TOURGURU : TOUR GUIDE MOBILE APPLICATION FOR TOURISTS

19-018

Software Requirements Specification

R.D.T.N. Rajarathna IT16003538

Supervisor:

Ms. Asanthika Imbulpitiya

Co-Supervisor:

Mr. Nuwan Kodagoda

B.Sc. (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

TOURGURU : TOUR GUIDE MOBILE APPLICATION FOR TOURISTS

19-018

Software Requirements Specification

B.Sc. (Hons) Degree in Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2019

DECLARATION

We declare that this is our own work and this SRS does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
R.D.T.N. Rajarathna	IT16003538	

The supervisor/s should certify the proposal report with the following declaration.

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor: Date

…..... .....

Signature of the Co-Supervisor: Date

…..... .....

Table of Contents

List of Tables

Table of Figures

1. Introduction

1.1 Purpose

This document discusses the requirements for smart map interface component of TourGuru system. It gives the overall description of the software system and the solution provided. It also defines requirements for the development of the software system. This document is intended for the users, creators, regulatory bodies and other stakeholders who are involved in developing and deploying the system.

1.2 Scope

This document covers the requirements for the smart map interface component of the TourGuru system that includes the mobile application and the web application associated with it. It will discuss the features and technologies of each application which will act as the reference to the developers and stakeholders in selecting the best design.

1.3 Definitions, Acronyms, and Abbreviations

Table 1 - Definitions

Term	Definition
TourGuru	Name of the system
Roadtrippers, Toureazy, Tour Buddy, සිංහලංකා AR	Systems that has similar components
NoSQL	Not only SQL (Structured Query Language)
MongoDB	An open-source document database and leading NoSQL database.
NodeJS	JavaScript framework for creating web applications

Table 2 - Acronym / Abbreviations

Term	Definition
SRS	Software requirements specification
AR	Augmented reality
API	Application programming interface
GPS	Global Positioning System
OS	Operating system
WIFI	Wireless Fidelity
RAM	Random Access memory

1.4 Overview

TourGuru is a tour guide assistance system that will help increase the touring experience of the user by helping them with some tasks. The route analysis component of the system will alert the user of nearby tourist locations throughout the journey. It will analyse route by collecting route information from many users.

1.4.1 Main Goals

Main objective of the TourGuru mobile application is to help travellers navigate through places of interest without much hassle. It needs to be a simple but very exciting and interactive application. It focuses mainly on being a virtual tour guide by narrating details about the places and answering any questions the user might get during their travels.

1.4.2 Specific Goals

- Guiding a tourist through a route by intelligent routing.
- Collect user flagged destinations.
- Merge Details with maps platforms.
- Machine learning is used for tracking usual locations, destinations and map route accordingly.
- Code the walk and Driving mode accordingly.
- Use of crowd sourced information further development of system.

1.4.3 Users

The main user group of this system are tourists, tour guides, drivers. This includes regular commuters, or people with driving jobs like taxi, bus drivers.

1.5 Organization of SRS

The rest of the document discusses the research component. The chapter 'Overall Descriptions' describes the overall description of the product which compares the product with other products, functions of the product, user characteristics, constraints, assumptions & dependencies and the order in which the requirements will be implemented. The chapter is 'Specific Objectives' discusses about the requirements of the proposed system. It describes the functional and non-functional requirements. It also discusses the interfaces of the system and the design, system attributes, reliability, availability, security and maintainability aspects of the system.

2. Overall Descriptions

The system of the TourGuru is a tour guidance system for tourits that informs the user of the tourist locations throughout the journey. The system comprises of a mobile and web application. The mobile application is based on the android platform and will collect user satisfaction information throughout the journey and post it to the web application after the journey has ended with the user's consent. The web application will be based on the NodeJS platform to support multiple concurrent requests from many users. The web application will store the information obtained from the mobile application in the database. The database will contain the raw data collected from many users. A service will then process this data to calculate coordinates throughout the journey and the resulting geolocation mapped information is stored in a database. This information would later be given to the users as a guidance when they take journey through the same route. To do this, the mobile application would download the journey data based on the destination that the user has selected and use that to provide speed information based on the user location. The system would also rate the user on how well they use the routes, locations, application.

2.1 Product Perspective

There are some tour guide assistance systems that has similarities with TourGuru system.

