Requirements Analysis and Specification Document of *myTaxiService*

Angelo Gallarello, Giacomo Locci, Edoardo Longo

October 30, 2015

Contents

Contents **2**

1 Introduction **3**
1.1 Purpose . . . 3
1.2 Scope . . . 4
1.2.1 Goals . . . 4
1.2.2 Applications . . . 4
1.3 Definitions . . . 5
1.4 References . . . 5
1.5 Overview . . . 5

2 Overall Description **6**
2.1 Product Perspective . . . 6
2.1.1 User Interfaces . . . 6
2.1.2 Hardware Interfaces . . . 6
2.1.3 Software Interfaces . . . 7
2.1.4 Communication Interfaces . . . 8
2.1.5 Memory Constrains . . . 9
2.1.6 Site Adaptation Requirements . . . 9
2.2 Product Functions . . . 9
2.2.1 Web Application and Mobile Application (*User*) . . . 9
2.2.2 Mobile Application (*Taxi Driver*) . . . 9
2.2.3 Back-End Application . . . 10
2.3 User Characteristics . . . 10
2.4 Constraints . . . 10
2.4.1 Regulatory Policies . . . 10
2.4.2 Hardware Limitations . . . 11
2.4.3 Interfaces to other applications . . . 11
2.4.4 Parallel Operations . . . 11
2.4.5 Reliability Requirements . . . 11
2.4.6 Safety and Security . . . 12
2.5 Assumptions And Dependencies . . . 12
2.5.1 Domain Assumptions . . . 12
2.5.2 Taxi Driver Assumptions . . . 12
2.5.3 User Assumptions . . . 12

1 Introduction

1.1 Purpose

The current document represents the *Requirement Analysis and Specification Document* (RASD) of *myTaxiService* system. This document is intended to be read and used by everyone involved with the development of the software (i.e. developers, testers, project managers). Therefore its purpose is to provide the intended reader with a fulfilling guideline, that can be used as the main resource for the development of the system. An incredible amount of effort has been put into every section of this document; every aspect of the *system-to-be* has been thoroughly explored, meticulously analyzed and accurately studied so that every situation can be addressed in a unique way and can be interpreted without ambiguity.

In the following sections the system is broken down into its components, examining its functional and non-functional requirements; an in depth analysis of the relationship between the system and the external world is conducted throughout the document, stressing the assumptions that have been taken and highlighting the constraints that every requirement has to satisfy in order to have a functional and useful system.

The reader of this document will find UML class diagrams, sequence diagrams and use cases that provide all the needed information required to build the system. Furthermore, every use case is justified by the goals and assumptions that lead to the need of a specific use case.

Moreover, the delegating institution will find a reason for every design choice taken and therefore is able understand the necessity of a specific feature.

This way, each and every reader will have a clear understanding of the *system-to-be*, by finding a justification for every decision taken concerning any aspect that is worth of considering for a successful development of the investigated system.

1.2 Scope

1.2.1 Goals

The aim of *myTaxiService* presented in the RASD is to:

[G1] Intuitive service
Make the process of requesting a service intuitive for any kind of user.

[G2] Fast
Lower the time required from a user to request or book a taxi. Request time is less than 30 seconds, booking time is less than 1,5 minutes.

[G3] Availability
Enable the user to take advantage of the taxi service from a wide range of locations, namely from every location where the user has access to an Internet enabled device that can run the application.

[G4] Fair queues
Guarantee a fair management of the taxi queue by minimizing the idle time of every taxi driver using the service.

1.2.2 Applications

To accommodate these goals the following pieces of software are to be developed:

Mobile Application (*User*) The mobile application is available for all major mobile OS (Android, iOS, Windows Mobile, Blackberry). This one is specifically addressed towards the *User*, giving him a broad range of actions to perform. It gives the *User* the opportunity to manage his profile, request a taxi and make or dismiss a reservation.

Mobile Application (*Taxi Driver*) The mobile application is available for all major mobile OS (Android, iOS, Windows Mobile, Blackberry). This one is specifically addressed towards the *Taxi Driver*. It enables the *Taxi Driver* to accept a request issued by the system or dismiss it, signal any problem to the *Operator*.

