**Get - A - way**

Software Requirements Specification

Version 1.0

27 Feb 2016

|  |  |  |
| --- | --- | --- |
| Sl.No | Name | USN# |
| 1. | Lisa Sarah Thomas |  |
| 2. | P Sai Vishwas |  |
| 3. | Shreyas G |  |
| 4. | Raghavendra G |  |
| 5. | Nagachandra R. Upadhya |  |
| 6. | Srinivas Akhil Malella |  |
| 7. | Nagashree A.C |  |
| 8. | Sharvel Mithali D’souza |  |
| 9. | Rakshaa S Chetty |  |
| 10. | Srinivas N Shavi |  |
| 11. | Parag Karguppikar |  |
| 12. | Mohammed Aizaz Ahamed |  |
| 13. | Reddammagari Sree Soumya |  |

Guide:

# **Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Approver** | **Comments** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Revision History

Document Approval

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

2. General Description

2.1 Product Perspective

2.2 User Characteristics

2.3 General Constraints

2.4 Assumptions and Dependencies

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware and Software in terms of how they would interact or how they would be executed

3.2 Functional Requirements

3.2.1 <Functional Requirement or Feature #1>

3.2.2 <Functional Requirement or Feature #2>

3.3 Non-Functional Requirements

3.4 Other Requirements

# 

# 

# 

# 

# **1. Introduction**

The first step to designing any successful product is identifying the requirements. It is critical for the successful development of the product and ensures that there are no major problems in downstream activities, since it gives the designers an in-depth understanding of the product and its requirements.

## **1.1 Purpose**

## Software requirements specification requires understanding the problem, the domain, the needs and the constraints involved. The requirements are classified into product requirements, process requirements, functional and nonfunctional requirements, as well as domain requirements and any other requirements are identified. This helps in prioritizing the requirements. SRS identifies functionality, external interface, performances and attributes.

## **1.2 Scope**

**Get-A-way** is Tourism Application that will allow users to select places of preference and provide them with an optimal path (In website as well as mobile app). Suggestions regarding places to visit are given relevant to the projected path. Users are able to access nearby basic amenities such as restaurants, mechanic shops, mall, hospitals etc along the route only on the app. The application has a Tourism based Chabot to make queries about the travel on the app.

This will not be supported for running on an IOS platform.

## 

## **1.3 Definitions, Acronyms, and Abbreviations**

Some of the Abbreviations and acronyms used are :

|  |  |
| --- | --- |
| Configuration | Android |
| **iOS** | **iPhone Operating System**  iOS (originally iPhone OS) is a mobile operating system created and developed by Apple and distributed exclusively for Apple hardware. |
| **API** | **Application Program Interface** is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact and APIs are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together. |
| **GPS** | **Global Positioning System** is a space-based navigation system that provides location and time information. |
| **FR** | **Functional Requirements** defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. |

**1.4 References**

The references involve some of the books and websites we can use for the development of the app. Existing app and website with similar functionality are also being referred.

Sites:

[1] [http://www.w3schools.com](http://www.w3schools.com/)

[2] [http://developer.android.com](http://developer.android.com/)

[3] [https://thenewboston.com](https://thenewboston.com/)

[4] [http://www.vogella.com](http://www.vogella.com/)

[5] <https://developers.google.com/maps/documentation/android-api>

Books:

[1] The Android Developer's Cookbook

[2] Head First Android Development

[3] Android Programming: The Big Nerd Ranch Guide

Apps:

[1] Karnataka Tourism

[2] Incredible India

[3] Karnataka Travel

**2.1 Product Perspective**

Our project is a travel app with a few features that are not regularly part of travel apps

like emergency services. We are building both an App and a website.

.A lot of pressure is on the API and on the intricacy of the algorithms we use. It's also an interactive one since the front-end is a website/Android app which is clear and transparent, this enables effective usage of the app/website for the users.

So our project, when compared with other related projects, is similar to a traditional travel app with a few interesting features that make it different from the rest.

High Level Interactions:



**2.2 User Characteristics:**

Our app is intended to tap the potential available in the tourism market. The users of our app would generally be people who have a keen interest to travel. It might be bikers on a road trip, a family on an outing or a group of trekkers. One of the standout characteristic of the users is that they would be moving frequently from one tourist place to another. Since they would generally be on the go, the users are expected to have an internet connection to avail the services provided by our app. They can use the app on a smartphone to use its services in case of any emergency they encounter while travelling.