Table 3 - Product similarities

Features	TourGuru	Roadtrippers	Toureazy	Tour Buddy	සිංහලංකා AR
Intelligent Trip routing (automatic route creation)	✓	✓			
Trip editor (Add or update custom places)	✓	✓			
Categorize locations (monuments places, restaurant etc)	✓	✓			
Map Filters	✓	✓			✓
Shared user activity	✓	✓		✓	
Traffic management	✓			✓	
Narrations or alerts on point of interest	✓		✓		
Waypoint management	✓	✓			✓
Collaborator management	✓	✓			
Distance slider for radius adjusting (proximity alert and activation)	✓	✓			

2.1.1 System interfaces

Google maps API will be used for location-based information such as displaying user routes in the map. This will require generating a google API key to be used with the application. There will also be integration with google maps android application when the user wants to use it to navigate while using the system.

2.1.2 User interfaces

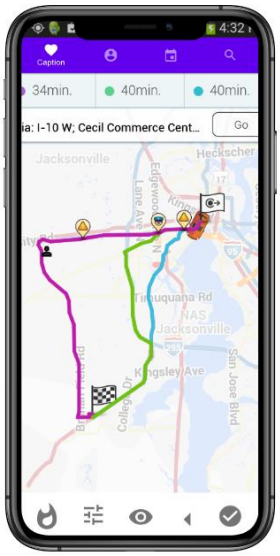

Figure 1- start tour and stop tour

2.1.3 Hardware interfaces

The application can be used on a GPS enabled android device. It will be used to collect driving information from the user and to provide location-based speed and other alerts. It should be also being capable of internet connectivity, for communication with the server.

2.1.4 Software interfaces

The system will use a NoSQL database such as MongoDB as the database to store information collected from users. It will also use google maps application for navigation purposes.

2.1.5 Communication interfaces

The GPS sensor in the mobile phone will provide location information to the application through OS interfaces and the data modem in the mobile phone will help communication with the API since WIFI access cannot be expected while travelling.

2.1.6 Memory constraints

The application will use about 200MB of internal/external storage on the mobile phone and around 2GB of RAM while the application is running.

2.1.7 Site adaption requirements

This system is provided as a SaaS service, that is the system is managed by the service provider, so the user is not required to additional configurations to do this.

2.2 Product functions

2.2.1 High Level Architecture Diagram

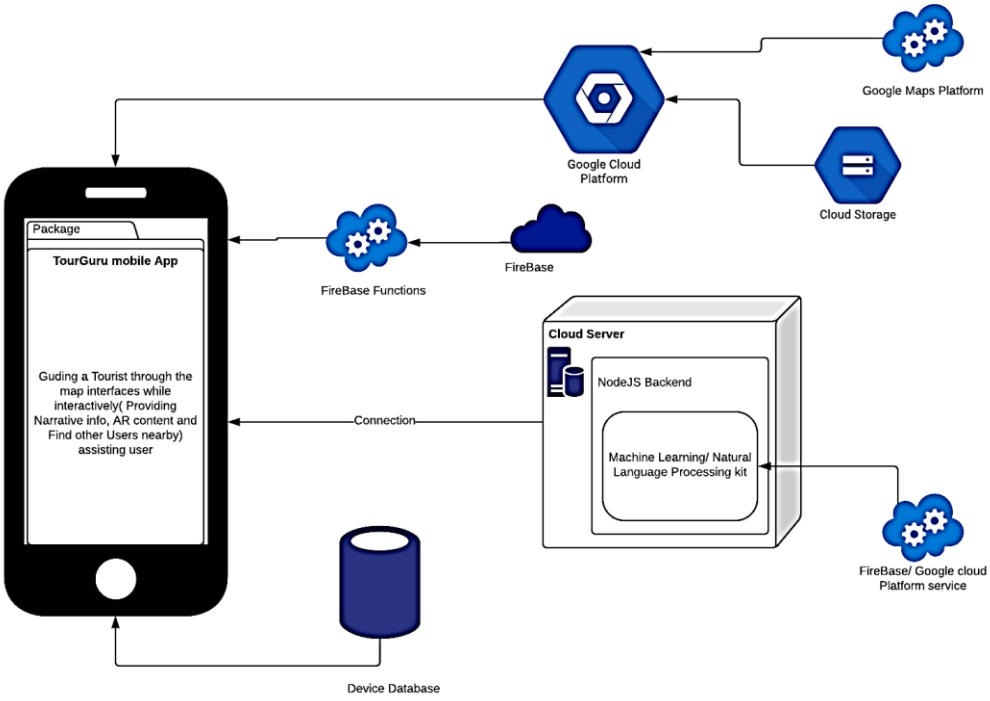

Figure 2 - System Architecture

2.2.2 Use Case Diagram

The requirements for the component are described using a use case diagram followed by the use cases themselves.

