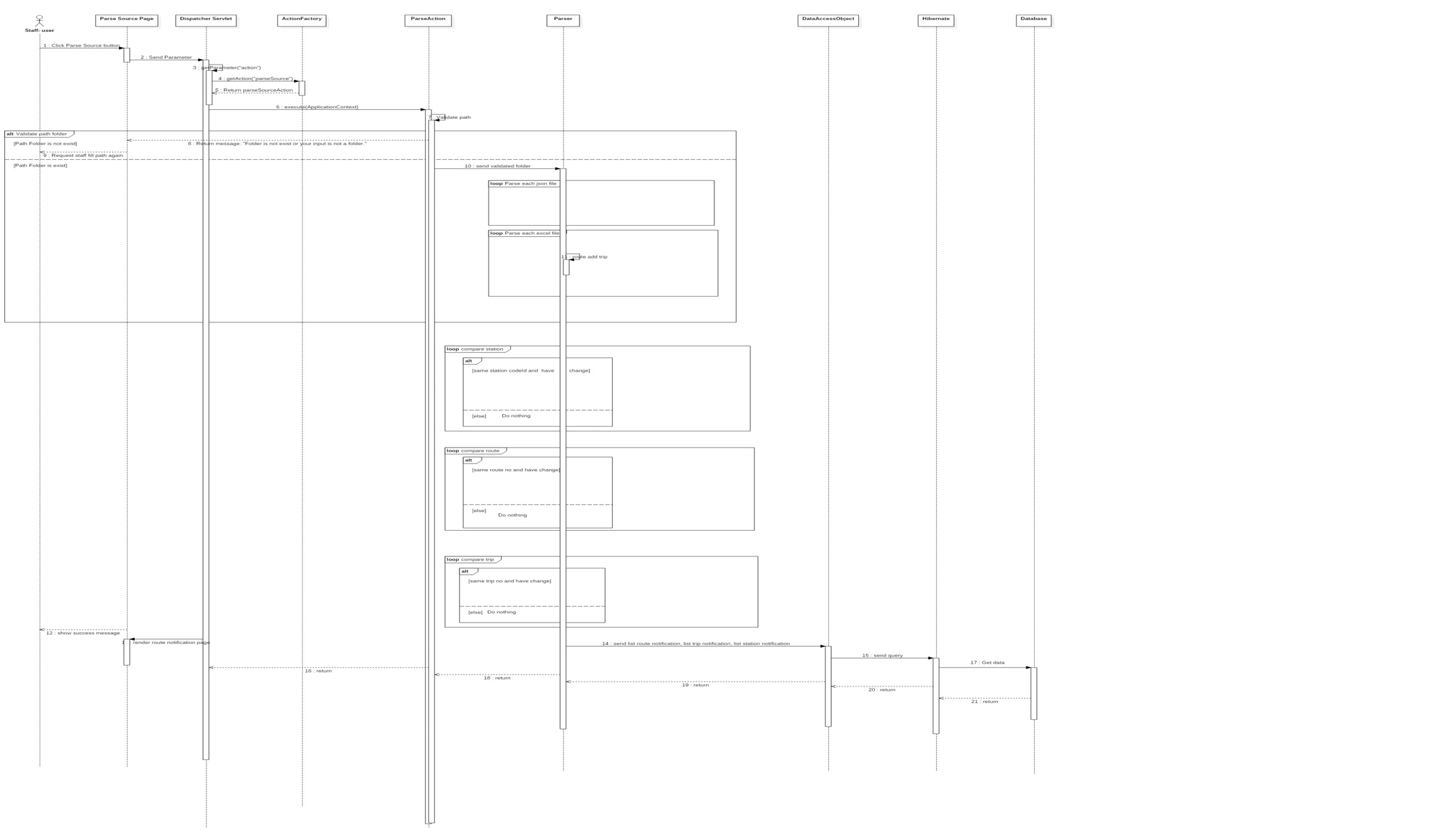
### Class Diagram Explanation

### Interactive Diagram

#### Web Application

##### Staff

###### Parsing source:

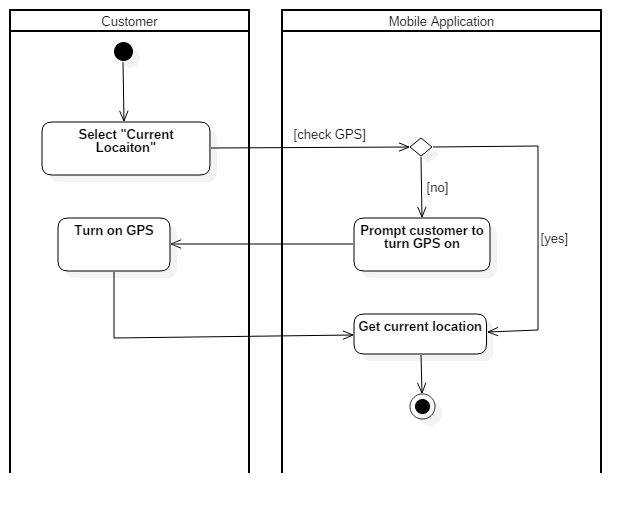
****

##### Admin

#### Mobile Application

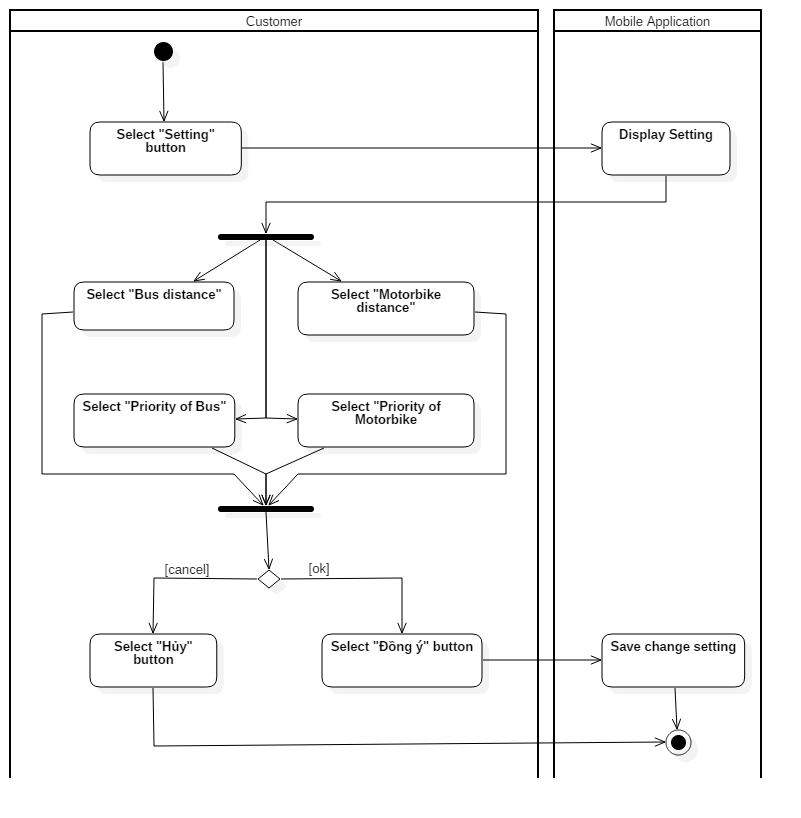
###### <Customer> Navigate Current Location on Map

* Summary: This diagram allows customer search current location on the map with GPS for mobile application.



