# 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 routing through more than two points. For example, Google Map and BusMap just supports on routing 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 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. 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 route through more than two points.
* BusMap doesn’t support using smart watch for finding route, 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:

* Find 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.

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 and will notify again one minutes later with special sound and vibrate the smart watch until user out of range.
* 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.
* 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 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:

* Synchronize data from server to mobile.
* 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.
* 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.

Bus Driver Component:

* Record time when bus driver arrived to one bus station.
* Synchronize data to server.

## 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 smart wear. In addition, we also provide mobile application for bus driver so system can get arrival time at each station.

### 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.
* **Mobile application stimulator for bus driver:** In developing time, we need arrival time data at each station for our algorithm. So we build a mobile stimulator for bus driver for collecting time arrival at each station.
* **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.

##### 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).

##### Bus Driver Mobile Application

This is a stimulator for bus driver to collect arrival time at each station and send to server:

* Get arrival time of each bus station and save to local storage.
* Combine that result to get average time and send to server.

#### 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.
* Checker mobile application for bus drivers.

#### 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 function includes:
    - View current location on map.
    - View bus route information.
    - View bus timetable information.
* Synchronize with database server.
* Configuration.
* Wear application includes:
* Map function includes:
  + View current location on map.
  + View a bus station in bus route list on map.
  + View a turn in motorbike turn list on map.
  + Navigate a bus station on map.
  + Navigate a motorbike route on map.
* View list of bus routes should to go.
* View list of motorbike turns should to go.
* 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.

### 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.
* Bus driver uploaded data ‘s timetable management function includes:
  + View all uploaded bus timetables information.
  + View detail of a bus route timetable information.
  + Approve or reject all uploaded bus timetables.
  + Approve or reject one detail of a bus route timetable.
* 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 notification. So all bus route 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.
  + Block one kind of notification. So that bus route or timetable information will not show in the future.
  + View list of block notification.
  + Unblock notification in block list. So that bus route timetable information can be notified again when data change.
* Configuration:
  + Turn on or turn off parser.
  + Set up parser for getting data from files or from web.

### Bus driver requirement

Bus driver is people who drive bus and help our system collect time arrival at each station more accurate. Bus driver can use some following functions:

* Collecting time arrival at each station.
* View time arrival at each station.
* Synchronize data to server.
* Configuration:
  + Set up time for auto synchronization.
  + Set up GPS time accuracy.

## 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 “\*”. * Encrypt password before being send to server. * 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 | Actor Action | 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 logs staff out automatically. * When staff logged out from system, system will redirect staff to login view. | | | |

Table 19: Use case WS01 - <Staff> Logout

###### <Staff> Approve or 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 notification will show on block and each notification includes:   + Content message – link to view detail notification.   + Action:     - Approve notification.     - Reject notification.     - Unblock notification.   + Time of arrived notification. | | 2 | Staff sends approve all command.  [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 change to de-active state, 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. * Current notification state (pending, solve, de-active) | | | |

Table 20: Use case WS02 - <Staff> Approve or reject all current system notifications.

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



Figure 9: <Staff> Approve or reject detail notification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS03 | **Use Case Version** | 2.0 |
| Use Case Name | Approve or reject detail notification | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to approve or reject detail notification, and this function will implement on web application.   Goal:   * System provides approve or reject detail notification for supporting staff manage detail 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.   + Time’s type: start time or end time.   + Old notification.   + New notification.   + 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 changed to reject state. 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 second. * Notification structure includes: * Date created. * Bus route or bus time change. * Current notification state (pending, solve, de-active) | | | |

Table 21: Use case WS03 - <Staff> Approve or reject detail notification.

###### <Staff> Block and Unblock notification



Figure 10: <Staff>Block and Unblock notification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS04 | **Use Case Version** | 2.0 |
| Use Case Name | Block or unblock notification. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to block or unblock notification and this function will implement on web application.   Goal:   * System provides block or unlock notification function for supporting staff to manage notification.   Triggers:   * Staff sends block or unlock notification command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: Notification will block or unlock. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 2 | Staff sends block or unlock notification command. | System block or unlock notification, which requested.  System show successful message. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Successful message:   + For notify about route timetable, content is “Notification for timetable of {route name} is blocked (or unblocked)”.   + For notify about bus route, content is “Notification for bus {route name} is blocked (or unblocked)”. * The initial status of notification is “Pending”. * When staff block notification, status of notification will change “Pending” to “Blocked”. * When staff unblock notification, status of notification will change “Blocked” to “Activated”. | | | |

Table 22: Use case WS04 - <Staff> Unlock notification.

###### <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:   * System just supports parse json file and excel file (“.xls”, “.xlsx”). * In presently, system just supports for map of Ho Chi Minh bus. | | | |

Table 23: Use case WS05 - <Staff> Configure parsing source.

###### <Staff> Configure parsing time



Figure 12: <Staff> Configure parsing time

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS06 | **Use Case Version** | 2.0 |
| Use Case Name | Configure parsing time. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to configure parsing time and this function will implement on web application.   Goal:   * System provides configure parsing time for parser knows when to parse.   Triggers:   * Staff sends configure parsing time command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: System records configured time and scheduler will parse at configured time. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to configure parsing time view. | Configure parsing time view shown with following information:   * Date Picker. * Repeat day. | | 2 | Staff configure time. |  | | 3 | Staff sends configure parsing time command. | System records configured time and call scheduler for parsing at configured time (reference use case Parse data periodically). |   Alternative Scenario: N/A.  Exceptions: N/A.  Relationships: N/A.  Business Rules:   * Default time for configuration is 0:00 AM. * Time for configuration must be from 11:00 PM to 11:59 PM, 0:00 AM to 04:00 AM. | | | |

Table 24: Use case WS06 - <Staff> Choose parsing time.