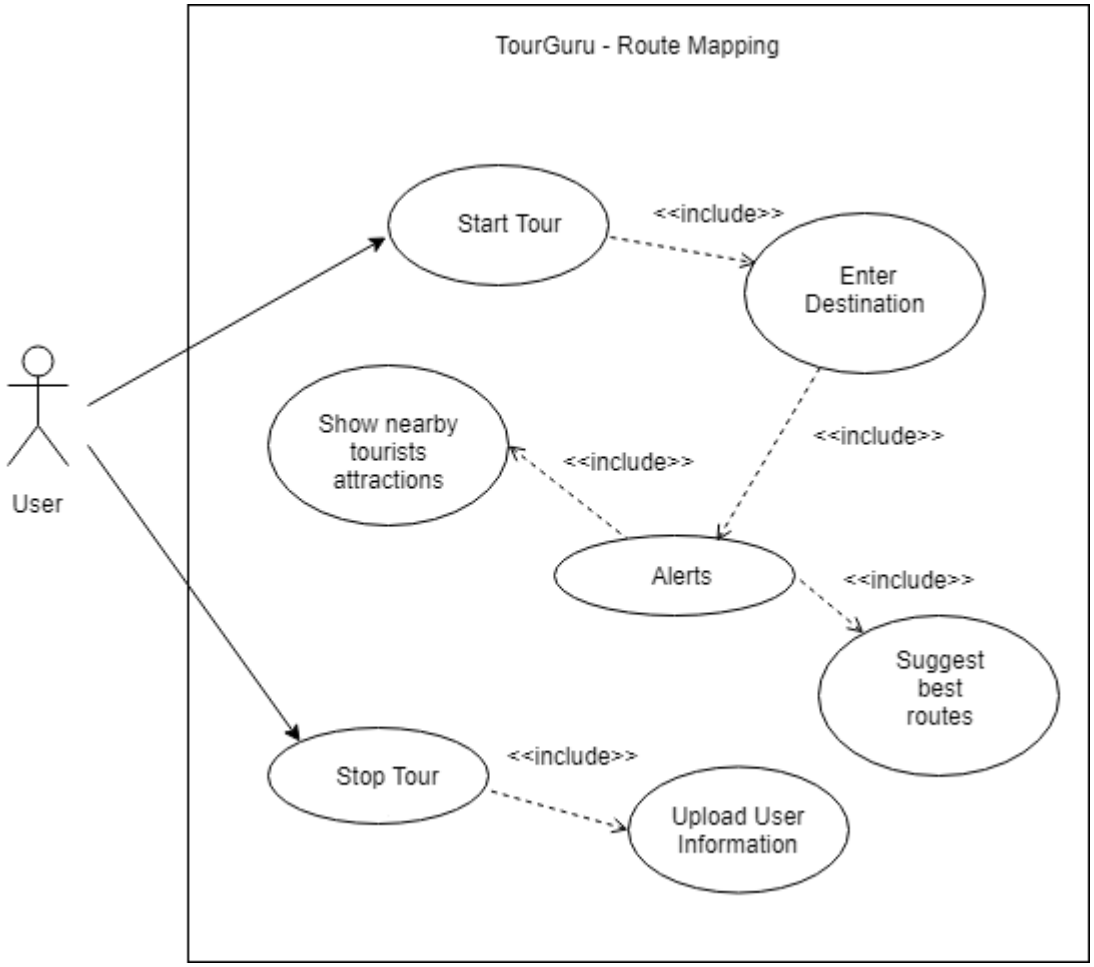

Figure 3- Use case diagram

2.2.3 Use Cases

Table 4 -Use case scenario 01

Use case ID	UC_01	
Use case name	Start tour	
Goal in context	User starts the tour	
Pre-condition	The application started and internet and GPS connectivity are available.	
Post-condition	None	
Primary actor	User	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	User starts the tour
	2	User receives alerts in screen.