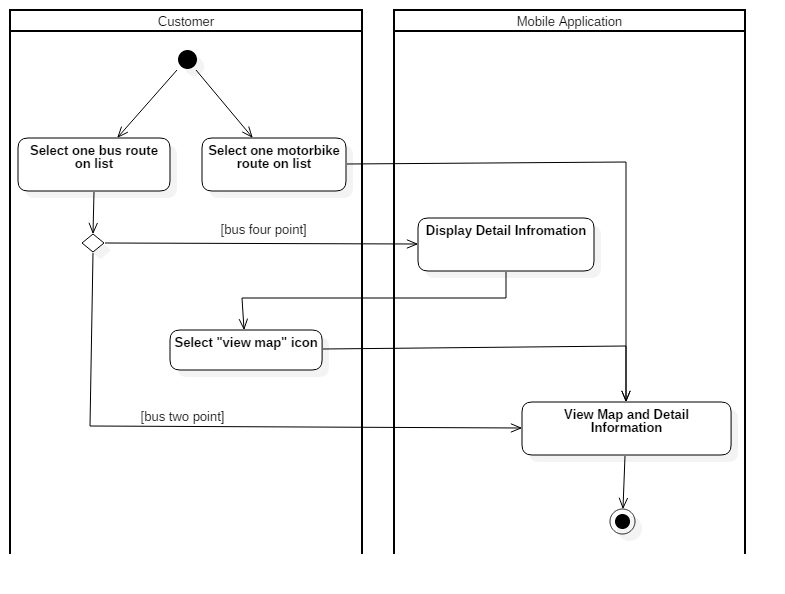
###### <Customer> Setting

Summary: This diagram allows the customer change setting from mobile application.



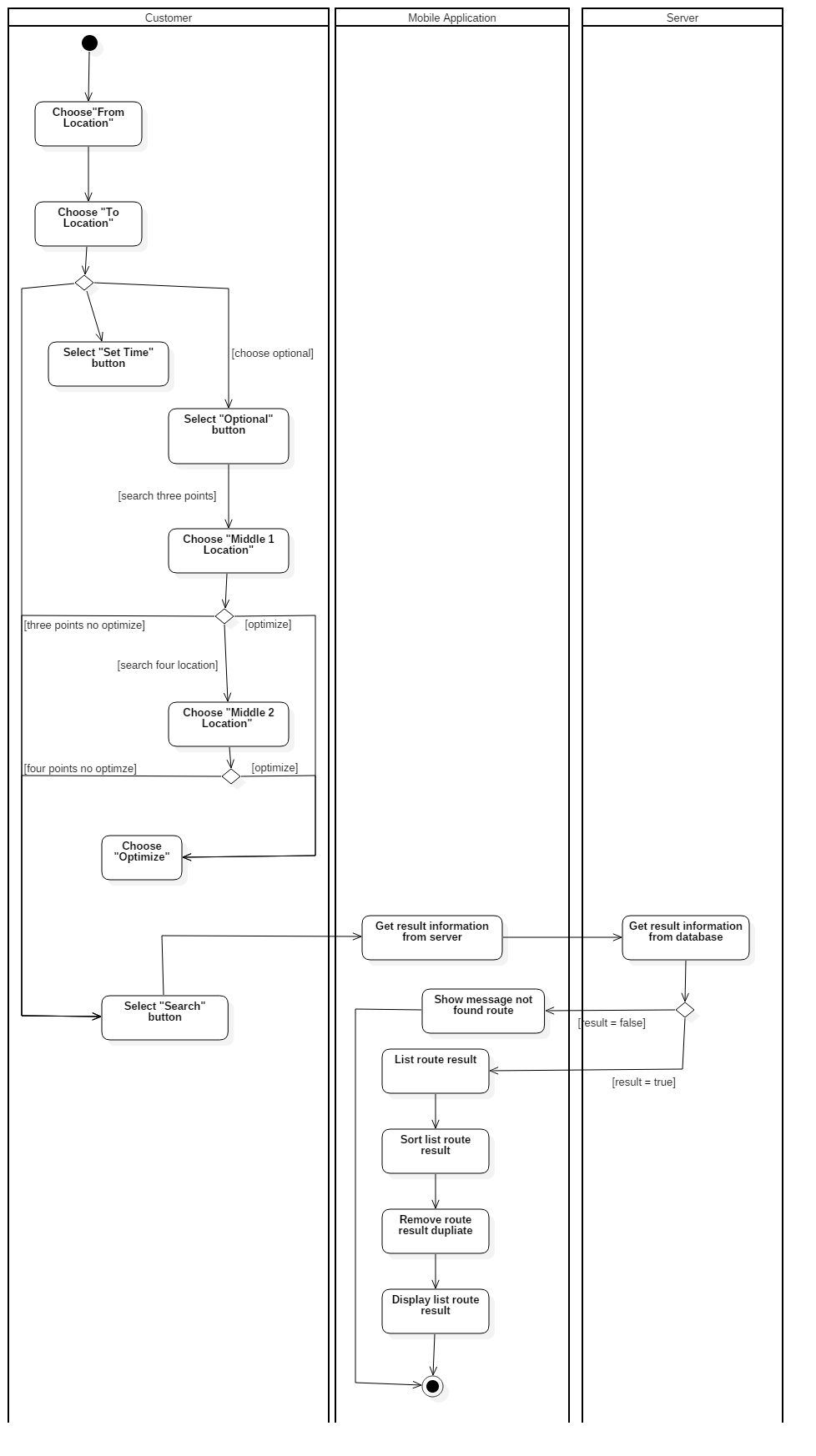
###### <Customer> View Detail

Summary: This diagram show detail information of route when customer select one route in list route.



