# 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 5](#_Toc430941747)

[Table 2: Roles and Responsibilities 9](#_Toc430941748)

[Table 3: Hardware requirement for continuous integrating server 12](#_Toc430941749)

[Table 4: Hardware requirement for web development 13](#_Toc430941750)

[Table 5: Hardware requirement for mobile development 13](#_Toc430941751)

[Table 6: Hardware requirement for wear development 13](#_Toc430941752)

[Table 7: Software requirement 13](#_Toc430941753)

[Table 8: Roles and responsibilities 15](#_Toc430941754)

[Table 9: Tools and Techniques 16](#_Toc430941755)

[Table 10:Software development life cycle 18](#_Toc430941756)

[Table 11: Requirements definition 19](#_Toc430941757)

[Table 12: System and software design 19](#_Toc430941758)

[Table 13: Implementation and unit test 20](#_Toc430941759)

[Table 14: Integration and system testing 20](#_Toc430941760)

[Table 15: Operation and maintenance 20](#_Toc430941761)

[Table 16:Naming conventions for drawables 21](#_Toc430941762)

[Table 17: Naming conventions for icons 22](#_Toc430941763)

[Table 18: Use case UC01 - <Customer> Search Arbitrary Location on map 28](#_Toc430941764)

[Table 19: Use case UC02 - <Customer> Search Bus Route 31](#_Toc430941765)

[Table 20: Use case UC03 - <Customer> Navigate Current Location On Map 33](#_Toc430941766)

[Table 21: Use case UC04 - <Customer> Search Motorbike Route 35](#_Toc430941767)

[Table 22: Use case UC05 - <Customer> Reference Bus Route Detail 36](#_Toc430941768)

[Table 23: Use case UC06 - <Customer> Reference Bus Timetable Detail 38](#_Toc430941769)

[Table 24: Use case UC07 - <Customer> Configuration Wear Setting 40](#_Toc430941770)

[Table 25: Use case UC08 - <Customer> Synchronize with Wear 41](#_Toc430941771)

[Table 26: Use case UC09 - <Customer> Synchronize Data with Server 43](#_Toc430941772)

[Table 27: Use case UC10 - <Customer> Show information of a bus station on map 45](#_Toc430941773)

[Table 28: Use case UC11 - <Customer> Show information of a bus station on map 46](#_Toc430941774)

[Table 29: Use case UC12 - <Customer> Navigate current place map 48](#_Toc430941775)

[Table 30: Use case UC13 - <Customer> Navigate current place map 49](#_Toc430941776)

[Table 31: Use case UC14 - <Customer> Notify at motorbike turn 51](#_Toc430941777)

[Table 32: Use case UC15 - <Customer> Show list bus station 52](#_Toc430941778)

[Table 33: Use case UC16 - <Customer> Show list motorbike turn 54](#_Toc430941779)

[Table 34: Use case UC17 - <Customer> Show bus station on map 55](#_Toc430941780)

[Table 35: Use case UC18 - <Customer> Show motorbike on map 57](#_Toc430941781)

[Table 36: Conceptual Diagram Data Dictionary 59](#_Toc430941782)

# List of Figures

[Figure 1: Waterfall model 14](#_Toc430941827)

[Figure 2: <Guest> Search Location 26](#_Toc430941828)

[Figure 3: <Customer> Search Arbitrary Location on map 27](#_Toc430941829)

[Figure 4: <Customer> Search Bus Route 29](#_Toc430941830)

[Figure 5: <Customer> Navigate Current Location On Map 31](#_Toc430941831)

[Figure 6: <Customer> Search Motorbike Route 33](#_Toc430941832)

[Figure 7: <Customer> Reference Bus Route Detail 35](#_Toc430941833)

[Figure 8: <Customer> Reference Bus Timetable Detail 37](#_Toc430941834)

[Figure 9: <Customer> Configuration Wear Setting 38](#_Toc430941835)

[Figure 10: <Customer> Synchronize with Wear 40](#_Toc430941836)

[Figure 11: <Customer> Synchronize Data with Server 41](#_Toc430941837)