Table 5 -User case scenario 02

Use case ID	UC_02	
Use case name	Stop tour	
Goal in context	User starts the tour	
Pre-condition	The application already started, and internet and GPS connectivity are available.	
Post-condition	None	
Primary actor	User	
Secondary actor	None	
Main flow	**Step**	**Action**
	1	User stops the tour
	2	Application request confirmation from user.

2.3 User characteristics

The users of this system would be regular commuters with a smartphone. This would include people with a wide range of computer skills. However, we presume that the user would be familiar with the usage of an android smartphone.

2.4 Constraints

The development of the mobile application requires that google play services are correctly installed in the devices. A GPS mock location service must be used to emulate the navigation when developing the application. The development requires Android Studio. The minimum requirements to run android studio could be found online.

The application will constantly collect data from users. However, posting this information constantly to the API will consume data and will also exhaust the battery. This information can instead be collected for a journey and sent. In that case, the file size could be large and should be compressed to use less data.

2.5 Assumptions and dependencies

It is assumed that,

- the android smartphone can consistently update GPS information within a considerable distance and with accuracy and these coordinates are accurate between different journeys.
- integration with the google maps application is possible for navigation purposes.
- the application can refresh itself with changing GPS coordinates with low latency.

2.6 Apportioning of requirements

The primary requirements of the system are the collection of usage data from users and displaying to the user back to the user based on his location / destination. The methodology used in calculating the best routes is yet to be decided. This needs to be included in the initial release of the application. The system for user behaviour is a desirable requirement and does not need to be in the initial release of the software however could be done in subsequent releases.

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

The mobile application will be used by the end users and will contain two UIs for this component. These are described below in detail.

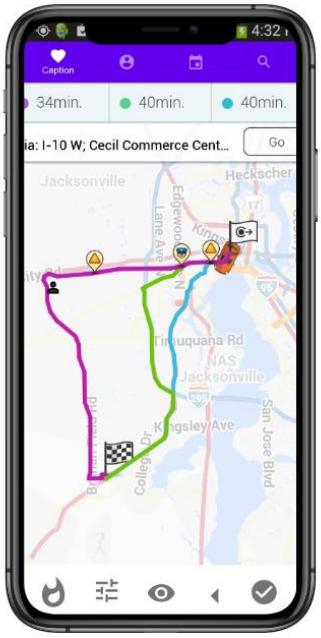

Figure 4 - Start and stop tour

3.1.1.1 Start tour

This user interface will provide the ability for users to select the destination for the journey and start the journey. It will also indicate in the map, the current location of the user. Once the user selects the start journey, the app will start providing speed related information. This user interface will provide the ability for users to select the destination for the tour and start the journey. It will also indicate in the map, the current location of the user. Once the user selects the start tour, the app will start providing tour related information.

3.1.1.2 Stop tour

When the tour has ended, the user can use the stop button to end the tour and will provided with a screen indicating the attributes of the tour like detailed description about location, nearby locations, time etc.

3.1.2 Hardware interfaces

The mobile application requires an android mobile phone with internet and GPS capabilities to collect location information and post it to the web application. The web application requires a cloud server to run on that is accessible from the mobile application through the internet.

3.1.3 Software interfaces

The mobile application will be built on the android platform. The major functionalities of the system will be based on location. For tracking the current user location, the mobile application will use (eg: FusedLocationProvider) library which provides a battery efficient location API. The app will use SQLite database for temporary storage on the mobile phone especially when we want to bulk collect information and send it to the server and as a backup when offline.

3.1.4 Communication interfaces

The mobile application requires the usage of GPS to fetch location data, but this communication is handled using libraries that hides the underlying communication between the app and the sensor. It also requires internet connectivity through the data modem/ router to access the rest services of the server. This is handled through the underlying OS implementation that hides the complexity from the application.