**2.3 General Constraints:**

Technical:

* Requires an always-on data connection- As it’s mobile, users constantly lose coverage(Assuming high speed network access)
* Memory and power consumption
* Draining memory and battery often leads to negative reviews and lower adoption and acceptance rates
* The expected duration of the journey shown between two places is only an estimate.
* In addition to this location-based information which is rendered by the GPS feature of an android phone can easily be manipulated.

Non Technical :

* Schedule - We have to be ready to demonstrate at the trade-show in by mid of April. The final delivery date is fixed based on input from the client and cannot be changed. It is now up to us to design a system that can be built within the required schedule.
* Team composition and make-up - In some cases there may be requirements that specific personnel be used or not used during project. For example, someone may be unavailable, already committed to another project or you may be required to include specific individuals, perhaps for training purposes.

**2.4 Assumptions and Dependencies:**

Assumptions :

1. It is assumed that the user is connected to the server at all times.
2. Availability of data which is obtained from Google Maps.
3. The data obtained from Google Maps is assumed to be accurate.
4. The expected duration of the journey shown between two places is only an estimate.
5. The Global Positioning System is assumed to be accurate.
6. The suggestions given by the application are dependent on user feedback.
7. The feedback of every user is assumed to be genuine.

Dependencies :

* Google Maps API and Google Places API

Google API services will be an integral part of the project. Especially the Google Maps API and Google Places API. The APIs can be used to enrich the app with high-accuracy location reporting and other location related services.

* Android Studio
* SQLite
* Bootstrap

**3.1 External Interface Requirements:**

**3.1.1 User Interfaces**

WEBSITE:

UI-1: Web application shall permit complete navigation and places selection using the keyboard alone, in addition to using mouse and keyboard combinations.

UI-2:HomePage of website has a form where user can select places to visit .

UI-3: A save my trip page where when user is done selecting places initial optimal route and an option of saving their trip appears which allows unregistered users to register and already registered users to login and save their trip.

UI-4:Login page which verifies registered user credentials and redirect them to their home page

UI-5: User’s home page has user’s saved trips.

APPLICATION:

UI-1:Login feature which verifies registered user credentials and redirect them to user’s home page where user can select one of his saved trips.

UI-2:Google Map with optimal route and nearby amenities

CHATBOT:

Tourism based Chatbot to make queries with the database stored on server about the travel.

**3.1.2 Hardware and Software in terms of how they would interact or how they would be executed**

ANDROID:

Hardware:

The end device must support GPS functionality. User enters places to visit. Application geocodes it in the browser. The app then sends the data to the server. The server responds with some data, such as nearby points of interest.

Software:

The mobile application communicates with GPS to access geographical information of selected places.GPS communicates with the server to fetch optimal path.

**3.1.3 Software Interfaces**

SI-1: Database - The system shall communicate with a database through a programmatic interface for the following operations:

SI-1.1: To manage registered users.

SI-1.2: To allow a user to save their trip.

SI-1.3: To make queries on database about travel.

**3.2 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR1 | Optimal travel route | The app/website shall allow the user to enter the destinations points. An optimal path is generated to cover the places the user desires to visit. |
| FR2 | Suggestions | The app shall discover interesting points of tourist attraction in the user’s vicinity and suggest them as possible destinations. |
| FR3 | Access to amenities | The app shall provide a user-friendly interface to basic amenities such as hospitals, atm, petrol bunks, etc. en route to the destination. |
| FR4 | ChatBot | The app shall support a chatbot which suggests places to the user depending on their requirement(where to visit, distance constraint, rating of a place) |

**3.3 Non-functional Requirements**

* **Maintainability** : The server should be maintainable by the admin with ease.
* **Reliability** : The application must not crash abruptly. It should handle all the use cases and exit, if it must, with grace.
* **Testability** : The application must be testable with various arguments.
* **Understandability** : The application must be easy to understand by the end user and must feature a self descriptive interface, be it buttons or text boxes.
* **Performance** : Any and all delays involved should be minimal. This includes the user opening the webpage to plan a trip, the user planning the trip using the interface, the user using the application to navigate through the determined route and also the user trying to find the amenities he/she needs during the trip around his/her current location.
* **Availability** : The server needs to be up 24 x 7 to service user requests.
* **Robustness** : Multiple users might be logging in to plan a trip. The server and the application must be able to handle this load.
* **Safety** : Any information regarding the user must be transmitted securely to the server for processing. A proper login mechanism has to be implemented.

**3.4 Other Requirements**

Since our product consists of a webpage and an android application, it is necessary that these two parts of the project need to be synchronised when it is being used. Therefore, a communication interface which handles the responsibility of communication between these two parts is required.

It is also essential that the product is kept free from manipulation from external sources. This may result in corruption or loss of data. Hence, safety requirements must also be taken care of.