##### <Scheduler> Overview Use Case

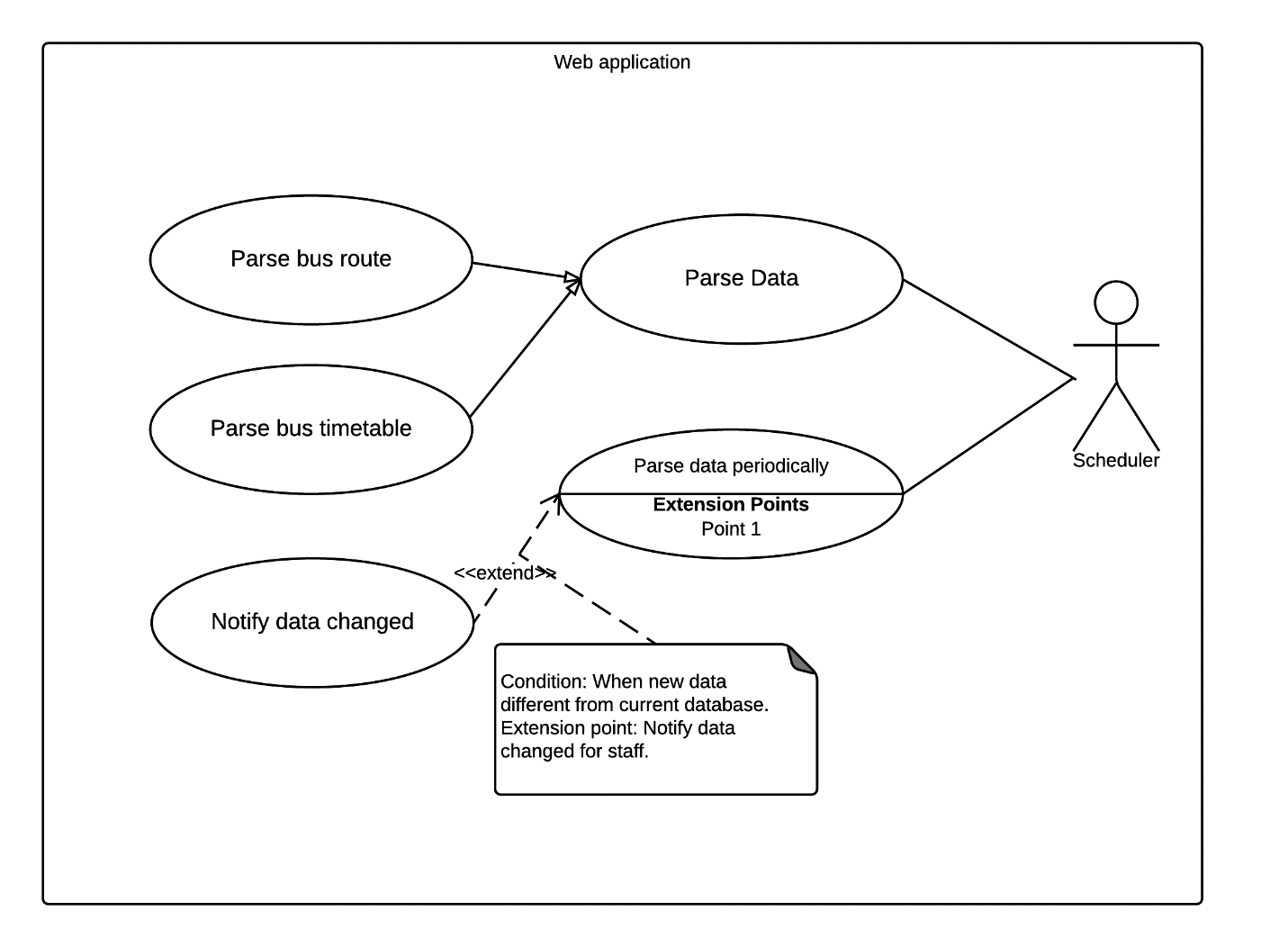


Figure 13: <Scheduler> Overview Use Case

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



Figure 14: <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. * 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 25: Use case SC01 - <Staff> Configure parsing source.

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



Figure 15: <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 an template file. |   Relationships: N/A.  Business Rules:   * File for parsing must correct following template file:      * + System currently just support excel file format.   + 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 26: Use case SC02 - <Staff> Parse bus timetable.