[Figure 12: <Customer> Overview use case 43](#_Toc430941838)

[Figure 13: <Customer> Show information of a bus station on map 44](#_Toc430941839)

[Figure 14: <Customer> Show information of a bus station on map 45](#_Toc430941840)

[Figure 15: <Customer> Navigate current place map 47](#_Toc430941841)

[Figure 16: <Customer> Navigate current place map 48](#_Toc430941842)

[Figure 17: <Customer> Notify at motorbike turn 50](#_Toc430941843)

[Figure 18: <Customer> Show list bus station 51](#_Toc430941844)

[Figure 19: <Customer> Show list motorbike turn 53](#_Toc430941845)

[Figure 20: <Customer> Show bus station on map 54](#_Toc430941846)

[Figure 21: <Customer> Show motorbike on map 56](#_Toc430941847)

[Figure 22: Conceptual diagram 58](#_Toc430941848)

# 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



#### Android Mobile Application



#### Android Wear Application



#### Bus Driver Mobile Application

### List of Use Case

#### Web Application

##### <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 implement on web application.   Goal:   * Guest can log in the system.   Triggers:   * Guest sends the login command.   Preconditions:   * N/A   Post Conditions:   * Success: Guest login the 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 login successful, guest will login system with login account role.  [Alternative 1] |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Guest fills incorrect information. | * If login fail, system show error message “Invalid username or password”. |   Exceptions: N/A  Relationships: N/A  Business Rules:   * Password field must be displayed “\*”. * Encrypt password before being send to server. * After login to system successful, guest will be redirected to specific 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 10: Use case WG01 - <Guest> Login

##### <Staff> Overview Use Case



###### <Staff> Logout



Figure 21 <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:   * Staff can logout the system.   Triggers:   * Staff sends logout command.   Preconditions:   * N/A   Post Conditions:   * Success: Staff logs out the system. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff sends logout command. | Staff exists from system.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Session timeout. | System log out automatically then navigate to login page because of long inactivity. |   Relationships: N/A  Business Rules:   * If user is inactive in 30 minutes, system will automatically log user out. | | | |

Table 24 Use case WS01 - <Staff> Logout

###### 1.1.1.2.6. <Staff> Approve or reject all current system notification



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

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS06 | **Use Case Version** | 2.0 |
| Use Case Name | Approve or reject all current system notification | | |
| 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 provide approve or reject all current system notifications for supporting staff manage system notifications.   Triggers:   * Staff sends approve or reject all current system notifications command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: All current system notifications will be approved or rejected. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to all system notifications view. | 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. * Approve all actions. * Reject all actions. | | 2 | Staff send approve command. [Alternative 1,2] | The system will approve or reject all system notifications. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * If staff approve all notifications, system will implement all changes from requires of notifications. (state …) * If staff reject all notifications, system will not implement all changes from requires of notifications. | | | |

Table 28 Use case WS04 - <Staff> Resolve compensation request

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



Figure 25 <Staff> Approve or reject detail notification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS07 | **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 provide 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 be 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 send approve or reject command. | The system will approve or reject notification. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * If staff approve detail notification, system will implement change from requires of notification. * If staff reject detail notification, system will not implement change from requires of notification. | | | |

Table 28 Use case WS04 - <Staff> Approve or reject detail notification.

