YamuColombo

**Colombo Bus Route Management System
Software Requirements Specification
For Mobile Devices**

Version 1.0

Revision History

Date	Version	Description	Author
<09/04/2016>	1.0	Software Requirement Specification	Lakshan Gamage

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to elaborate and describe the fully functional and non-functional requirements of the YamuColombo – Bus Route Management System as well as the external behavior of the system. This will describe the functionalities of the system, system design constraints and detailed description of the system requirements.

1.2 Scope

This is the SRS for an android base application addressing a common problem in day to day life of most of the people who travel in the city of Colombo by bus. The problem is finding the best buses to be chosen to reach a particular destination at a particular time. What this application basically provides is an optimum route for the traveler to reach his destination based on the time, distance, cost and traffic at a particular moment. Therefore it will be useful for those who don't know which bus to be taken as well as for those who can't decide which one of all the possible routes should be taken to reach the destination in time.

1.3 Definitions, Acronyms, and Abbreviations

SRS – Software Requirement Specification
RUP – Rational Unified Process
GUI – Graphical User Interface
GPS – Global Positioning System
API – Application Program Interface

1.4 References

[1] "Google Places API," Google, [Online]. Available: https://developers.google.com/places/android-api/. [Accessed 06 March 2016].

[2] "Google Maps Android API," Google, [Online]. Available: https://developers.google.com/maps/android/. [Accessed 06 March 2016].

[3] "Colombo Bus Route," Google Play, [Online]. Available: https://play.google.com/store/apps/details?id=cbr.arima&hl=en. [Accessed 06 March 2016].

[4] "Algorithms to find routes," StackOverflow, [Online]. Available: http://stackoverflow.com/questions/483488/strategy-to-find-your-best-route-via-public-transportation-only. [Accessed 03 03 2016].

[5] "Android Studio," Google, [Online]. Available: http://developer.android.com/sdk/index.html. [Accessed 03 03 2016].

[6] "Google APIs Terms of Service," Google, [Online]. Available: https://developers.google.com/terms/. [Accessed 16 03 2016].

1.5 Overview

Here on this document will describe several aspects if the system. Section 2 will provide a comprehensive description of the system and the third section will describe in detail the requirements of the system.

2. Overall Description

2.1 Product Perspective

This application allows users to find the best bus route by 3 options. He/she can give the starting location and the destination from a list, Take the current location of the user as the starting point using phone's GPS and give the destination only, mark both the starting point and destination on a map and find the route. The results can also be filtered based on cost, distance and traffic on that particular route at the moment.

2.2 Product Functions

- Users will be able to select the starting point and the destination of the journey and search for buses.
- Users will be able to mark only the destination on the map and the starting point will be taken by the application automatically though GPS.
- Users will be able to find out what are the buses that travel near to their current location.
- Once a user searches for a route in any of the methods, he/she will be given all the possible routes which can be used to reach the destination.
- Users will be able to decide a route upon the time, cost, distance and traffic at the moment.
- Once a route is selected, he/she will be shown the map with the current progress of their journey.
- At the end of the journey, the duration the journey and time of the day will be confirmed and the user will be asked for any comments about the selected route (Traffic Conditions, etc.).
- These time details will be used when another user searches for the same route to find optimum route. Route suggestions will be calculated considering the durations of the previous users and time of the day which they travelled. Then that person will receive more reliable information because it is calculated from previous user experience. Algorithms will be optimized to avoid false information from the users.
- Once a user select a route, they can view the comments of the previous users on that route and get an idea about the feasibility of the route.

2.3 User Characteristics

 This application will be designed to provide a good user experience for anyone. But the user should have a basic knowledge of using an android phone and a typical android application.

2.4 Constraints

As this application calculates optimum routes in real time, the algorithm for that should be much efficient because users won't want to wait long to get the results. Therefore this should have a considerably high efficiency.

There will be a backend server which processes all the functionalities of the system. This should be available all the time because this application relies very much upon that.

The server should have a considerable amount of performance because it should be able to handle all the user requests simultaneously and calculate the routes for all of them efficiently.

As Google map API v2 is supported only by phones which has android version 2.2 or higher, this application will run on any android phone which is higher than android version 2.2. Therefore it covers a huge percentage of the total android phones available in the market.

2.5 Assumptions and Dependencies

As this application depend very much on Google APIs, there's a risk which those functionalities will be changed or the information provided becomes unreliable. A small change in those APIs and their services can affect this application very much.

If some users give false feedbacks on a particular route, that will considerably affect the reliability of the application because other users may see those feedback and take

decisions based on them.

The user should have a sufficiently speed internet connection to communicate with the server and google maps.

The information provided should be up to date. If there's a change in a bus route, it should be updated in our system also. Otherwise the users will be provided with false information and the application will be rejected by the users.

As we are using Google APIs in this application, we have to adhere to their Terms of Service.

It is assumed that the user has the basic knowledge on using an android phone and a typical android application.

It is assumed most of the feedback of the users are true. Those that deviate a lot from the rest will be ignored.

3. Specific Requirements

3.1 Functionality

3.1.1 Log In to the System

In order to use the application it is required that the user is logged into the system. He/she can log into the system using the username and password given at the first use of the application. Providing username and password may not always needed as there will be an option to automatically login after they log into the system for the first time.

3.1.2 Get Starting Point and Destination

Once the user logs into the system, he/she will be able to select the starting point and the destination of the journey.

3.1.2.1 Get current location of the User

User's current location can be taken as the starting point of the journey and therefore GPS of the mobile phone will be used to automatically find out the current location. Then the nearest bus halt for that location will be taken and displayed on the map as the starting point.

