SOFTWARE REQUIREMENTS SPECIFICATION

FOR THE

ANDROID APPLICATION FOR TRAVELLING SALESMAN PROBLEM (TSP)

**Prepared for:**

CSCI 6838

**Instructor:**

Dr. Alfredo Perez-Davila

**Team#2**

1. Shravani Reddy Tappati

2. Yelamanchili Tejasri

3. Divya Karukonda

4. Nikunj Tibdewal

**Table of contents**

**Contents Page No**

1. Introduction
   1. Purpose 3
   2. Scope 3
   3. Definitions, Acronyms and Abbreviations 3
   4. System Overview 3
   5. References 3
2. System Overview 4
3. Use Cases

3.1 Overview of the application 4

3.2 Use Case Diagram 6

4. Functional Requirements

4.1 Searching the Optimized Route 7

4.2 Fixing the path between the locations 8

5. Software Requirements 9

6. Non Functional Requirements 9

7. Other Requirements 9

**1. Introduction**

* 1. **Purpose**

The Purpose of the Software Requirements Specification is to provide the technical, Functional and non functional features, required to develop a Google Android Mobile Application. The entire application is designed to provide user flexibility for finding the shortest and/or time saving path.

* 1. **Scope**

The scope of this project is to develop an application for Traveling salesman problem (TSP) on Android Mobile operating System. This allows the user to enter multiple locations to visit, and find the optimal route between those locations and then list them on Google map.

**1.3 Definitions, Acronyms and Abbreviations**

TSP - Travelling Sales Person

UC - Use Case

API - Application Programming Interface

**1.4** **Overview of the document**

The Requirements Specification captures system requirements for the following areas: functionality, usability, reliability, performance, supportability and the design constraints.

**1.5 References**

None

**2. System Overview:** This section provides a detail overview of the TSP application.

* 1. **Application Perspective**

This application allows user to input locations by typing in the addresses, as one would do in Google maps (maps.google.com). Along with the locations, the user can specify a relative order among his locations. The traveler can also fix the order for a part of his journey. The output produced is a map with an optimal route. The route is most likely the shortest route possible.

* 1. **Operating environment**

1. Windows operating system for development.
2. Android Mobile Operating System for deployment.
   1. **Dependencies**

The project uses Android Platform. Generally, Android applications are written in Java.

* 1. **User Classes and Characteristics**

Mobile application for TSP should cater to the following user classes.

**Primary User** – It defines a Salesman or a traveler who wants to visit different locations with an optimized route.

**Developer** – The role of a developer is to maintain the application. It is assumed that the user is adept in Google Maps API, Android, XML and Java.