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

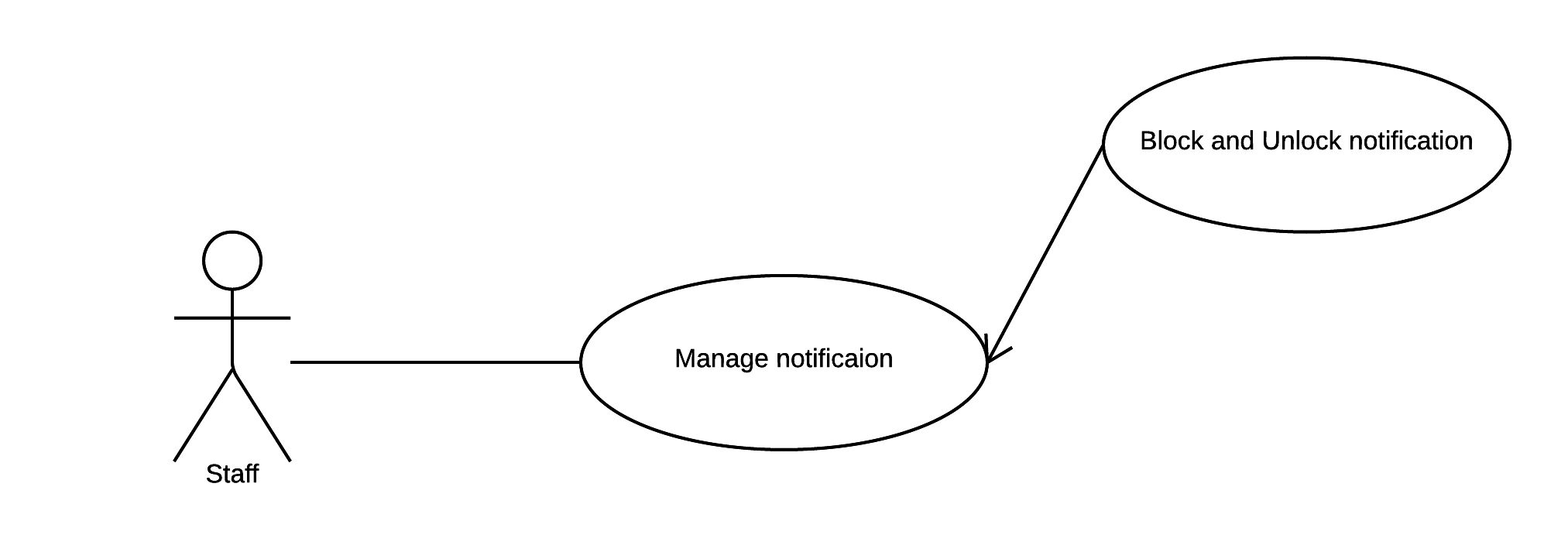


Figure 25 <Staff> Unblock notification

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS08 | **Use Case Version** | 2.0 |
| Use Case Name | Block and unblock notification. | | |
| Author | DatNT | | |
| Date | 23/09/2015 | **Priority** | medium |
| Actor:   * Staff.   Summary:   * This use case allows staff to unblock notification and this function will implement on web application.   Goal:   * System provide block and unlock notification function for supporting staff to manage notifications.   Triggers:   * Staff sends view list of block notifications 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 | | 1 | Staff send block and unlock notification command. | The system block and unlock requested notification. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * If staff block notification, the system will not notify staff about this notification. * If staff unblock notification, the system will notify staff about this notification. | | | |

Table 28 Use case WS04 - <Staff> Unlock notification.

###### 1.1.1.2.10. <Staff> Configure parsing source



Figure 25 <Staff> Configure parsing source

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS08 | **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 choose parsing source command.   Preconditions:   * Staff must login into the system with role Staff.   Post Conditions:   * Success: Show successful message. * Fail: N/A.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff goes to configure parsing source view. | System have two option parsing:   * Parsing from file: staff chooses file from local computer. * Parsing from web: system automatic parses from specified web. | | 2 | Staff sends configure parsing source command. | System will notify success. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Currently, system just supports Ho Chi Minh bus system website. * Currently, system just supports excel files provided by Ho Chi Minh bus system website. | | | |

Table 28 Use case WS04 - <Staff> Configure parsing source.

###### 1.1.1.2.11. <Staff> Configure parsing time



Figure 25 <Staff> Configure parsing time

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS04 | | | |
| Use Case No. | WS08 | **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 provide 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: Show successful message. * 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. | | 2 | Staff send configure parsing time command. | System will show success message.  If configuration does not change then system shows message that for staff know that no configuration change. |   Alternative Scenario: N/A.  Exceptions: N/A.  Relationships: N/A.  Business Rules:   * Time must be from 11:00 PM to 11:59 PM, 0:00 AM to 04:00 AM. | | | |

Table 28 Use case WS04 - <Staff> Choose parsing time.