###### <Scheduler> Parse data periodically

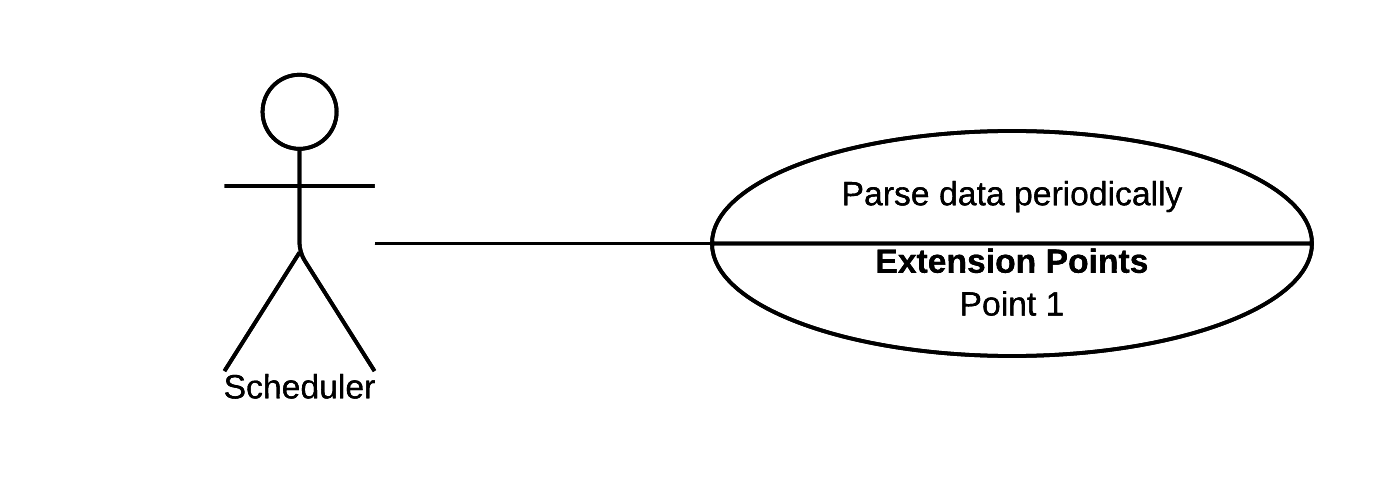


Figure 16: <Staff> Parse data periodically

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – SC03 | | | |
| Use Case No. | SC03 | **Use Case Version** | 2.0 |
| Use Case Name | Parse data periodically. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows scheduler to parse data periodically and this function will implement on web application.   Goal:   * The system provides parse data periodically for automating get source when periodically time arrive. Therefore, system will always keep data latest with third-party server.   Triggers:   * Scheduler sends parse data periodically command.   Preconditions:   * Time has configured.   Post Conditions:   * Success: Show successful message. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Scheduler sends parse data periodically command. | - When periodical time arrive, system calls parse bus route or parse bus timetable (reference use case *<Scheduler> Parse bus route* and *<Scheduler> Parse bus timetable*). |   Alternatives Scenario: N/A  Exceptions: N/A.  Relationships: N/A.  Business Rules:   * When finish parsing, system will notify message for staff. * Parsing time from 11PM to 11:59PM and 0:AM to 4:00 AM. | | | |

Table 27: Use case SC03 - <Staff> Configure parsing source.

##### <Admin> Overview Use Case

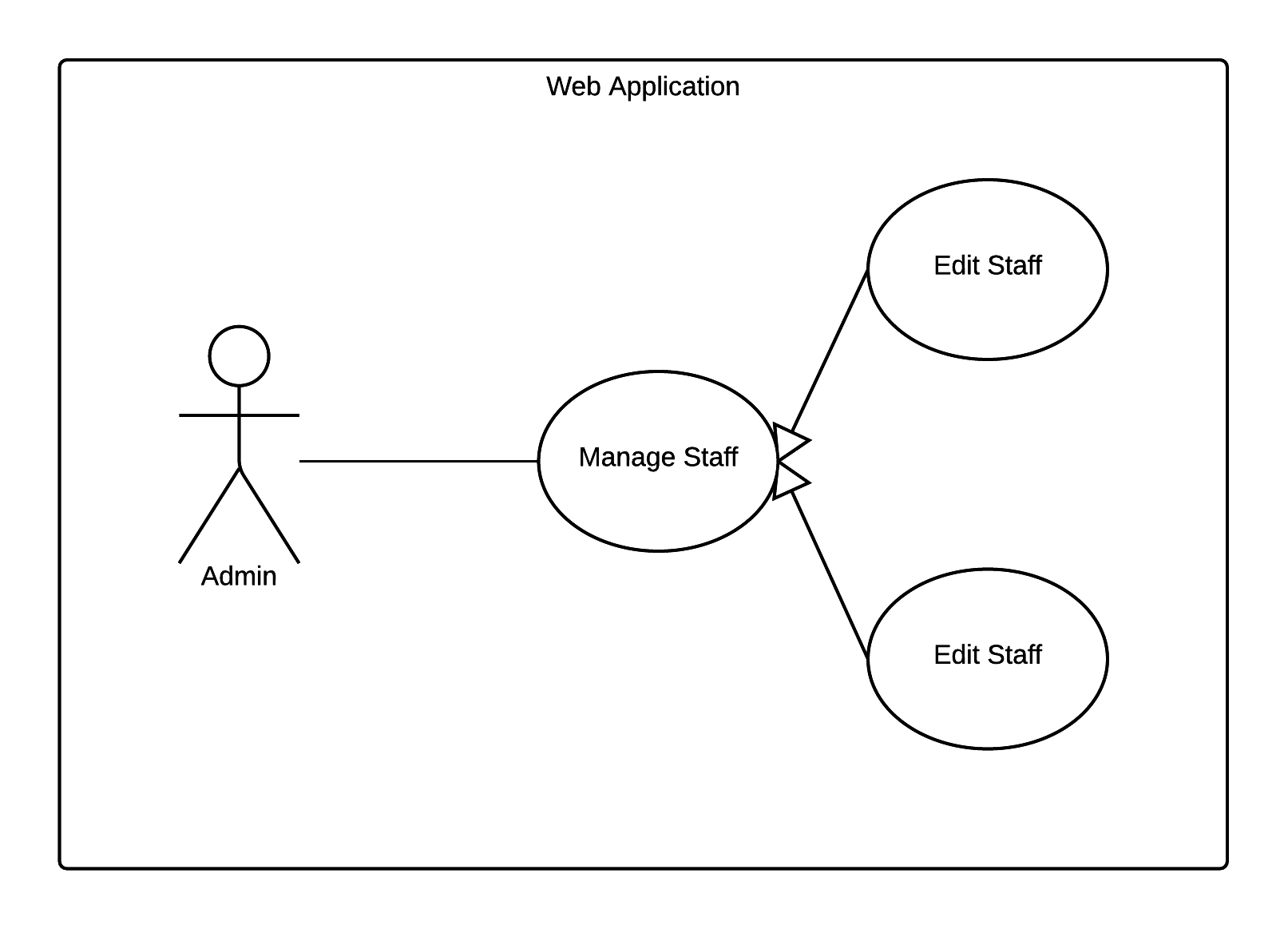
. 

