# Table of contents

[Table of contents 1](#_Toc430522705)

[List of Table 2](#_Toc430522706)

[List of Figures 2](#_Toc430522707)

[Definitions, Acronyms, and Abbreviations 3](#_Toc430522708)

[A. Introduction 4](#_Toc430522709)

[1. Project Information 4](#_Toc430522710)

[2. Introduction 4](#_Toc430522711)

[3. Current Situation 4](#_Toc430522712)

[4. Problem Definition 5](#_Toc430522713)

[5. Proposed Solution 5](#_Toc430522714)

[5.1. Feature functions 5](#_Toc430522715)

[5.2. Advantages and disadvantages 6](#_Toc430522716)

[6. Functional Requirements 6](#_Toc430522717)

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

[B. Software Project Management Plan 8](#_Toc430522719)

[1. Problem Definition 8](#_Toc430522720)

[1.1. Name of this Capstone Project 8](#_Toc430522721)

[1.2. Problem Abstract 8](#_Toc430522722)

[1.3. Project Overview 8](#_Toc430522723)

[2. Project organization 12](#_Toc430522724)

[2.1. Software Process Model 12](#_Toc430522725)

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

[3. Project Management Plan 14](#_Toc430522727)

[3.1. Software development life cycle 14](#_Toc430522728)

[3.2. Phase Detail 17](#_Toc430522729)

[3.3. Task sheet 18](#_Toc430522730)

[3.4. All Meeting Minutes 18](#_Toc430522731)

[4. Coding Convention 18](#_Toc430522732)

[4.1. Java Coding Convention 18](#_Toc430522733)

[4.2. Android Coding Convention 19](#_Toc430522734)

[G. Appendix 20](#_Toc430522735)

# List of Table

[Table 1: Definitions, Acronyms, and Abbreviations 2](#_Toc430510047)

[Table 2: Roles and Responsibilities 7](#_Toc430510048)

[Table 3: Hardware requirement for continuous integrating server 9](#_Toc430510049)

[Table 4: Hardware requirement for web development 10](#_Toc430510050)

[Table 5: Hardware requirement for mobile development 10](#_Toc430510051)

[Table 6: Hardware requirement for wear development 10](#_Toc430510052)

[Table 7: Software requirement 10](#_Toc430510053)

[Table 8: Roles and responsibilities 12](#_Toc430510054)

[Table 9: Tools and Techniques 12](#_Toc430510055)

[Table 10: Software development life cycle 15](#_Toc430510056)

[Table 11: Requirements definition 16](#_Toc430510057)

[Table 12: System and software design 16](#_Toc430510058)

[Table 13: Implementation and unit test 17](#_Toc430510059)

[Table 14: Integration and system testing 17](#_Toc430510060)

[Table 15: Operation and maintenance 18](#_Toc430510061)

[Table 16:Naming conventions for drawables 19](#_Toc430510062)

[Table 17: Naming conventions for icons 20](#_Toc430510063)

# List of Figures

[Figure 1: Waterfall model 10](#_Toc430509953)

# 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) |
| Trip | 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. |
| Route | Route is a trip 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 : 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:ifofficial 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:App 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 in two minutes 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 | TrầnThanh 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 : Roles and Responsibilities

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

Official name: Smart Wear on Your Route

Vietnamese name: Hỗtrợđiđườngvớithiếtbịđeotaythông 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-2) 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 : 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 : 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 : 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 : 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 : 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 : 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 : 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 :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 : 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 : 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 : 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 : 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 : 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 :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 : Naming conventions for icons

# 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-2)