#### Android Mobile Application

##### <Customer> Overview Use Case



Figure 2: <Guest> Search Location

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



Figure 3: <Customer> Search Arbitrary Location on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC01 | | | |
| Use Case No. | UC01 | **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 arbitrary a location on map on mobile application.   Goal:   * The system shows location on map and gives detail information about location.   Triggers:   * Customer sends location command.   Preconditions:   * Mobile must be connected to internet.   Post Conditions:   * Success: Show location on map and detail information about it. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to search view. | System display search view include:   * Location search: text input * Show list history | | 2 | Customer input arbitrary location.  [Alternative 1] | System suggestion location while typing. | | 3 | Customer sends command search | - System process and show location on map.  - If system cannot find location on map then system will show message notify to customer that location not found on map. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends location from history. | System process and show location on map. |   Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile doesn’t have internet | Show error message that this device haven’t connected to internet yet. |   Relationships: N/A  Business Rules:   * The information is loaded from Google map. * Search location will be stored in history database. * History will be limit in 20 results. | | | |

Table 18: Use case UC01 - <Customer> Search Arbitrary Location on map

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



Figure 4: <Customer> Search Bus Route

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC02 | | | |
| Use Case No. | UC02 | **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 thought some locations (from two to four) on mobile application.   Goal:   * The system will suggest optimal paths for customer.   Triggers:   * Customer sends search command.   Preconditions:   * Mobile must be connected to internet. * GPS enable.   Post Conditions:   * Success: System draw route from start location to end location and show on map. * Fail: Show error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to search route view. | Display search route view including:   * Start location : text * End location : text * Send search command: button | | 2 | Customer goes to start and end location view. | Display new view with following information:   * Location : input text * Location with GPS * Location from history | | 3 | Customer input start and end location.  [Alternative 1]  [Alternative 2] | Input field will show location that customer has selected. | | 4 | Customer goes to choose departure time view. | System will show date time picker for customer selecting time. | | 5 | Customer can optional choose departure time. | System will update date on date time picker. | | 6 | Customer goes to choose middle locations view. | Display new view with following information:   * First middle location : text * Second middle location : text | | 7 | Customer can input two middle locations.  [Alternative 1]  [Alternative 2] | Input field will show location that customer has selected. | | 8 | Customer can choose optimal option. | System will check the checkbox. | | 9 | Customer sends command search | System show list with some optimal paths.  [Exception 2] | | 10 | Customer chooses one path command. | System will drag path on map and show detail about it. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends location from history. | System process and show location that customer has selected. | | 2 | Customer chooses location with GPS | System process and show location that customer has selected. |   Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile doesn’t have internet | Show error message that this device haven’t connected to internet yet. | | 2 | If start location or end location text field not input. | System show missing field message to customer. |   Relationships: Has an extension optimize search for bus route.  Business Rules:   * Using Google map for displaying map. * Location must be in HCM city. * The algorithm for finding bus route is called from web service. | | | |

Table 19: Use case UC02 - <Customer> Search Bus Route

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



Figure 5: <Customer> Navigate Current Location On Map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC03 | | | |
| Use Case No. | UC03 | **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 map with GPS on mobile application.   Goal:   * The system shows current location on map and give detail information about current location.   Triggers:   * Customer chooses send GPS command.   Preconditions:   * GPS enable. * Mobile must be connected to internet.   Post Conditions:   * Success: Show location on map and detail information about it. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends current location command. | System will display that location on map, show information or system show message notify to customer cannot found location. |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile doesn’t have internet | Show error message that this device haven’t connected to internet yet. |   Relationships: N/A  Business Rules:   * Using Google map for displaying location. | | | |