Web Application The web application is thought as an alternate way, for the *User* to manage his profile, request a taxi and make or dismiss a reservation.

Back-End Application The back-end is the application that manages the request system and queue system. This also provides a control panel used by the *Operator* to have an overall view of the whole system, and enables him to issue warnings to *Users* or *Taxi Drivers*, and to respond to warnings issued by the *Taxi Driver*.

1.3 Definitions

Here is a list of the words used in this document:

User
He/She is the end user of the service that, once subscribed, makes use of the taxi service, by

Taxi Driver
He/She is another kind of customer of the service. He/She is the one that, once subscribed, will be commissioned with a work.

Operator
He/She is the person that is in charge of the customer service and interfaces with the customers, both the taxi drivers and the users.

Technician
He/She is responsible for the maintenance.

1.4 References

- IEEE Std 830-1998: "IEEE Recommended Practice for Software Requirements Specifications"
- Project description: "Software Engineering 2 Project, AA 2015/2016"

1.5 Overview

This document provides a detailed description of the system and it is structured in three main sections:

Introduction
This section gives a high level description of the main topics covered throughout the whole document, that is the system purpose and its scope.

Overall description
: This section offers the reader an overview over the general factors that affect the product and its requirements.

Specific requirements
: This offers an insight over every functional and non functional requirement.

2 Overall Description

2.1 Product Perspective

As stated in the *Scope* section the product we will release is composed by four main software applications.

- A **Web Application** addressed to the *users* to use our service. This application must interface mainly with the **Back-End Application** and with Google's Maps API.
- A **Mobile Application (*User*)** addressed to the *users* to use our service and available for Android, iOS, Windows Mobile and Blackberry. This application must interface mainly with our **Back-End Application** and with Google's Maps API
- A **Mobile Application (*Taxi Driver*)** addressed to the *taxi drivers* to use our service and available for Android, iOS. This application must interface mainly with our **Back-End Application** and with Google's Maps API
- A **Back-End Application** that will handle all the business logic and that must interface mainly with Google's Maps API and with a MongoDB database .

2.1.1 User Interfaces

2.1.2 Hardware Interfaces

Both the **Mobile Application (*User*)** and the **Back-End Application** must interface with the GPS module and with the Network module.

2.1.3 Software Interfaces

Web Application

- MyTaxyService API
  - Mnemonic : Back-end
- Google Maps API
  - Mnemonic : Google Maps API
  - Version Number : V3
  - Source : `https://developers.google.com/maps/documentation/javascript/`

Mobile Application (*User*) and Mobile Application (*Taxi Driver*)

- MyTaxyService API
  - Mnemonic : Back-end
- Google Maps API
  - Mnemonic : Google Maps API
  - Version Number : V3
  - Source : `https://developers.google.com/maps/`
- Android SDK
  - Mnemonic : Android
  - Version Number : 6.0
  - Source : `http://developer.android.com/sdk/index.html`
- iOS SDK
  - Mnemonic : iOS
  - Version Number : 9.2
  - Source : `https://developer.apple.com/ios/download/`
- Windows Mobile SDK
  - Mnemonic : Windows Mobile
  - Version Number : 6.5

  - Source : `http://www.microsoft.com/en-us/download/details.aspx?id=17284`

- BlackBerry SDK
  - Mnemonic : BlackBerry
  - Version Number : 10
  - Source : `https://developer.blackberry.com/`

Back-End Application

- Node.js API
  - Mnemonic : Node.js
  - Version : 4.2.1
  - Source : `https://nodejs.org/api/`
- MongoDB API
  - Mnemonic : MongoDB
  - Version : 3.0
  - Source : `https://docs.mongodb.org/manual/`
- Google Maps API
  - Mnemonic : Google Maps API
  - Version Number : V3
  - Source : `https://developers.google.com/maps/documentation/javascript/`
- Javascript API
  - Mnemonic : Javascript
  - Source : `https://developer.mozilla.org/en/docs/Web/API`

2.1.4 Communication Interfaces

Web Application
Every application must interface with the Internet network. This interface is handled by the operative systems and not by The applications themselves.

2.1.5 Memory Constrains

Web Application and Mobile Application (*User*) and Mobile Application (*Taxi Driver*) The Mobile and Web Applications can not exceed 75MB of RAM usage.