3.2 Classes/Objects

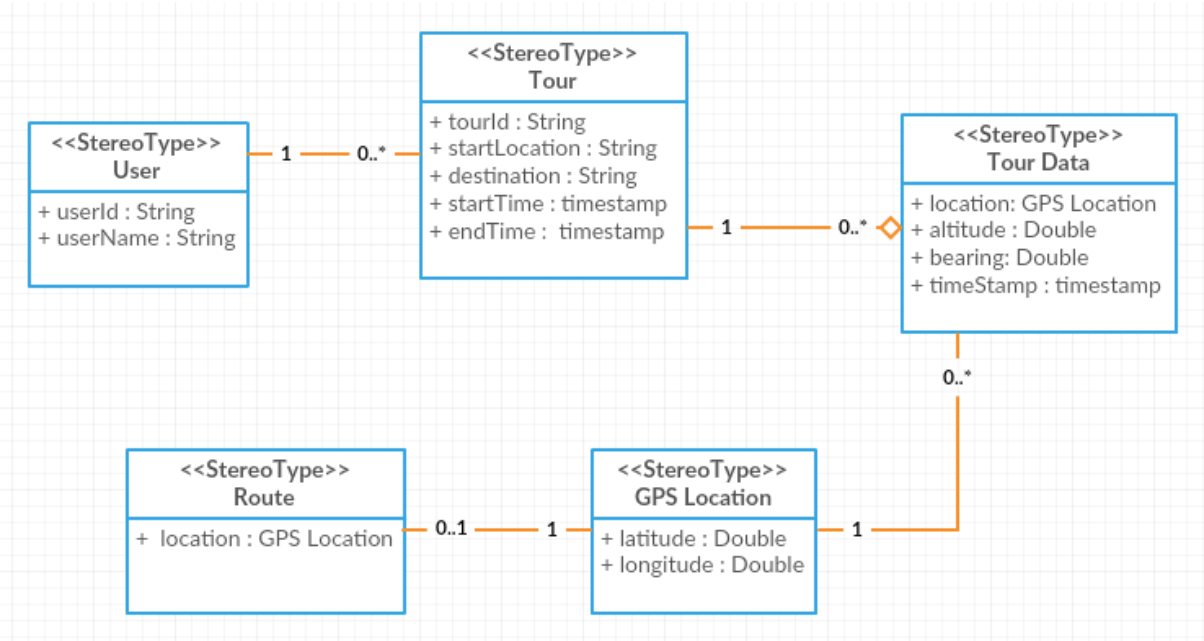

Figure 5- Class Diagram

3.3 Performance requirements

The API will receive concurrent requests from many users with the route information therefore it should be able to handle these concurrent requests and process them without blocking other users. NodeJS backend is suitable for this as it can serve multiple asynchronous requests.

3.4 Design constraints

The mobile application will use audio alerts to alert the user, therefore it should be running in the background enabling the user to use navigation application while driving. However, it should be also possible to start and stop the journey. This would require integration with navigation applications so that the transition between the use cases would be seamless and the application would not be an overhead to the user.

3.5 Software system attributes

3.5.1 Reliability

The system will not give out speed suggestions until a considerable amount of user data is collected and a reliable prediction could be made. A threshold needs to be set for this to find the most stable and reliable speed predictions.

3.5.2 Availability

The system will have an availability of 95%. Hosting provider AWS promises a 99% availability for the application, but maintenance of the application would be required during the initial stages of the application for purposes like data repairs.

3.5.3 Security

The system deals with location history of users. To protect the privacy of users, this data should be protected. There are layers to secure in the system. The communication between the mobile application and the server should be encrypted to protect sniffing by malicious parties. The mobile application uses a local database therefore it should also be secured from malicious access in case the device is lost/ crash.

3.5.4 Maintainability

The system collects large amount of data overtime and needs to be frequently archived when not used. This would require a background service that could automatically archive this data in a backup database after it has been processed. It is best to use backup database for emergency.

3.6 Other requirements

The application will use the tour coordinates to find bends and then use the GPS information of users to learn a neural network that can later predict path at different bends.

4. Supporting information

4.1 References

[1] Juan M. Corchado, Juan Pavón, Emilio S. Corchado and Luis F. Castillo.(2004); Development of CBR-BDI Agents: A Tourist Guide Application.

[Accessed 2019-03-02]

[2] Fang-Ming Hsu, Yu-Tzeng Lin, Tu-Kuang Ho: Design and implementation of an Intelligent Tourist Attractions System (ITAS) for tourist attractions: The integration of Engel–Blackwell–Miniard (EBM) model, Bayesian network and Google Maps

Available:https://www.researchgate.net/publication/220216082_Design_and_implementation_of_an_intelligent_recommendation_system_for_tourist_attractions_The_integration_of_EBM_model_Bayesian_network_and_Google_Maps

[Accessed 2019-02-26]

[3] A Trip Reconstruction Tool for GPS-based Personal Travel Surveys