Table 20: Use case UC03 - <Customer> Navigate Current Location On Map

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



Figure 6: <Customer> Search Motorbike Route

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC04 | | | |
| Use Case No. | UC04 | **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 mobile application.   Goal:   * The system will suggest an optimal path for customer.   Triggers:   * Customer sends motorbike search route command.   Preconditions:   * Mobile must be connected to internet.   Post Conditions:   * Success: System draw route from start location to end location and show on map. * Fail: Show error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to search route view. | Display search route view including:   * Start location : text * End location : text * Send search command: button | | 2 | Customer goes to start and end location view. | Display new view with following information:   * Location : input text * Location with GPS * Location from history | | 3 | Customer input start and end location.  [Alternative 1]  [Alternative 2] | Input field will show location that customer has selected. | | 4 | Customer goes to choose departure time view. | System will show date time picker for customer selecting time. | | 5 | Customer can optional choose departure time. | System will update date on date time picker. | | 6 | Customer goes to choose middle locations view. | Display new view with following information:   * First middle location : text * Second middle location : text | | 7 | Customer can input two middle locations.  [Alternative 1]  [Alternative 2] | Input field will show location that customer has selected. | | 8 | Customer can choose optimal option. | System will check the checkbox. | | 9 | Customer sends command search | System show list with some optimal paths.  [Exception 2] | | 10 | Customer chooses one path command. | System will drag path on map and show detail about it. |   Alternative Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends location from history. | System process and show location that customer has selected. | | 2 | Customer chooses location with GPS | System process and show location that customer has selected. |   Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Mobile doesn’t have internet | Show error message that this device haven’t connected to internet yet. | | 2 | If start location or end location text field not input. | System show missing field message to customer. |   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. | | | |

Table 21: Use case UC04 - <Customer> Search Motorbike Route

###### <Customer> Reference Bus Route Detail



Figure 7: <Customer> Reference Bus Route Detail

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC05 | | | |
| Use Case No. | UC05 | **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 show detail all bus route on mobile application.   Goal:   * The system displays list bus route detail for reference.   Triggers:   * Customer sends reference bus route detail command.   Preconditions:  Post Conditions:   * Success: System display list bus route detail on screen. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to bus route list view. | System will display list of bus route. | | 2 | Customer sends reference bus route command with bus route number parameter. | System will display 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.   System returns empty list if no bus route found. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: Has an extension “Reference Bus Timetable Detail”  Business Rules:   * The information is loaded from local database. * Bus route just supported in HCM city. | | | |

Table 22: Use case UC05 - <Customer> Reference Bus Route Detail

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



Figure 8: <Customer> Reference Bus Timetable Detail

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC06 | | | |
| Use Case No. | UC06 | **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 will display timetable of bus route for reference.   Triggers:   * Customer sends bus route detail command.   Preconditions:   * System have list of bus route detail.   Post Conditions:   * Success: System display detail timetable of bus route. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to bus route detail view. | System will display bus route detail. | | 1 | Customer sends bus route detail command. | System will display detail timetable of bus route include:   * Start time of Trip. * End time of Trip.   System returns empty list if no bus route found. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * The information is loaded from local database. * Bus route just supported in HCM city. | | | |

Table 23: Use case UC06 - <Customer> Reference Bus Timetable Detail

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



Figure 9: <Customer> Configuration Wear Setting

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC07 | | | |
| Use Case No. | UC07 | **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 set new time.   Triggers:   * Customer sends command set time.   Preconditions:  Post Conditions:   * Success: System set new time. * Fail: N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer goes to config time view. | System display config view including:   * Set time: number | | 2 | Customer sends command to set new time. | System show successfully message  If the time doesn’t change then system show message that nothing change. |   Alternative Scenario: N/A  Exceptions: N/A  Relationships: N/A  Business Rules:   * Set default time get GPS is 3 seconds. * Maximum time is 5 seconds. | | | |

Table 24: Use case UC07 - <Customer> Configuration Wear Setting

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



Figure 10: <Customer> Synchronize with Wear

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UC08 | | | |
| Use Case No. | UC08 | **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 customer can synchronize data between mobile and wear device on mobile application.   Goal:   * Data have to synchronize between mobile and wear.   Triggers:   * Customer sends synchronize command.   Preconditions:   * Pair between mobile and wear device with [Android Wear app](https://play.google.com/store/apps/details?id=com.google.android.wearable.app). * Search route on mobile device and choice one route.   Post Conditions:   * Success: Wear application has data from mobile application. * Fail: Show message warning.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends commands synchronize data. | - System process and show message notify to customer that synchronize successful.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends commands synchronize data. | Show message notify to customer that synchronize fail. |   Relationships: N/A  Business Rules: N/A | | | |