Figure 17: <Admin> Admin Overview Use Case

###### <Admin> Add staff

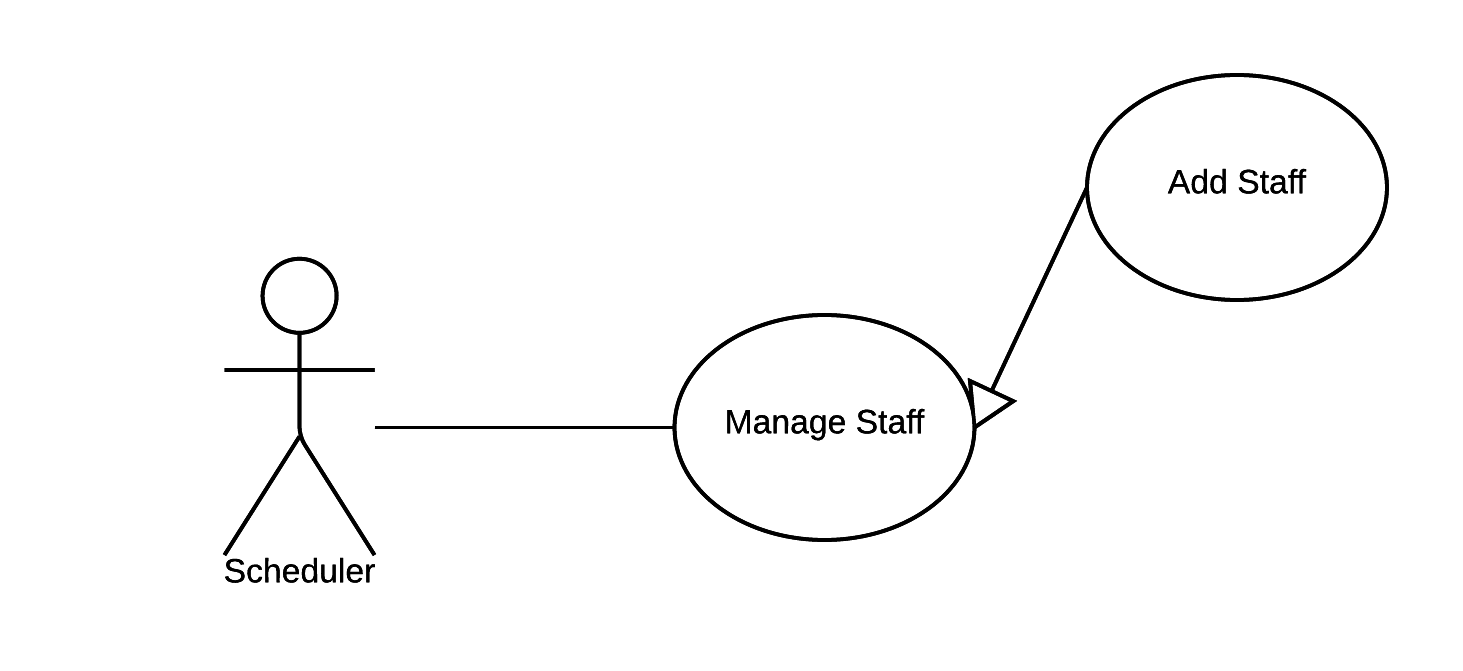
**

Figure 18: <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 display 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 | Actor Action | 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 send to server. * Password must be encrypted before save to database. | | | |

Table 28: Use case WS07 - <Staff> Add staff.