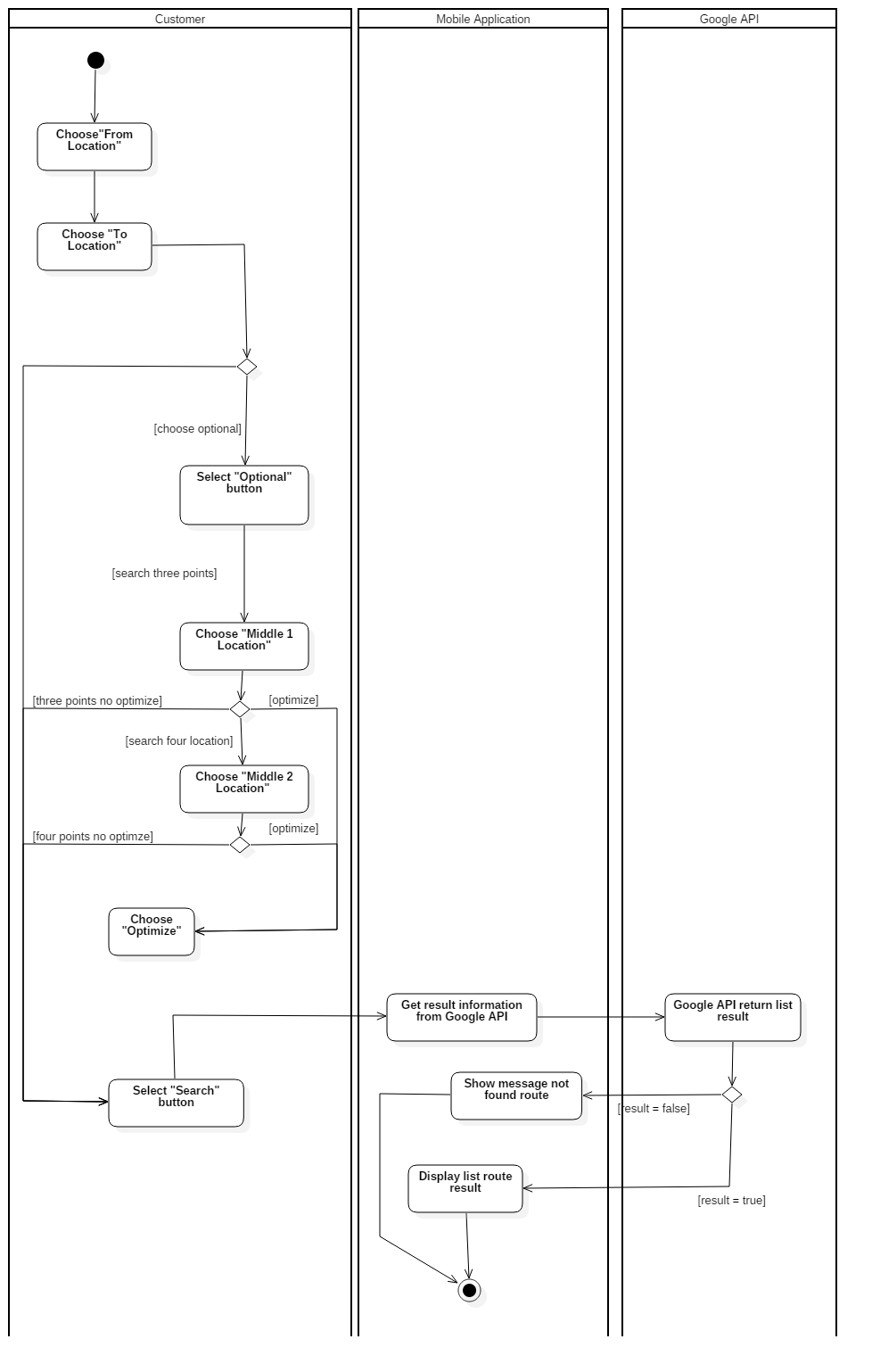
###### <Customer> Search Bus Route

Summary: This diagram allows customer search bus route through some locations (from two to four points) on the mobile application.



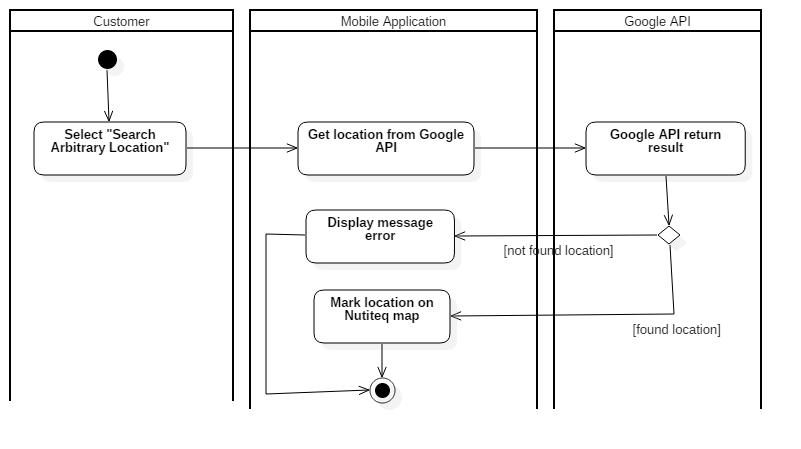
###### <Customer> Search Motorbike Route

Summary: This use case allows customer search motorbike route thought some locations (from two to four) on the mobile application.



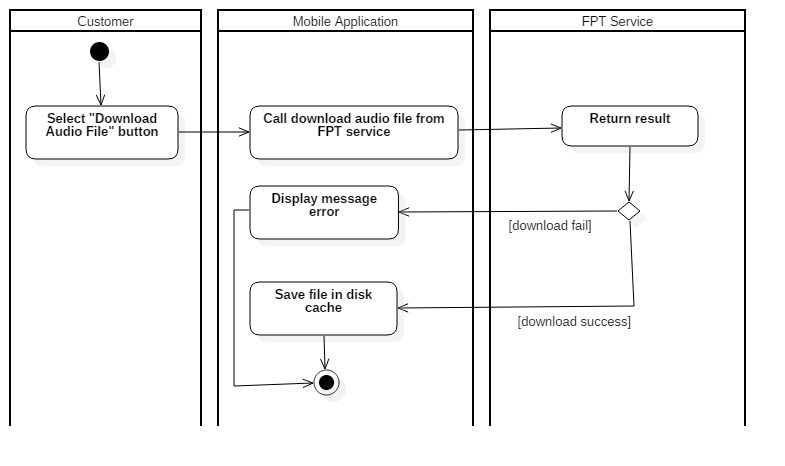
###### <Customer> Search Arbitrary location on map

Summary: This diagram allows customer search an arbitrary location on the map for the mobile application.



###### <Customer> Download Audio File

Summary: This use case allows the customer download audio file from FPT service for mobile application.