Table 25: Use case UC08 - <Customer> Synchronize with Wear

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



Figure 11: <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 allow customer synchronize data with server from on mobile application.   Goal:   * Data have to synchronize between server and mobile.   Triggers:   * Customer sends synchronize command.   Preconditions:  Post Conditions:   * Success: Show synchronize data success message. * Fail: Show error message   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer send commands synchronize data. | System process and show message to notify customer that synchronize data success.  [Exception 1] |   Alternative Scenario: N/A  Exceptions:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer sends command synchronize data | Show message to notify customer that synchronize fail. |   Relationships: N/A  Business Rules: N/A | | | |

Table 26: Use case UC09 - <Customer> Synchronize Data with Server

#### Android Wear Application

##### <Customer> Overview Use Case



Figure 12: <Customer> Overview use case

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



Figure 13: <Customer> Show information of a bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS10 | **Use Case Version** | 1.0 |
| 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:   * Show detail information of bus station on map which customer choose. 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. * Wear is synced with mobile. * Mobile must enable GPS. * Mobile must be connected to internet.   Post Conditions:   * Success: Map move to new place and show 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 bus station on map. | Application show information of bus station.  [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:   * Using 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 including: * Bus station name. * Order of this station in path. * Distance from this station to current location. | | | |

Table 27: Use case UC10 - <Customer> Show information of a bus station on map

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



Figure 14: <Customer> Show information of a bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS11 | **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 turn on map which customer choose. 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. * Mobile must enable GPS. * Mobile must be connected to internet.   Post Conditions:   * Success: Map move to new place and show 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 move to motorbike turn which customer Select on and show 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:   * Using 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 including: * Motorbike turn name. * Order of this turn in path. * Distance from this turn to current location. | | | |

Table 28: Use case UC11 - <Customer> Show information of a bus station on map

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



Figure 15: <Customer> Navigate current place map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS12 | **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 move to a current place to customer knows his/her location.   Triggers:   * Customer send 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 | Send current place command. | Map move 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 zoom and current place will be show on center of map. * Using 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 29: Use case UC12 - <Customer> Navigate current place map

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

  
Figure 16: <Customer> Navigate current place map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS13 | **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 show 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 30: Use case UC13 - <Customer> Navigate current place map

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



Figure 17: <Customer> Notify at motorbike turn

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS14 | **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 show 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:   * 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 31: Use case UC14 - <Customer> Notify at motorbike turn

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



Figure 18: <Customer> Show list bus station

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS15 | **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:   * Show all bus station of trip on list. Therefore, customer can see all bus stations should to go.   Triggers:   * Select and hold one second on one of bus station.   Preconditions:   * Wear is paired with mobile. * Wear is synced with mobile. * Mobile must enable GPS. * Mobile must be connected to internet.   Post Conditions:   * Success: Wear show 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 32: Use case UC15 - <Customer> Show list bus station

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



Figure 19: <Customer> Show list motorbike turn

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS16 | **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:   * Show 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 | Select and hold one second on one of motorbike turn | Show 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 33: Use case UC16 - <Customer> Show list motorbike turn

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



Figure 20: <Customer> Show bus station on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS17 | **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:   * Show leaving bus station and information of it on map. Therefore customer can know more detail about bus station will go.   Triggers:   * Send show bus station 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: 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. | Show 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 zoom 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 34: Use case UC17 - <Customer> Show bus station on map

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



Figure 21: <Customer> Show motorbike on map

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – WS10 | | | |
| Use Case No. | WS18 | **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:   * Show motorbike turn and detail information of it on map.   Triggers:   * Send show motorbike turn 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: Wear show notification to customer. * Fail: Show error message.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Go to motorbike turn list | System will show motorbike turn list. | | 2 | Send show motorbike turn command. | Show 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 35: Use case UC18 - <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 22: 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 36: 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)