###### <Admin> Edit staff

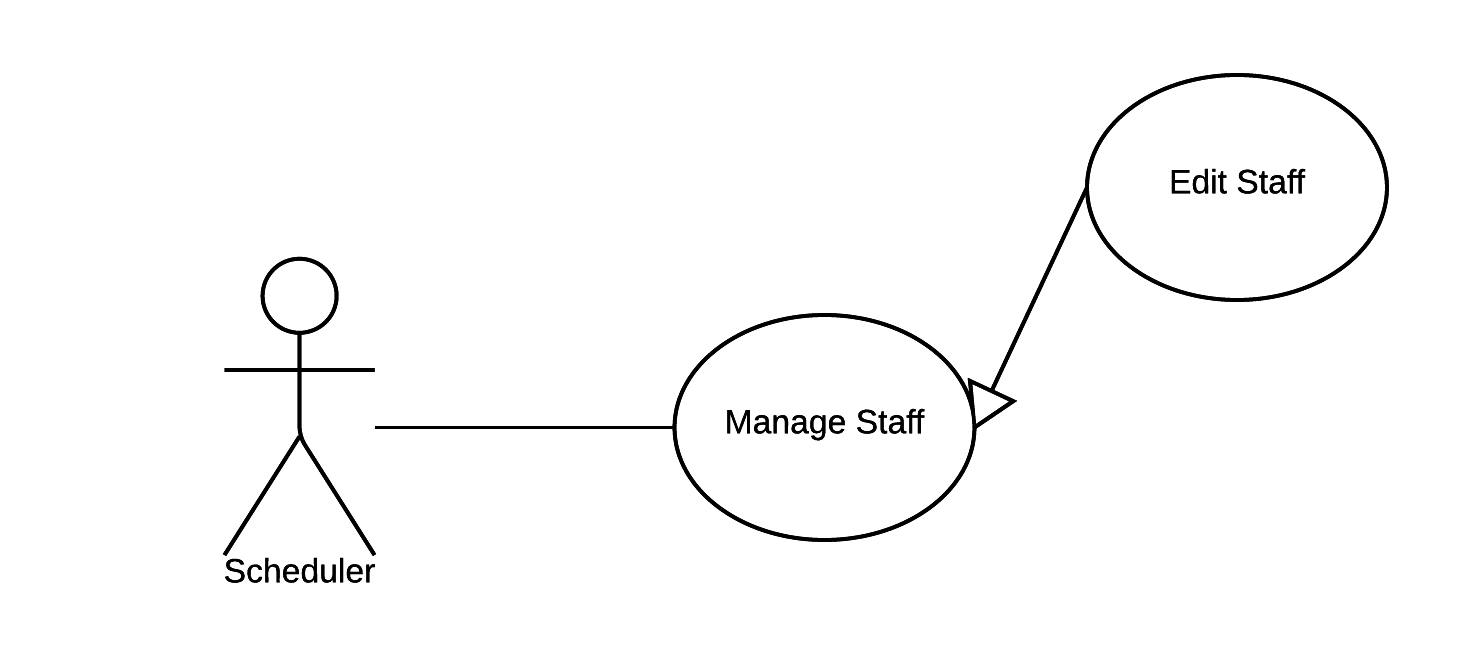


Figure 19: <Admin> Edit staff

|  |  |  |  |
| --- | --- | --- | --- |
| 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 role. | | 2 | Admin selects 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 | Actor Action | 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 send to server. * Password must be encrypted before save to database. | | | |

Table 29: Use case WS08 - <Staff> Add staff.

#### 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 and gives detailed information about the location. 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 and detailed information about it. * 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 | Actor Action | 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 Google Map for rendering a place on 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. * Searched 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 | | 2 | When customer select start location input text or end location input text. | Display new view for selecting place with following information:   * Location: free text input. * Current location by using GPS function. * Location lists from history. | | 3 | If customer inputs start or end location.  [Alternative 1,2] | Input field shows location that customer has selected. | | 4 | The customer goes to choose the departure time view.  [Alternative 3] | 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: * “Tuyến nhanh nhất” * “Số lần chuyển tuyến tối đa” * “Số m đi bộ tối đa giữa hai trạm” | | 7 | The customer can input two middle locations.  [Alternative 1,2] | 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 one of three values from:   * “Tuyến nhanh nhất” * “Số lần chuyển tuyến tối đa” * “Số m đi bộ tối đa giữa hai trạm” | System will show checkbox for the option user has selected. | | 9 | Customer sends search bus route command. | System shows a list with some optimal paths.  [Exception 1,2,3] | | 10 | 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 | If customer chooses current location by using GPS function. | System will show the current location on free input text. |   [Alternative 3]   |  |  |  | | --- | --- | --- | | 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. |   Relationships: Has an extension optimize search for bus route.  Business Rules:   * System uses Google 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: * Tuyến nhanh nhất: system will find optimal path minimum time from start location to end location. * Số lần chuyển tuyến tối đa: system will find fasted path that number of transfer doesn’t exceed certain number (input by user). * Khoảng cách di chuyển 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. | | | |

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 map and gives detailed information about the current location. 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 and detail information about it. * 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 map and shows detail information.  [Exception 1,2,3] |   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 internet yet. |   [Exception 3]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 3 | Server responses error with some error code. | If error code is:   * UNKNOW\_ERROR: 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: shows customer message that server currently cannot support more queries. |   Relationships: N/A  Business Rules:   * Using Google map to display the location. * Using Google Maps Geocoding API for converting latitude and longitude to normal string address. * 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 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 | | 2 | When customer select start location input text or end location input text. | Display new view with following information:   * Location: free text input. * Location with GPS. * Location from history. | | 3 | If customer inputs start and end location.  [Alternative 1,2] | 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: * “Tuyến ngắn nhất” * “Số lần chuyển tuyến tối đa” * “Số m đi bộ tối đa giữa hai trạm” | | 7 | The customer can input two middle locations.  [Alternative 1,2] | 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 sends a command search. | System shows a list with some optimal paths.  [Exception 1, 2, 3] | | 10 | 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 | | 2 | If customer chooses a location with GPS. | System processes and input field shows the location that customer has selected. |   [Alternative 3]   |  |  |  | | --- | --- | --- | | 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:   * Using Google map for rendering route. * 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> Reference Bus Route Detail