## Interface

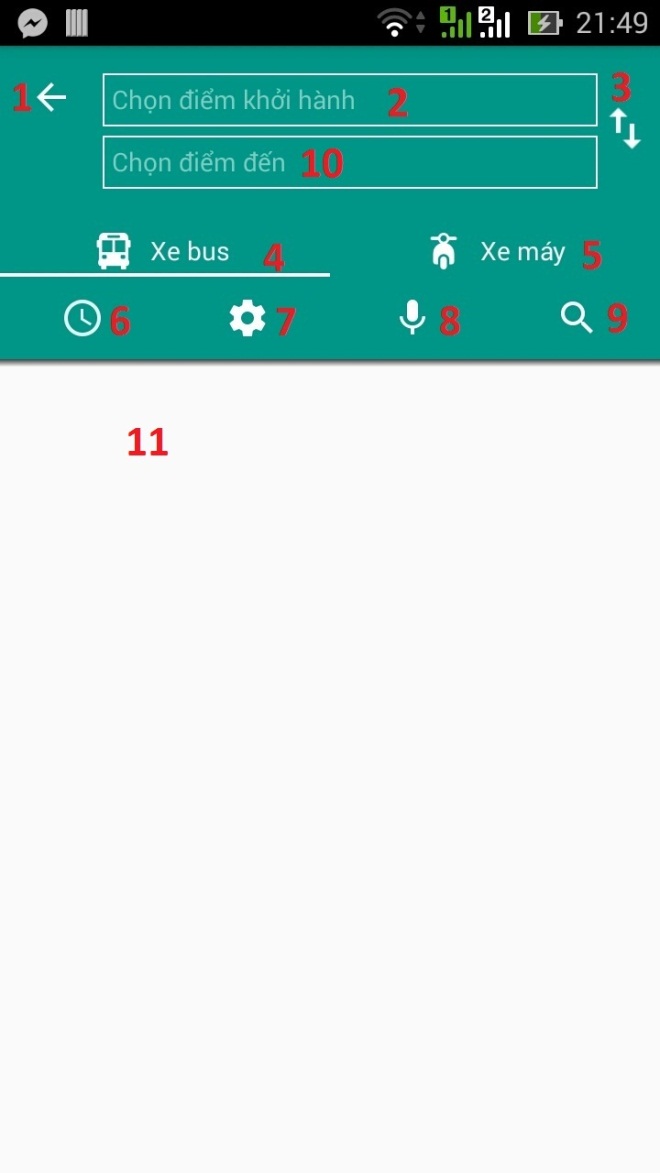
### Mobile Application Design

#### Main Screen



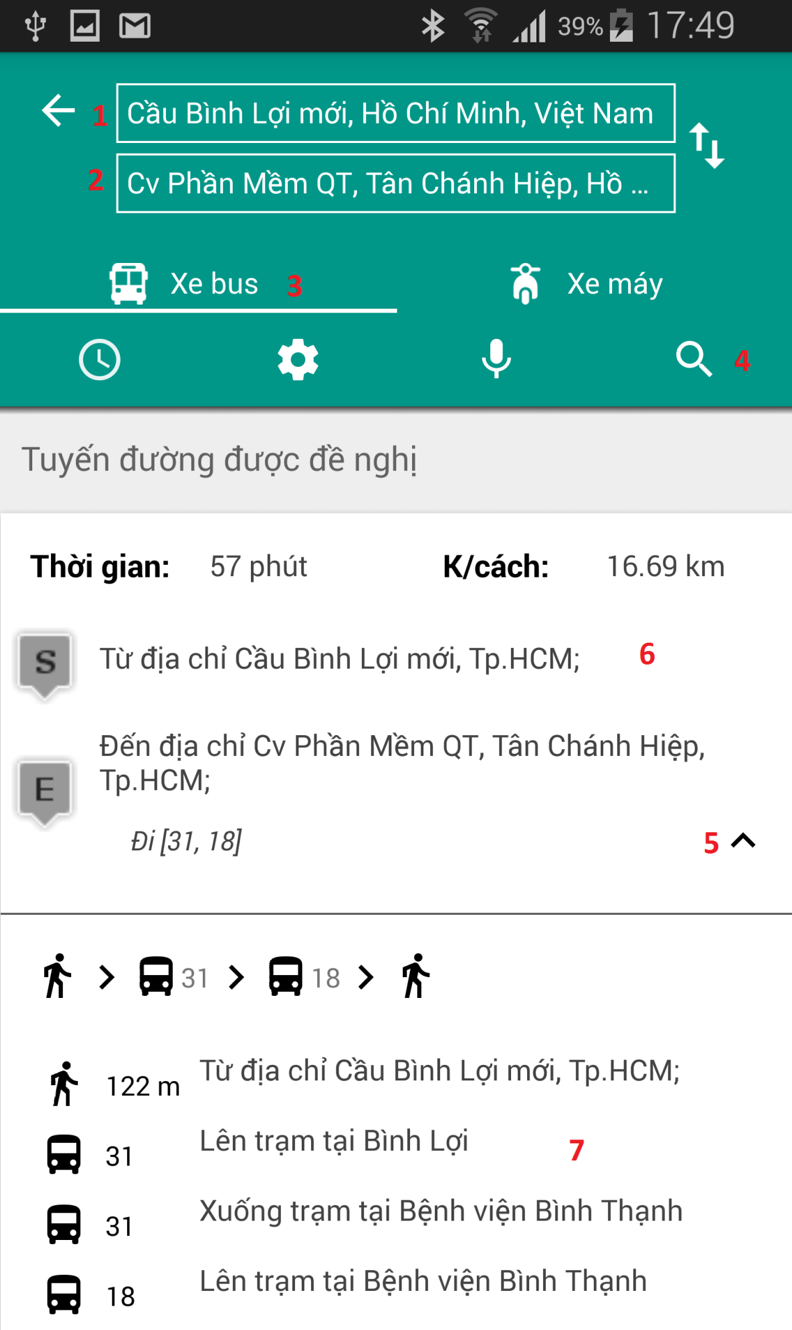
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button to display menu. |
| 2 | Customer touches on search “textview” to transfer to autocomplete search activity. |
| 3 | Customer touches on  button to transfer to search voice activity. |
| 4 | Customer touches on  button to get current location. |
| 5 | Customer touches on  button to start search route. |

#### Search Route



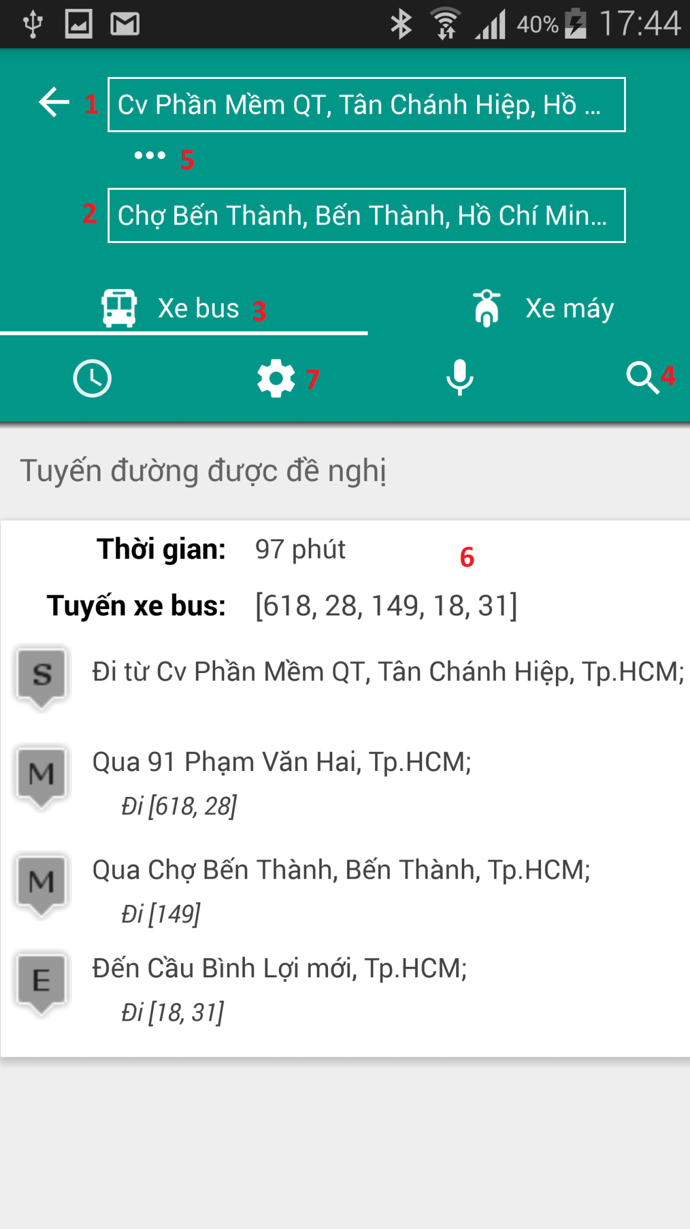
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button to return to previous activity. |
| 2 | Customer touches on search textview to transfer to autocomplete search activity and search start location. |
| 3 | Customer touches on  button to swap between start and end location. |
| 4 | Customer search with bus. |
| 5 | Customer search with motorbike. |
| 6 | Customer touches on  button to set time search route for bus (show dialog set time). |
| 7 | Customer touches on  button to add some optional for bus (refer to screen setting 5.1.12). |
| 8 | Customer touches on  button to transfer to search voice activity (refer to screen autocomplete search 5.1.11). |
| 9 | Customer touches on  button to search route. |
| 10 | Customer touches on search textview to transfer to autocomplete search activity and search end location. |
| 11 | Display result search route. |