3.1.2.2 Mark points on the map

Without going to the option of considering the current user location as the starting point, they can also mark both the starting point and the destination on the map itself. Then the nearest bus stops to the marked points will be taken for the route calculation.

3.1.2.3 Choose bus halts from a list

Without using the options in 3.1.2.1 and 3.1.2.2 one can also choose the starting point and the destination of the journey from a predefined list of halts.

3.1.3 Calculate the possible bus routes

Once the starting point and the destination of the journey is being taken, the calculation of which are the routes to be taken will be calculated based on the current scenario. Algorithms will be optimized to give better performance and increase efficiency.

3.1.4 Display possible bus routes

When the calculations are done, the list of possible bus routes will be displayed to the user. User may choose which one they should use. However an optimum route will be displayed based in the cost, distance and traffic at the moment.

3.1.5 Filter and sort the possible routes

When all the possible routes are displayed, user can filter and sort the routes in an order based on the cost, traffic and distance.

3.1.6 Comment on the route and the experience

After a user completes his/her journey, they can comment about the route and the experience they got on that route. These comments will be important for the other users who plans to choose the same route thereafter.

3.1.7 View Past user experience

When the user receives a list of possible routes, they can view the comments of other users on each of these routes and get an idea before deciding which one to go with. These will be about whether the bus is clogged by a lot of passengers, bus waits in some halts for a long time, the total time it took for the journey, conditions of the busses, speed of the busses etc.

3.1.8 Navigate throughout the journey

Once the user selects a particular route, he/she will be able to navigate on the google maps throughout the journey. Then they can view the progress, when and where to get off, and when and where to take the next bus if there is one.

3.2   Usability

3.2.1   Simple GUIs

The user interfaces will be designed in a very simple way which won't become complicated for the user to make use of the functionalities of the system. Complicated GUIs will make the user uncomfortable and will make them to take a long time to get familiar with.

3.2.2   Attractive GUIs

Even though the GUIs are simple, they will be designed in an attractive manner. Otherwise the app will be boring for most of the users.

3.2.3 Eye catching color schemes

Color schemes of the GUIs will be selected so as not to make the user uncomfortable.

3.2.4 Sufficiently large buttons and text

Text and buttons on the GUIs will be large enough for any user to see and touch easily.

3.2.5 User Guide

There will be a simple user guide in any case the user still finds it hard to understand to perform a certain functionality.

3.3 Reliability

3.3.1   Availability

As this application depends very much on the backend server, it has to be on and running all the time. Whenever a user needs to use the application, the server should be readily available to offer the service.

3.3.2 Accuracy

The results of the calculation should be accurate and should not have any misleading or unavailable routes. Otherwise the reliability of the application will reduce a lot and the application will have a negative impact from the users.

3.3.3 Mean Time to Recover

If the server break down for some reason, it should be back on at least within 24hours. More the time it takes, more will be the damage on the user base of the application.

3.4 Performance and Security

3.4.1   Response Time

When the user inputs are taken, the calculated results should be available on the phone in less than 10 seconds. Users don't want to wait a long time for the results to appear. However the available internet connection also affects this response time.

3.4.2   Transactions per second

The backend database can support up to 80 transactions per second. But it will depend on the use case as well.

3.4.3   Resource Utilization

The application won't use much of the memory of the mobile device and if the GPS option is used, then that will cause the phone battery to drain much faster. Same if the data connection is weak, the battery drainage will be higher.

3.5 Supportability

3.5.1   Coding Standards

Coding standards will be maintained throughout the code so that it'll be readable and understandable by anyone to make changes in the future. Therefore standard naming conventions will also be used.

3.6 Design Constraints

3.6.1   Optimized Algorithms

As this application calculates optimum routes in real time, the algorithm for that should be much efficient because users won't want to wait long to get the results. Therefore this should have a considerably high efficiency.

3.6.2   Performance of the Server

There will be a backend server which processes all the functionalities of the system. This should be available all the time because this application relies very much upon that. The server should have a considerable amount of performance because it should be able to handle all the user requests simultaneously and calculate the routes for all of them efficiently.

3.6.3   Supporting Devices

As Google map API v2 is supported only by phones which has android version 2.2 or higher, this application will run on any android phone which is higher than android version 2.2. Therefore it covers a huge percentage of the total android phones available in the market.

3.6.4   GPS facility

To use the GPS option the phone should have a GPS system available. However majority of the phones has this option and almost all the phones that comes under the set of devices mentioned in 3.6.3 will have this facility.

3.7 On-line User Documentation and Help System Requirements

A user guide will be included in the application itself as an offline facility. Therefore an online help will not be necessary.

3.8 Purchased Components

This application will be using google free license for the initial phase and if the number of transactions exceeds the free limit, there will be a necessity to purchase an enterprise license. However this won't be

used for the initial phase.

The application will be hosted in a free webserver initially and for increasing the performance, reliability and availability of the system, a paid server will be necessary.

3.9 Interfaces

3.9.1 User Interfaces

There will be user interface for selecting the starting point and destination, a map view for marking and displaying those two points, an interface to show the results and filter or sort if needed, the navigation view of the selected routed, view the past user experience(comments), comment on the selected routes, view the user guide and the splash screen.

3.9.2 Hardware Interfaces

Any android device running android version 2.2 or higher will run this application because they also have enough memory to support the system.

3.9.3 Software Interfaces

Android APIs, Google Maps API V2, Google places API will be used and

3.9.4 Communications Interfaces

Client (Mobile Device) - server communication will be done through HTTP/HTTPS protocols.

3.10 Database Requirements

The database should be able to handle many users simultaneously and the transactions per second rate should be high enough to minimize the response time of the system.

3.11 Licensing, Legal, Copyright, and Other Notices

As Google APIs for maps are used, adhering to their terms and conditions of service is mandatory.