Figure 25: <Customer> Reference Bus Route Detail

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB05 | | | |
| Use Case No. | MB05 | **Use Case Version** | 2.0 |
| Use Case Name | Reference Bus Route Detail | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer to show detail of all bus route on mobile application.   Goal:   * The system displays a list bus route detail for reference. So customer can view bus route information before participating traffic.   Triggers:   * Customer sends reference bus route detail command.   Preconditions:   * N/A   Post Conditions:   * Success: System displays a list of detailed bus route on the screen. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to bus route list view. | System displays a list of bus route. | | 2 | Customer sends reference bus route command with bus route number parameter. | System displays bus route detail include:   * Route number. * Route name. * List of stations in this bus route include: * Bus station name. * Bus station location. * Bus Station Street.   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | If no bus route was founded. | System returns empty route list. |   Exceptions: N/A  Relationships: Has an extension “Reference Bus Timetable Detail”  Business Rules:   * The information is loaded from local database. * Bus routes just supported in HCM city. | | | |

Table 34: Use case MB05 - <Customer> Reference Bus Route Detail

###### <Customer> Reference Bus Timetable Detail



Figure 26: <Customer> Reference Bus Timetable Detail

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB06 | | | |
| Use Case No. | MB06 | **Use Case Version** | 2.0 |
| Use Case Name | Reference Bus Timetable Detail | | |
| Author | NgoanTT | | |
| Date | 23/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer can show detail about timetable of a bus route on mobile application.   Goal:   * The system displays the timetable of bus route for reference.   Triggers:   * Customer sends a bus route timetable command.   Preconditions:   * System has a list of bus timetable detail. Using this function, customer can know when bus trips start and how many bus trips in one day so customer can choose their departure time appropriately.   Post Conditions:   * Success: System displays a detailed timetable of bus route. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to bus route detail view. | System displays bus route detail. | | 2 | Customer sends bus route timetable detail command. | System displays a detail timetable of bus route including:   * Start time of Trip. * End time of Trip.   [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | If no bus route timetable detail was founded. | System returns empty list. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from local database. * Bus route just supported in HCM city. | | | |

Table 35: Use case MB06 - <Customer> Reference Bus Timetable Detail

###### <Customer> Configuration Wear Setting



Figure 27: <Customer> Configuration Wear Setting

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB07 | | | |
| Use Case No. | MB07 | **Use Case Version** | 2.0 |
| Use Case Name | Configuration Wear Setting | | |
| Author | NgoanTT | | |
| Date | 24/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows customer can config accurate time for getting GPS from wear on mobile application.   Goal:   * The system sets new delay time for getting GPS on Wear. This function will help set GPS accurate on Wear. If user set refresh time small, GPS will be more accurately but battery will drain faster. Refresh time big, GPS will be less accurately but battery will drain slower.   Triggers:   * Customer sends a set time command.   Preconditions:   * N/A   Post Conditions:   * Success: System sets new time. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to config time view. | System displays config view including:   * Set time: number. | | 2 | Customer sends command to set new time. | System shows a successful message.  [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | If the time doesn’t change. | System shows a notify message that time doesn’t change. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * Set default time get GPS is 3 seconds. * Minimum time is 1 second and maximum time is 5 seconds. | | | |

Table 36: Use case MB07 - <Customer> Configuration Wear Setting

###### <Customer> Synchronize with Wear



Figure 28: <Customer> Synchronize with Wear

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – MB08 | | | |
| Use Case No. | MB08 | **Use Case Version** | 2.0 |
| Use Case Name | Synchronize with Wear | | |
| Author | NgoanTT | | |
| Date | 24/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows the customer to synchronize data between the mobile and wear device on mobile application.   Goal:   * Data has to synchronize between the mobile and wear. Using this function, wear application will always have newest data from mobile so wear application will display data more accurately.   Triggers:   * Customer sends a wear synchronize command.   Preconditions:   * Pair between the mobile and wear device with [Android Wear app](https://play.google.com/store/apps/details?id=com.google.android.wearable.app). * Search a route on mobile device and already choose one route.   Post Conditions:   * Success: Wear application has data from mobile application. * Fail: Show a warning message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | A customer sends commands synchronize data. | System processes and shows a notify message that synchronize data success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Android device and wear device haven’t paired together yet. | Show a message that wear device and mobile device haven’t paired yet. |   [Exception 2]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Android device and wear device are out of range. | Show a message that wear device and mobile device are out of range. |   Relationships: N/A  Business Rules:   * When wear device and mobile device paired, system will go to connected state. * When wear device and mobile device unpaired, system will go to unconnected state. * If wear device and mobile device are in connected state and system cannot connect those device, we move to out of range state. When system is on this state, system will show out or range message for customer. | | | |

Table 37: Use case MB08 - <Customer> Synchronize with Wear

###### <Customer> Synchronize Data with Server



Figure 29: <Customer> Synchronize Data with Server

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC9 | | | |
| Use Case No. | UC9 | **Use Case Version** | 2.0 |
| Use Case Name | Synchronize Data with Server | | |
| Author | NgoanTT | | |
| Date | 24/09/2015 | **Priority** | Normal |
| Actor:   * Customer   Summary:   * This use case allows the customer synchronize data with server from mobile application.   Goal:   * Data has to synchronize between a server and mobile. By this function, mobile will always keep latest data for supporting searching bus route information or bust timetable information when mobile doesn’t connect to internet.   Triggers:   * A customer sends the web synchronize command.   Preconditions:   * Mobile must be connected with internet.   Post Conditions:   * Success: Show synchronize data success message. * Fail: Show an error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | A customer sends commands synchronize data. | System processes and shows a notify message that synchronize data success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:  [Exception 1]   |  |  |  | | --- | --- | --- | | No. | Actor Action | 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:   * When mobile application recognizes that server has new data, mobile application will confirm user before download data. | | | |