#### Search Bus Two Point.



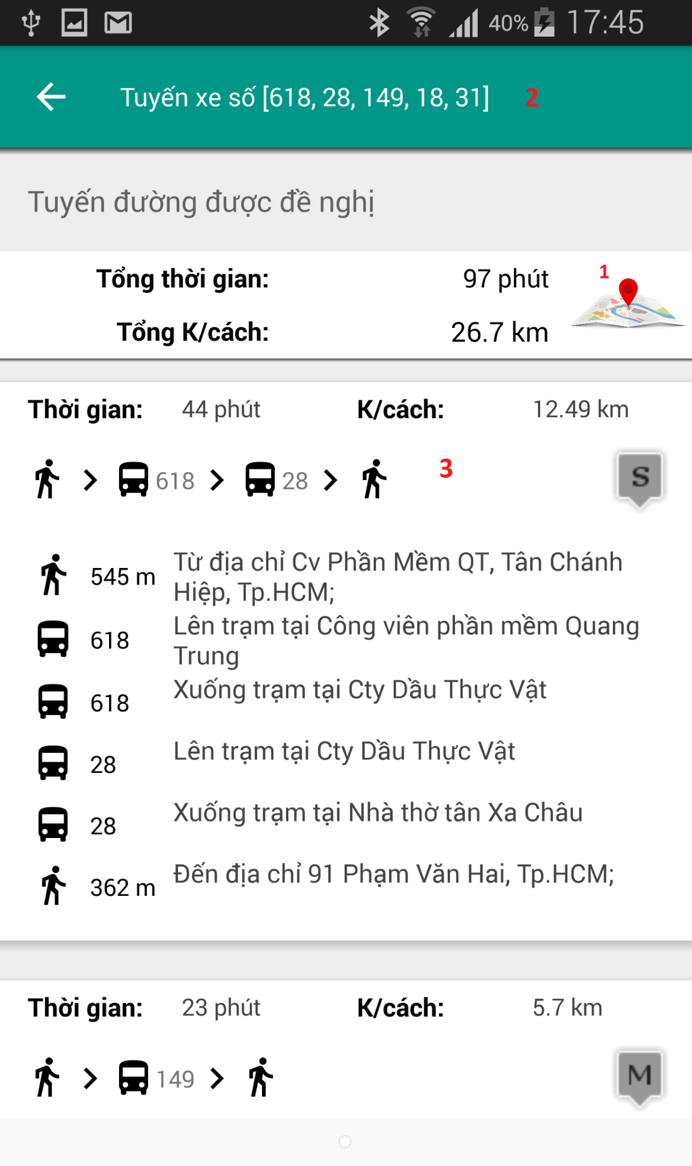
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer input start location. |
| 2 | Customer input end location. |
| 3 | Customer select bus option. |
| 4 | Customer touches on  button to search. |
| 5 | Customer touches on  button to open or close detail information for search bus two points. |
| 6 | Result when member search two points and when click result to transfer to view map (refer to view map bus 5.1.6). |
| 7 | Display detail information for bus two points. |

#### Search Bus Four Point.



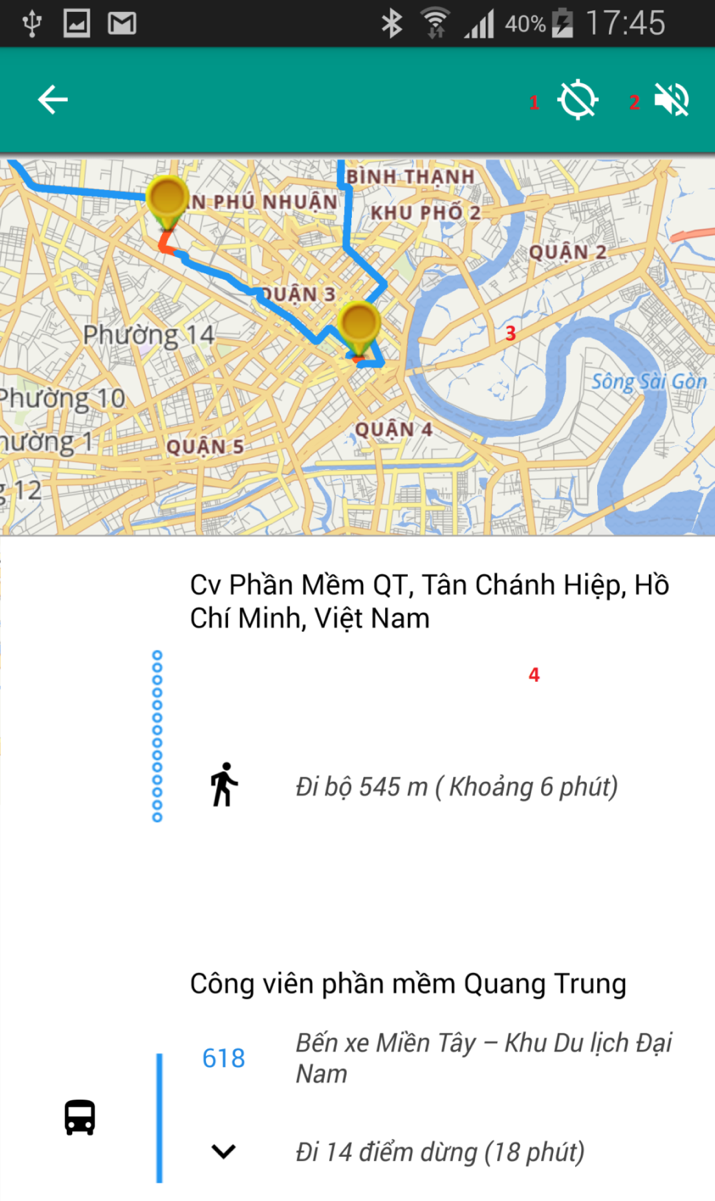
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer input start location. |
| 2 | Customer input end location. |
| 3 | Customer select bus option. |
| 4 | Customer touches on  button to search. |
| 5 | Customer touches on  button to show one or two middle location. |
| 6 | Result when member search three or four points and when click result to transfer to view detail information (refer to View Detail Result Bus Three or Four Point 5.1.5). |
| 7 | Customer touches on  button to input middle location (refer to Setting 5.1.12). |

