
Software Requirements Specification

for

MyTripDiary

Version 1.0 approved

Prepared by

Tai Chen An

Tan Ming Rui, Ezra

Goel Armaan

Sim Guanyu

Xu Yinfeng

Nepal Aaradh

Nanyang Technological University, Team No Idea

1 Apr 2023

Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope.....	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions.....	2
2.3 User Classes and Characteristics.....	2
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation.....	3
2.7 Assumptions and Dependencies.....	3
3. External Interface Requirements	4
3.1 User Interfaces.....	4
3.2 Hardware Interfaces	4
3.3 Software Interfaces.....	4
3.4 Communications Interfaces.....	4
4. System Features	4
5. Other Nonfunctional Requirements	4
6. Other Requirements	5
Appendix A: Data Dictionary	5

Revision History

Name	Date	Reason For Changes	Version
Goel Armaan	01/04/23	Fulfill TBD list and split document	1.0

1. Introduction

1.1 Purpose

This document specifies the software requirements for the application MyTripDiary Version 1.0.

MyTripDiary is a Mobile app that enables its users to keep track of their daily commutes allowing them to budget better and save time by informing them about alternative faster, cheaper routes and modes of transport. Users can visualize their commute routes, how much they spent on them and how long their trip took. MyTripDiary gives an estimate of the cost of each trip based on factors such as start location, destination, mode of transport (including Private Car, Taxi, Bike, Public Transport and Walking), and transport fares.

1.2 Document Conventions

Priority of Requirements: The priority of higher-level requirements is assumed to be inherited by detailed requirements unless explicitly stated otherwise. The priority of all higher-level requirements is assumed to be equal unless explicitly stated otherwise.

Level 1 Heading: Font Family: Times, Font Size: 18, Font Weight: Bold

Level 2 Heading: Font Family: Times, Font Size: 14, Font Weight: Bold

Level 3 Heading: Font Family: Times, Font Size: 12

Content: Font Family: Arial, Font Size: 11

Throughout this document, 'the app' or simply 'app' refers to MyTripDiary unless explicitly specified otherwise.

Other conventions and definitions of terms used in this document can be found in **Append A: Data Dictionary**.

1.3 Intended Audience and Reading Suggestions

This document is intended for users of the app, the software developers of the app, the documentation writers, the project managers, the marketing staff, and testers.

This document details the description, use cases, functional and non-functional requirements, interfaces, constraints of the app. It is intended to be read in sequence by all stakeholders involved in this app.

1.4 Product Scope

MyTripDiary is an app that aims to enable people to organize and track their daily commutes. With MyTripDiary, daily commutes become faster, cheaper, more reliable, and more personalized. The simple to use interface, allows its users to take control of how they reach their destination making commuting more than just going from point A to point B.

1.5 References

- I. Source Code (GitHub): <https://github.com/ardnep/SC2006-Project-No-Idea>
- II. React Native: <https://reactnative.dev/>
- III. Expo: <https://expo.dev/>
- IV. Firebase: <https://firebase.google.com/>
- V. Firestore: <https://firebase.google.com/docs/firestore>
- VI. Google Maps API: <https://developers.google.com/maps>
- VII. OneMap API: <https://www.onemap.gov.sg/docs/>
- VIII. FareFinder API: <https://www.taxifarefinder.com/api.php>
- IX. TollGuru API: <https://tollguru.com/toll-api-docs/>

2. Overall Description

2.1 Product Perspective

MyTripDiary is a new app and is not part of a larger product family or product line.

2.2 Product Functions

MyTripDiary has the following main functions:

- I. The app allows users to save, edit and delete trips
- II. The app gives an estimate of how much a trip will cost
- III. The app allows the user to visualize their daily commutes showing them information including but not limited to how much the user spent on public transport in a month, on average, how long they spend on taxis, etc.
- IV. The app displays a history of each individual trip that the user has completed
- V. The app allows a user to star a trip. A starred trip is pinned to the top of the list of saved trips
- VI. The app gives its user the flexibility to edit the price of a previously executed trip enabling them to adjust for any discrepancies existent in the price of the trip estimated by the app

2.3 User Classes and Characteristics

1. Young adults (18 – 30) and Teens (13 – 18): The primary target user class for MyTripDiary is teens and young adults who are tech-savvy and rely on daily commutes for school, work, or leisure activities. They are interested in budgeting their transportation expenses, saving time and money. They are also likely to be familiar with mobile apps.

2. General Users: MyTripDiary is also designed for general users of all age groups who want to keep track of their daily commutes, visualize their routes, and estimate the cost of their trips. These users may include working professionals, students, parents, and individuals who regularly commute for various purposes. They may have varying levels of familiarity with technology.

2.4 Operating Environment

The app operates in Android Version 5+ and above and iOS Version 11 and above.

The development environment for the app is as follows:

Development Environment
<u>Front-end</u> : React Native with Expo for Mobile Application
<u>Back-end</u> : Firebase

2.5 Design and Implementation Constraints

- I. The app relies on Firebase for user login and registration and Firestore as a database. Hence, it is subject to Firebase services being available. In cases where Firebase services are unreachable or down, the app may not function as expected. The solution to this is to wait for the Firebase services to be available.
- II. Initially, the app will only be available in the English language.

2.6 User Documentation

A demonstration of how the app works along with a guide to install and run the app is provided in the README.md file on the GitHub repository which contains the source code for this app (see Section 1.5 References).

2.7 Assumptions and Dependencies

- I. The app relies on external APIs to get information such as the route of a trip and the estimated cost of the trip. For maps and directions, it relies on the Google Maps API. For pricing, it relies on the OneMap, FareFinder and TollGuru APIs.
- II. It is assumed that the information obtained from these APIs is accurate.
- III. In cases where the API services become unavailable, the app may not meet some requirements specified in this document. Without the Maps API, the app will not allow the user to add or execute new trips. If one of the Pricing API fails however, the app will inform the user that the price cannot be estimated at the current time, but addition and execution of trips will still be allowed.

3. External Interface Requirements

3.1 User Interfaces

Please refer to the UI_Mockups folder for User Interface Mockups.

3.2 Hardware Interfaces

- I. MyTripDiary requires a mobile phone that supports either the Android or iOS platform. It is assumed that the information obtained from these APIs are accurate
- II. The device must have Internet connection capabilities to communicate with the database server and the APIs.

3.3 Software Interfaces

The app requires devices running Android 5.0+ or iOS 11 and above.

The architecture of the app is as follows:

A) Front-end

- React Native: framework to develop native mobile applications. Uses JavaScript.
- Expo: Helps build cross-platform apps easily. React Native code is written once and can then be deployed on both Android and iOS.

B) Back-end

- Firebase: provides functionality for user login and registration
- Firestore: A part of Firebase's service suite that provides a cloud NoSQL database for mobile and web development

3.4 Communications Interfaces

The communication with Firebase and Firestore is done via a REST API. The Google Maps and Pricing APIs use the HTTPS protocol with GET/POST requests.

4. System Features

Please refer to the Functional Requirements document

5. Other Nonfunctional Requirements

Please refer to the Non-Functional Requirements document

6. Other Requirements

Appendix A: Data Dictionary

Please refer to the Data Dictionary document