Table 38: Use case MB09 - <Customer> Synchronize Data with Server

#### Android Wear Application

##### <Customer> Overview Use Case



Figure 30: <Customer> Overview use case

###### <Customer> Show information of a bus station on map



Figure 31: <Customer> Show information of a bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR01 | | | |
| Use Case No. | WR01 | **Use Case Version** | 2.0 |
| Use Case Name | Show information of a bus station on map | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer select bus station on map to see information of it and implement on wear device.   Goal:   * System shows detail information of bus station on map which customer has chosen. Using this function, customer can locate a bus station on map for easier navigate where they are in bus route.   Triggers:   * Customer sends a select bus station command.   Preconditions:   * Wear is paired with mobile. * Mobile has sent data to wear. ~~(Wear is synced with mobile)~~ * If wear device doesn’t have GPS function, mobile must be enabled GPS. * 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 it. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer drags on the screen. | Map moves follow customer’s action.  [Exception 1, 2, 3, 4] | | 2 | Customer selecst on one bus station on map. | Application shows information of bus station.  [Exception 1, 2, 3, 4] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | System shows message to notify that wear is not paired. | | 2 | Mobile hasn’t sent data to wear. | System shows message to notify that wear hasn’t received data yet. | | 3 | If wear doesn’t have GPS and mobile’s GPS does not enable. | System shows message to notify that mobile’s GPS does not enabled. | | 4 | If wear doesn’t have internet and mobile is not connected to internet | Show message to notify that mobile is not connected to internet. |   Relationships: N/A  Business Rules:   * System will use Google Map for rendering bus station. * 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 of each bus station includes: * Bus station name. * Order of this station in path. * Distance from this station to current location. | | | |

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

###### <Customer> Show information of a motorbike turn on map



Figure 32: <Customer> Show information of a bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR02 | | | |
| Use Case No. | WR02 | **Use Case Version** | 1.0 |
| Use Case Name | Show information of a motorbike turn on map | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer select motorbike turn on map to see notification of it and implement on wear device.   Goal:   * Show detail information of motorbike’s turn 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 motorbike turn command.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile. * If wear device doesn’t have GPS function, mobile must be enabled GPS. * 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 it. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Drag on the screen | Map moves follow customer’s action.  [Exception 1, 2, 3, 4] | | 2 | Select on one motorbike turn of trip. | Map moves to motorbike turn which customer has selected and shows information of it.  [Exception 1, 2, 3, 4] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. | | 3 | Mobile’s GPS does not enable | Show message to notify that mobile’s GPS does not enabled. | | 4 | Mobile is not connected to internet | Show message to notify that mobile is not connected to internet. |   Relationships: N/A  Business Rules:   * System uses Google Map for rendering bus station. * 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: * Motorbike turn name. * Order of this turn in path. * Distance from this turn to current location. | | | |

Table 40: Use case WR02 - <Customer> Show information of a bus station on map

###### <Customer> Navigate current place on map



Figure 33: <Customer> Navigate current place map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR03 | | | |
| Use Case No. | WR03 | **Use Case Version** | 1.0 |
| Use Case Name | Navigate current place on map | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer turn back to current place and implement on wear device.   Goal:   * Map moves to a current place to customer knows his/her location.   Triggers:   * Customer sends current place command.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile. * Mobile must enable GPS. * Mobile must be connected to internet.   Post Conditions:   * Success: Map move to current place. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to map view. | System will show map view.  [Exception 1, 2, 3, 4] | | 2 | Customer sends current place command. | Map moves to current place.  [Exception 1, 2, 3, 4] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. | | 3 | Mobile’s GPS does not enable | Show message to notify that mobile’s GPS does not enabled. | | 4 | Mobile is not connected to internet | Show message to notify that mobile is not connected to internet. |   Relationships: N/A  Business Rules:   * Map will be zoomed and current place will be showed on center of map. * System uses Google Map for rendering bus station. * If android wear has own GPS, application will use android wear’s GPS. Otherwise, android wear application will use GPS from mobile device. | | | |

Table 41: Use case WR03 - <Customer> Navigate current place map

###### <Customer> Notify at leaving bus station

  
Figure 34: <Customer> Navigate current place map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR04 | | | |
| Use Case No. | WR04 | **Use Case Version** | 1.0 |
| Use Case Name | Notify at leaving bus station | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows wear notifies when customer is in circular range of station.   Goal:   * Wear will show notification when customer drives nearly the leaving bus station of trip. Therefore, customer can prepare for leaving bus.   Triggers:   * Customer is near the leaving bus station.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile. * Mobile must has GPS – enable. * Mobile must be connected to internet.   Post Conditions:   * Success: Wear shows notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer is nearly the leaving bus station of trip. | Application will show message that customer should to leave.  [Exception 1, 2, 3, 4] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. | | 3 | Mobile’s GPS does not enable | Show message to notify that mobile’s GPS does not enabled. | | 4 | Mobile is not connected to internet | Show message to notify that mobile is not connected to internet. |   Relationships: N/A  Business Rules:   * Notify to customer when bus is in circular range of leaving bus station and notify again after 1 minute later with special sound and vibrate. After one minute, wear will stop vibrate. | | | |