#### View Detail Result Bus Three or Four Point.



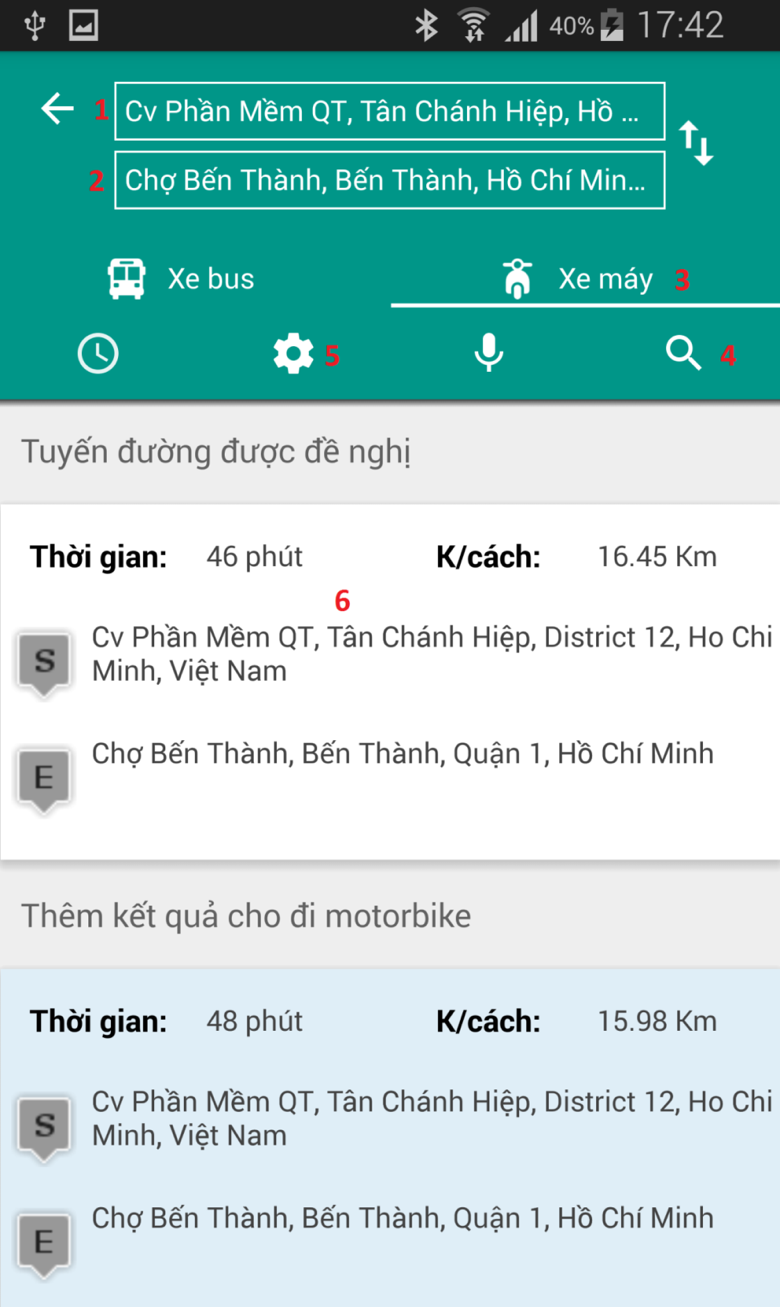
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on image button to transfer to view detail map (refer to View Map Bus Route 5.1.6). |
| 2 | Display number of bus route. |
| 3 | Display detail information when customer find with three or four point. |

#### View Map Bus Route.



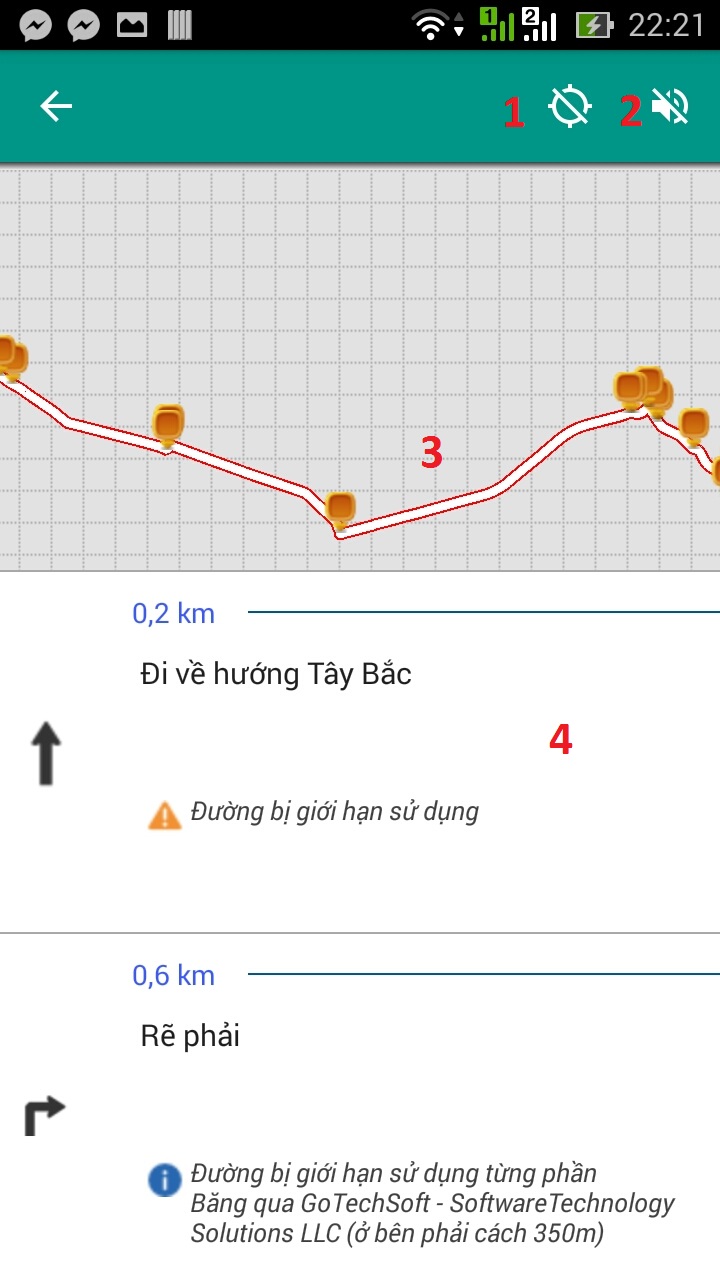
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button, application will change to “tracking mode”. In tracking mode, application will periodically get system’s GPS and compare to search route. So when user is near a bus station or a turn, application will notify message for user to know. |
| 2 | Customer touches on  button to download voice. |
| 3 | Drawing route on map. |
| 4 | Display detail information for result. |