**3. Use Cases**

**3.1 Overview of the application:**

**Android Application for TSP**

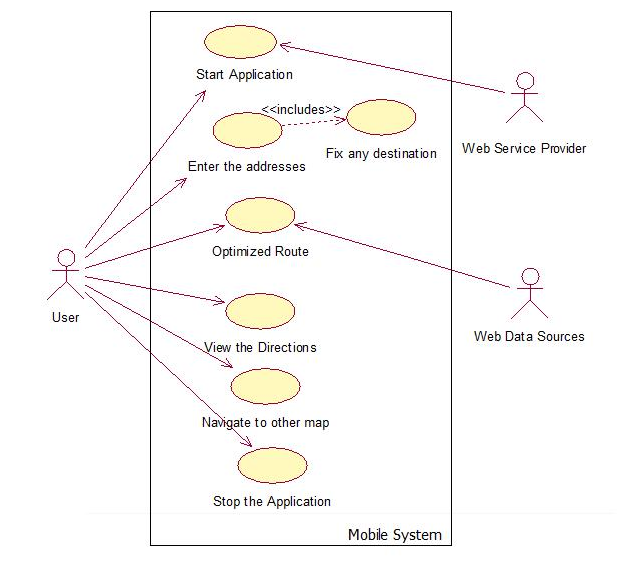
**Sales Person**

**Figure (a)**

|  |  |
| --- | --- |
| **Name** | **Enter Address** |
| Summary | Actor has to enter or choose among the locations he wants to visit. |
| Rationale | Once the actor enters the addresses, he has an option to choose route manually (Fix Route) or an optimized route or both. |
| Users | Sales Person, Traveler |
| Precondition | Actor must have access to a running Android application. |
| Course of Event | It is a one to one relation.  Actor directly interacts with the Android application and enters the addresses to visit the respective destinations. |
| Post Condition | When the set of addresses have been saved, the actor using the Android application for TSP has two choices to make, he can either choose to fix the route according to his wish or follow the optimized route or both. |

**3.2 Use Case**

**Diagram for TSP:**

****

**Figure (b): Use case Diagram for Android Mobile Application for TSP**

|  |  |
| --- | --- |
| **Name** | **Use Case Diagram for TSP** |
| Summary | Shows the UC Diagram for Androids mobile application for TSP. |
| Rationale | Upon running of the application, the user can enter the addresses, where he/she has an option for fixing the path.  The optimized route is obtained using the web data sources, which can be plotted in the Google maps or the directions. |
| Users | Sales Person, Traveler. |
| Precondition | Actor must have access to the Android Mobile phone with web service provided. |
| Course of Event | It is a one to one relation. Actor directly interacts with the Android application and enters the address to visit. |
| Post Condition | Using the Android application for TSP, the actor has an option to either choose to fix the route or an optimized route or both. |

**4. Functional Requirements**

This section of the document illustrates different functions provided by the application:

**4.1 Searching the optimized route**

**4.1.1 Description:**

This feature of the application shall provide the optimized route for the given addresses. Our application uses the Google Maps API to plot the route.

**4.1.2 Rationale:**

Once the user starts the application, he/she has an option for entering the addresses. When the user submits his/her request, the information is requested from the web data sources, which returns the data in XML format. The data obtained is plotted on the Google Maps using pins.

**4.1.3 Requirements:**

In order to use this application, the User should have a mobile phone which runs on Android Mobile Operating System, with an active internet connection. Also the web server of the data source should be up and running as data is retrieved from it once the user uses this function.

At any point of time only one search can be performed on the map. The search should include valid addresses.

**4.2 Fixing the Path between locations and obtaining the optimized route:**

**4.1.1 Description:**

This feature provides the user to select an optimized route for the addresses given and fixing any path between the desired locations. The user might not be comfortable with the optimal path produced by the application and hence a situation arises where he/she can select a path and fix the route. This application uses the Google Maps API to plot the route.

**4.1.2 Rationale:**

Once the addresses are entered, the user has an option for fixing the locations which he would like to visit first. Once the user submits this request the information is requested from the wed data sources, which returns the data in XML format. The data obtained is plotted on the Google Maps using pins.

**4.1.3 Requirements:**

In order for the user to able to use this feature of the application, he/she should have a mobile phone which runs on Android Mobile Operating System with an active internet connection. Also the web server of the data source should be up and running as data is retrieved from it once the user uses this function.

At any point of time only one search can be performed on the map. This search should include a detailed and valid address.

5 Software Requirements

5.1 Java Eclipse 3.5

Since Android supports JAVA, the application was built on it.

**5.2 Google Android SDK 1.0**

The Android platform is a software stack for mobile devices including an operating system, middleware and key applications. Developers can create applications for the platform using the Android SDK. Applications are written using Java programming language.

### **5.3 XML**

### XML is a simple, very flexible text format which is designed to carry data, not to display data.

**6. Non Functional Requirements**

Fast internet connection is not mandatory, but it would increase the performance of the application. Also, after every user request, new maps are needed to be loaded.

**7. Other Requirements**

As Google Maps API is being used for this application, it is mandatory that we abide by the terms of use specified by Google.