# 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 loose 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.

### 1.3. Project Overview

#### **1.3.1. 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 database on mobile device and synchronize with server database for avoiding some look up when no network around.

Moreover, we have built a background handler for checking third-party server periodically every 0 AM 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.

* + - * 1. **Background Handler**
* Check new data periodically at 0 AM each day.
* Parsing data from website

##### **Bus Routing Mobile Application**

This application is used by user and do 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).

##### **1.3.2.1.5 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 those 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 of this system is Vietnamese.
* Mobile devices run android 4.4 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.

#### **1.3.4. 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 and high-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**

##### **1.3.5.1. 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 2MB  Bluetooth Connection 4.0 | Wi-Fi Connection 12MB  Bluetooth Connection 4.0 |
| Operating System | Android 4.4 | 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

##### **1.3.5.2. 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 |
| 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 Trello |
| Others | Microsoft Word, Microsoft Excel, Adobe Photoshop |

Table 1 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 7 Roles and responsibilities

### 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, Lucid Chart |
| Document tool | Microsoft Word 2013, Microsoft Excel 2013 |

Table 8 Tools and Techniques

## Project Management Plan

### 3.1. 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 9 Software development life cycle

### 3.2 Phase Detail

#### **3.2.1 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  Nguyễn Trung Nam  Trần Thanh Ngoan  Ngô Tiến Đạt |

#### **3.2.2 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  Nguyễn Trung Nam  Trần Thanh Ngoan  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  Nguyễn Trung Nam  Trần Thanh Ngoan  Ngô Tiến Đạt |
| Design database. | - Design database for the system. | Huỳnh Quang Thảo  Nguyễn Trung Nam  Trần Thanh Ngoan  Ngô Tiến Đạt |

#### **3.2.3 Phase 3: Implementation and Unit Testing**

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

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

#### **3.2.4 Phase 4: Integration and System Testing**

#### **3.2.5 Phase 5: Operation and Maintenance**

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

### 3.3 Task sheet

Refer to “Task sheet” folder.

### 3.4 All Meeting Minutes:

Refer to “Meeting Minutes” folder.

## Coding Convention

### 4.1 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.

### 4.2 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 |

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 |

1. # 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. [↑](#endnote-ref-1)