#### Search Motorbike



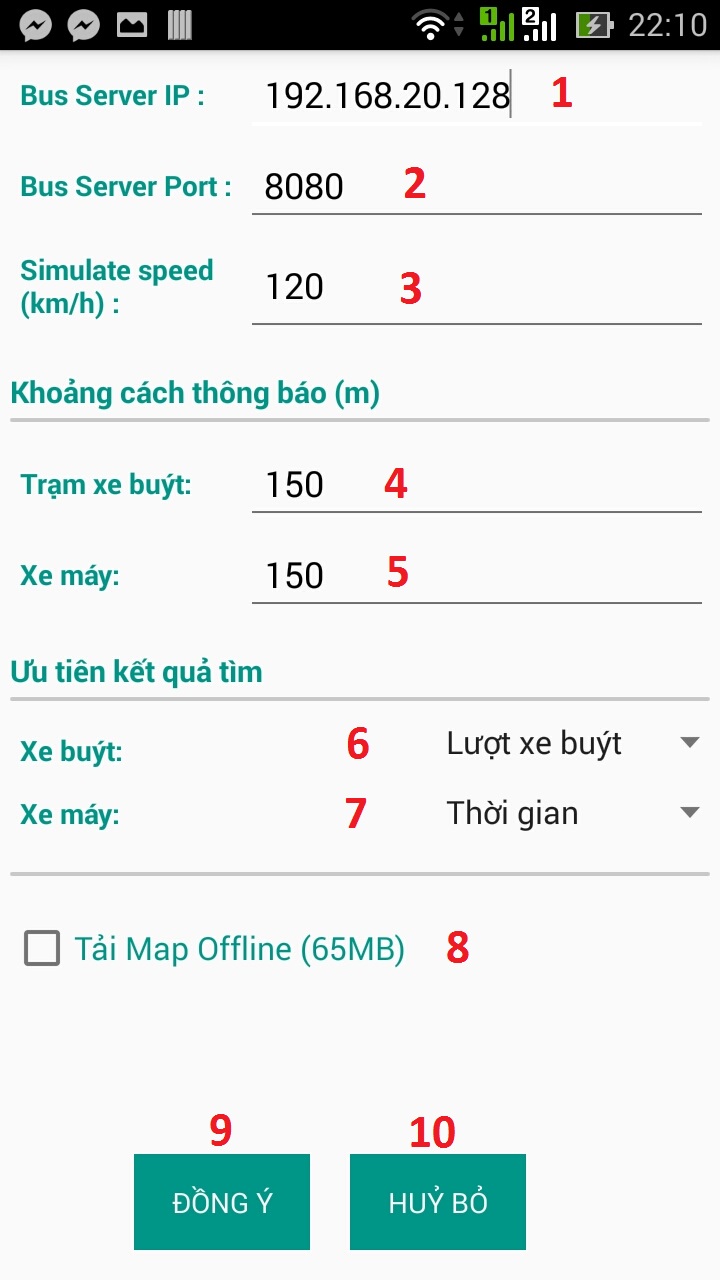
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer input start location. |
| 2 | Customer input end location. |
| 3 | Customer select motorbike option. |
| 4 | Customer touches on  button to search. |
| 5 | Customer touches on  button to input middle location (refer to Setting 5.1.12). |
| 6 | Display information for result and when touches one result display map (refer to View Map Motorbike 5.1.8) |

#### View Map Motorbike



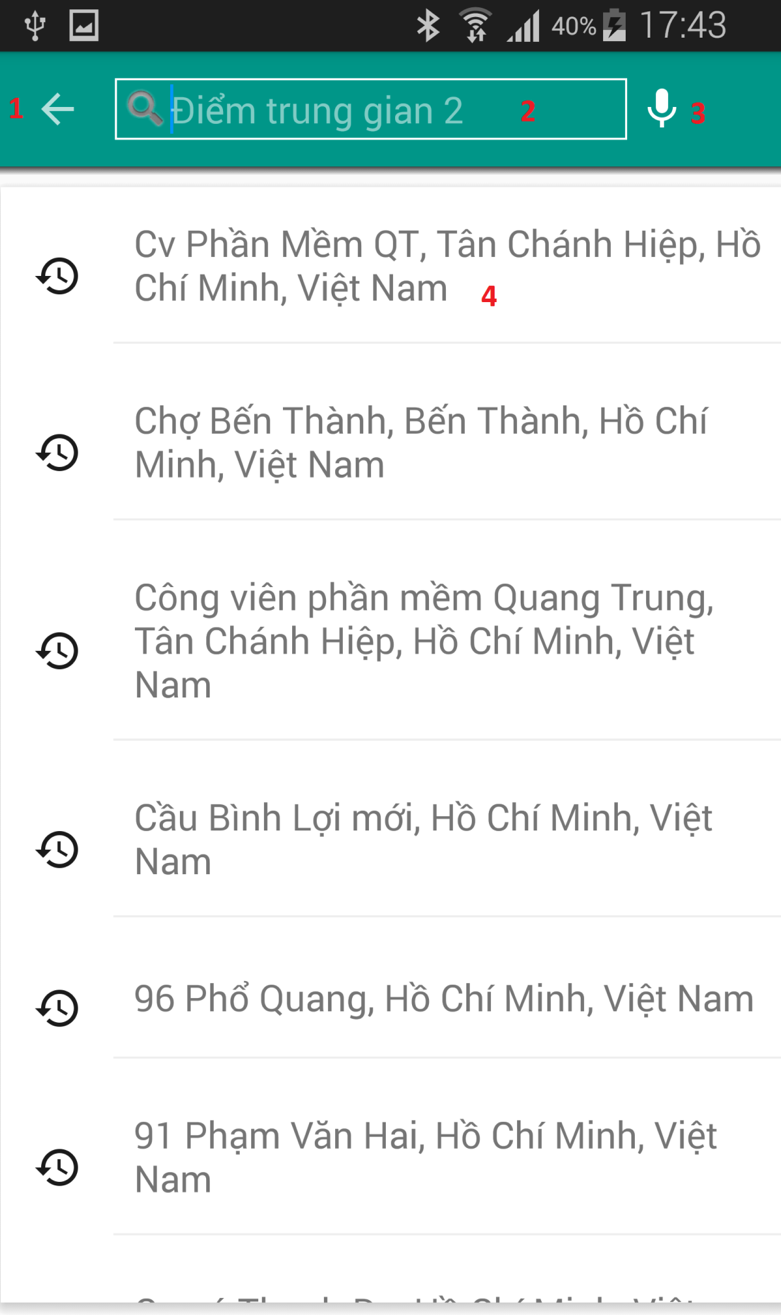
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button, application will change to “tracking mode”. In tracking mode, application will periodically get system’s GPS and compare to search route. So when user is near a bus station or a turn, application will notify message for user to know. |
| 2 | Customer touches on  button to download voice. |
| 3 | Drawing route on map. |
| 4 | Display detail information for one result. |