Table 42: Use case WR04 - <Customer> Navigate current place map

###### <Customer> Notify at motorbike turn



Figure 35: <Customer> Notify at motorbike turn

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR05 | | | |
| Use Case No. | WR05 | **Use Case Version** | 1.0 |
| Use Case Name | Notify at motorbike turn | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows wear notifies when customer is in circular range of turn.   Goal:   * Wear will show notification when customer drives near the motorbike turn. Therefore, customer can prepare changing turn.   Triggers:   * Customer is near the motorbike turn.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile. * Mobile must has GPS – enable. * Mobile must be connected to internet.   Post Conditions:   * Success: Wear shows notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer is near the motorbike turn of trip. | Application will show message that customer should to turn and where to turn.  [Exception 1, 2, 3, 4] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. | | 3 | Mobile’s GPS does not enable | Show message to notify that mobile’s GPS does not enabled. | | 4 | Mobile is not connected to internet | Show message to notify that mobile is not connected to internet. |   Relationships: N/A  Business Rules:   * System will notify to customer when he/she is in circular range of motorbike turn and notify again after 1 minute later with special sound and vibrate. After one minute, wear device will stop vibrate. | | | |

Table 43: Use case WR05 - <Customer> Notify at motorbike turn

###### <Customer> Show list bus station



Figure 36: <Customer> Show list bus station

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR06 | | | |
| Use Case No. | WR06 | **Use Case Version** | 1.0 |
| Use Case Name | Show list bus station | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer see all bus station of trip.   Goal:   * System shows all bus station of trip on list. Therefore, customer can see all bus stations should to go.   Triggers:   * Customer selects and holds one second on a of bus station.   Preconditions:   * N/A   Post Conditions:   * Success: Wear shows notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select and hold one second on one of bus station | Show list of leaving bus station. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * All of leaving bus stations will be highlight on list. * Selected bus station will be center on screen. * Each bus station displays following information: * Bus route name. * Bus route street. * Distance from that bus station to current place. | | | |

Table 44: Use case WR06 - <Customer> Show list bus station

###### <Customer> Show list motorbike turn



Figure 37: <Customer> Show list motorbike turn

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR07 | | | |
| Use Case No. | WR07 | **Use Case Version** | 1.0 |
| Use Case Name | Show list motorbike turn | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer see all of motorbike turn.   Goal:   * System shows all of motorbike turn on list.   Triggers:   * Select and hold one second on one of motorbike turn.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile.   Post Conditions:   * Success: Wear show notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer selects and holds one second on one of motorbike turn | System shows list of motorbike turn. |   Alternative Scenario: N/A  Exceptions:  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. |   Relationships: N/A  Business Rules:   * Selected bus station will be center on screen. * Each bus station displays following information: * Bus route name. * Bus route street. * Distance from that bus station to current place. | | | |

Table 45: Use case WR07 - <Customer> Show list motorbike turn

###### <Customer> Show bus station on map



Figure 38: <Customer> Show bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR08 | | | |
| Use Case No. | WR08 | **Use Case Version** | 1.0 |
| Use Case Name | Show bus station on map | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer see detail of leaving bus station on map.   Goal:   * System shows leaving bus station and information of it on map. Therefore, customer can know more detail about bus station will go.   Triggers:   * Customer sends show bus station command.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile.   Post Conditions:   * Success: Wear show notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to bus station list view. | System will display list of stations. | | 2 | Customer sends show bus station command. | System shows bus station and information of it on map.  [Exception 1, 2] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. |   Relationships:   * N/A   Business Rules:   * Map will be zoomed and bus station will be on center of map. * Each message displays following information: * Bus route name. * Bus route street. * Distance from that bus station to current place. | | | |

Table 46: Use case WR08 - <Customer> Show bus station on map

###### <Customer> Show motorbike turn on map



Figure 39: <Customer> Show motorbike on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WR09 | | | |
| Use Case No. | WR09 | **Use Case Version** | 1.0 |
| Use Case Name | Show motorbike turn on map | | |
| Author | NamNT | | |
| Date | 27/05/2015 | **Priority** | Medium |
| Actor:   * Customer.   Summary:   * This use case allows customer see detail of motorbike turn on map.   Goal:   * System shows motorbike turn and detail information of it on map.   Triggers:   * Customer sends show motorbike turn command.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile.   Post Conditions:   * Success: Wear show notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to motorbike turn list | System will show motorbike turn list. | | 2 | Customer sends show motorbike turn command. | System shows motorbike turn and information of it on map.  [Exception 1, 2] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Wear is not paired | Show message to notify that wear is not paired. | | 2 | Wear is not synced | Show message to notify that wear is not synced. |   Relationships: N/A  Business Rules:   * Motorbike turn will be on center of map. * Each message displays following information: * Motorbike turn name. * Motorbike turn street. * Distance from that turn to current place | | | |

Table 47: Use case WR09 - <Customer> Show motorbike on map

### Bus Driver Mobile Application

##### <Bus Driver> Overview Use Case

## 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



Figure 40: Conceptual diagram

**Data Dictionary**

|  |  |
| --- | --- |
| Entity Data dictionary: describe all content of all entities | |
| Entity Name | **Description** |
| Route | Represents a sequence of stations of bus |
| Station | Represent distinct location in the network where one can board or get off a vehicle (bus, train) |
| PathInfo | Represent an information between two consecutive stations in one bus route. |
| MiddlePoint | Represent a point on map that bus must go through between two consecutive stations. |
| Trip | Represent a journey of one route with start time from first station and end time of last station. |
| Connection | Represent a path info with time arrival and time departure of one trip. |
| TicketType | Represent ticket type model. |
| TicketPrice | Represent ticket price for each ticket type of each route. |

*Table 48: Conceptual Diagram Data Dictionary*

# 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)