#### Setting



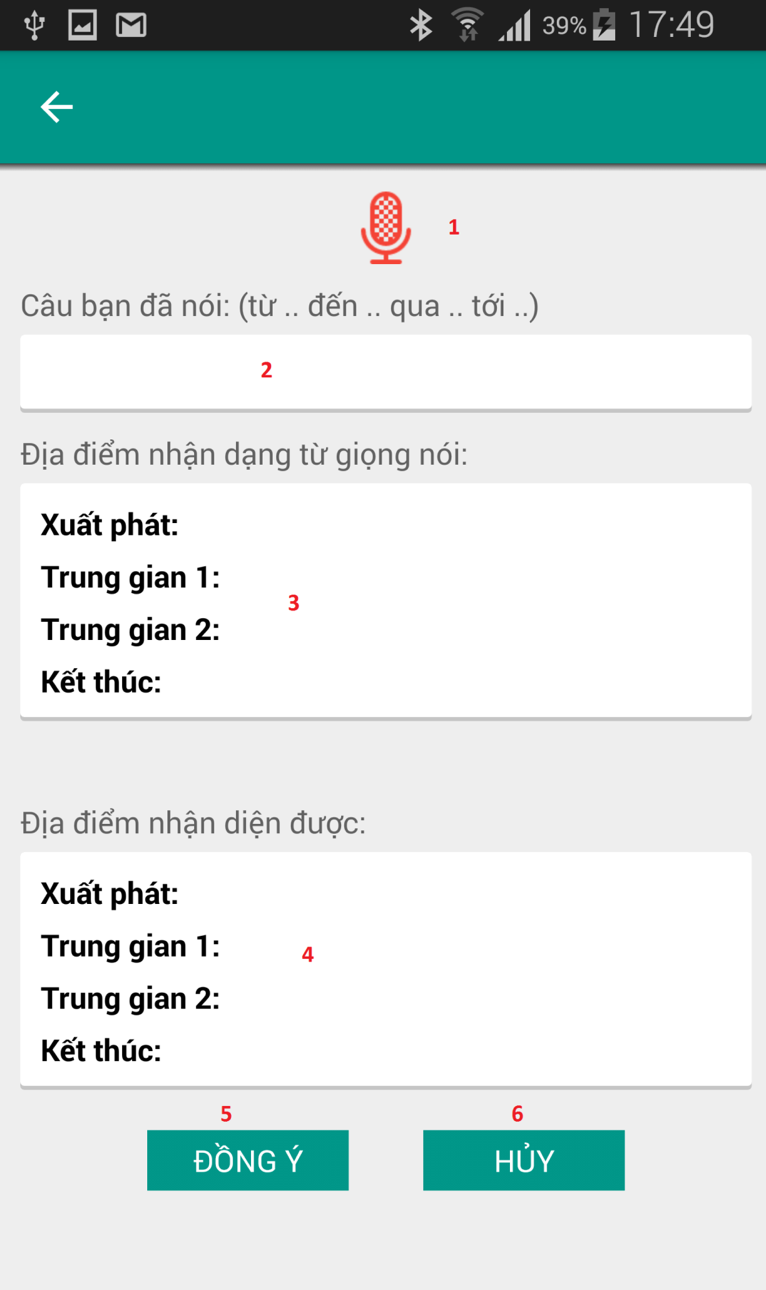
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer set IP address to access to server. |
| 2 | Customer set port to access to server. |
| 3 | Customer set speed (Km/h) to simulate of motorbike or bus on map. |
| 4 | Customer set distance to notify when near to bus station. |
| 5 | Customer set distance to notify when near to step (instruction) goes to route. |
| 6 | Customer select option to display priority result according (Lượt xe bus, Thời gian, Khoảng cách). |
| 7 | Customer select option to display priority result according (Thời gian, Khoảng cách). |
| 8 | Customer checks to download map offline. |
| 9 | Customer touches on “Đồng ý” button to accept change. |
| 10 | Customer touches on “Hủy bỏ” button to cancel change. |

#### Autocomplete Search



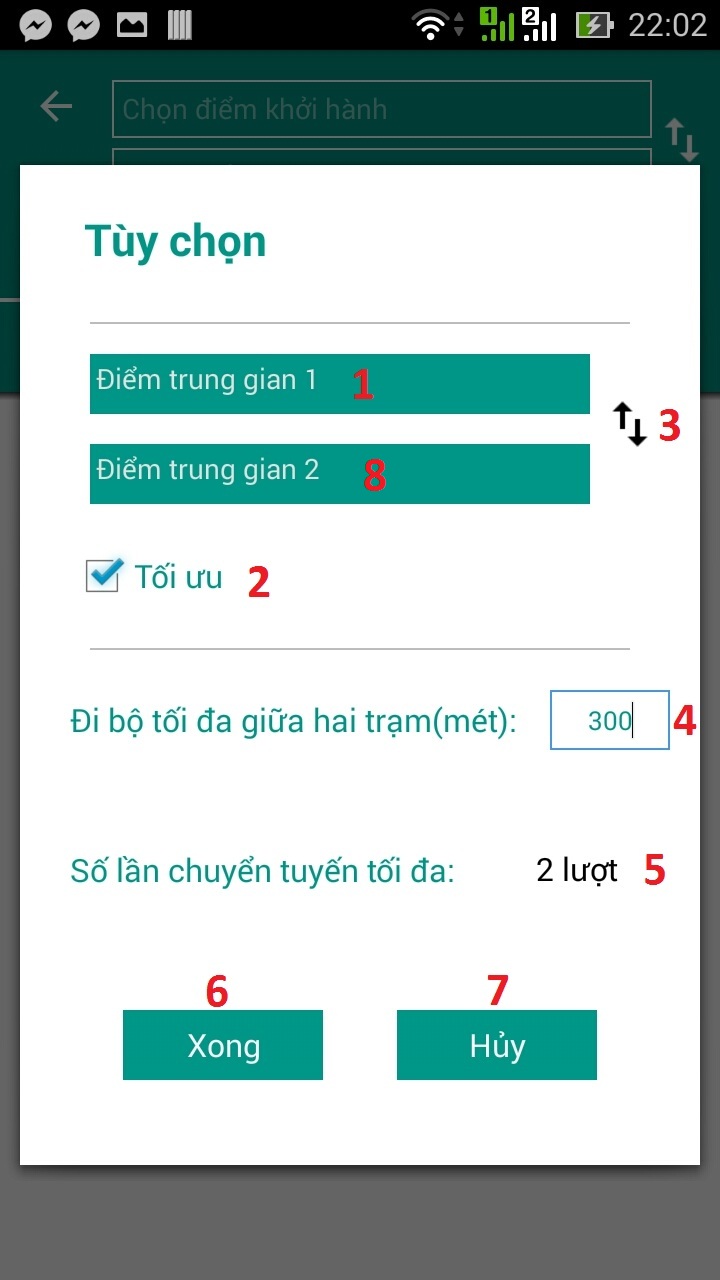
|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button to return previous activity. |
| 2 | Customer input route information search. |
| 3 | Customer touches on  button to search voice activity (refer to Search with Voice 5.1.11). |
| 4 | Display list history and result search when Customer input route information. |

#### Search with Voice



|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on  button to start search with voice. |
| 2 | Display sentence customer say. |
| 3 | Display location identification from sentence customer say. |
| 4 | Display location result. |
| 5 | Customer touches on “Đồng ý” button to accept change. |
| 6 | Customer touches on “Hủy” button to cancel change. |

#### Optional



|  |  |
| --- | --- |
| No. | Description |
| 1 | Customer touches on search textview to transfer to autocomplete search activity and search “middle 1” location. |
| 2 | Customer touches checkbox to enable or disable option optimize find route. |
| 3 | Customer touches on  button to swap between “middle 1” and “middle 2” location. |
| 4 | Customer can input distance walking maximum between two stations (m). System assigns default 300m walking. |
| 5 | Customer can change number of transit maximum. System assigns default “2” transit. |
| 6 | Customer touches on “Xong” button to accept change. |
| 7 | Customer touches on “Hủy” button to cancel change. |
| 8 | Customer touches on search textview to transfer to autocomplete search activity and search “middle 2” location. |

# G. Appendix

1, RAPTOR algorithm is based on paper “Round-Based Public Transit Routing” written by Daniel Delling, Renato F. Werneck (Microsoft Research Silicon Valley), Thomas Pajor (Karlsruhe Institute of Technology), public in 2012.

1. [↑](#endnote-ref-1)