Back-End Application The Back-End application can not exceed 15GB of total RAM usage.

2.1.6 Site Adaptation Requirements

Software Adaptation
Every Mobile Application must be developed according to the platforms design guidelines.

2.2 Product Functions

This section highlights the main product functions sorted by application.

2.2.1 Web Application and Mobile Application (*User*)

[**F1**] The app must allow the user to book a taxi on the spot.

[**F2**] The app must allow the user to reserve a taxi ride up to 7 days before the chosen date.

[**F3**] The app must allow the user to share his ride with other people whose destination is included in the ride path.

2.2.2 Mobile Application (*Taxi Driver*)

[**F4**] The app must provide position updates to the *back-end application*

[**F5**] The app must allow the taxi driver to accept or refuse a ride request.

[**F6**] The app must allow the taxi driver to report issues that could delay his arrival (traffic jam, car faults, etc ...)

[**F7**] The app must allow the taxi driver to overview a history of his last rides.

[**F8**] The app must allow the taxi driver to report a absent user.

2.2.3 Back-End Application

[F9] The application must provide an interface for the registration process of users and taxi drivers

[F10] The application must handle the request queue using a fair policy [Rif needed]

[F11] The application must provide an accurate [rif needed] estimate of the total cost and time of a ride

[F12] The application must provide an interface to allow an operator to perform all basics CRUD operations

2.3 User Characteristics

This section identifies all the different users of the product.

User [rif needed] The user has to be at least 13 years old.
Has to be familiar with standard mobile applications interfaces.

Taxi Driver [rif needed] The driver has to be a regularly licensed taxi driver.
The driver has to be familiar with standard mobile applications interfaces and is required to own a compatible device [rif needed]

Maintenance Guy [rif needed] The maintenance guy must be able to debug and identify issues in a full stack environment. The maintenance guys must be familiar with the application requirements and the overall infrastructure and design.

Support Operator The support operator must be familiar with the overall application design and has to know how to resolve common user's issues.

2.4 Constraints

2.4.1 Regulatory Policies

- **Privacy Policy** Data should be collected and stored following the privacy policy guidelines provided by Mozzilla Foundation [rif needed] `https://developer.mozilla.org/en-US/Marketplace/Publishing/Policies_and_Guidelines/Privacy_policies`

- **Taxi Driver Policy** The client application for taxi drivers must be accessible only by registered and regularly licensed driver.[rif neeeded]

2.4.2 Hardware Limitations

Mobile Application (User) The Mobile Application (User) must be developed in order to be available for the 80% of the devices currently on the market for each different platform.

Web Application (User) The Web Application must be developed in order to be available for the 100% of devices that support the latest version [rif needed] of most used browser [rif needed]

2.4.3 Interfaces to other applications

Mobile Application (*User*) The developer of the **Mobile Application (*User*)** must follow the design guideline [rif needed] of the platform and has to interface with the respective Developer Console and make sure that the app will be accepted and published.
The application must interface with Google's Maps API [rif needed].

Web Application The application must interface with Google's Maps API [rif needed].

Mobile Application (*Taxi Driver*) The application must interface with Google's Maps API [rif needed].

Back-End Application The application must interface with Google's Maps API [rif needed].

2.4.4 Parallel Operations

Back-End Application The **Back-End Application** must handle parallel request from multiple users without generating conflicts or inconsistency. The system must decline a request from a user if it finds another active request from the same user.

2.4.5 Reliability Requirements

Back-End Application This requirement of reliability and availability refers to the total time the system is available for the applications of the end

users.
The system must reach a 99.99% uptime (corresponding to a inactivity time of 50 minutes/year).

2.4.6 Safety and Security

Mobile Application (*Taxi Driver*) The application must include an automatic night mode and gallery mode in order not to compromise the driver security [rif needed]

2.5 Assumptions And Dependencies

2.5.1 Domain Assumptions

We assume that the system will be deployed in a city of about 1.5M inhabitants with about 4800 regular licensed taxi drivers.
We assume that at launch time the end user **Mobile Application (*User*)** will reach an audience of about 100k users while the **Web Application** will reach an audience of about 10k users.
We assume that the city is dived in taxi zones.

2.5.2 Taxi Driver Assumptions

We assume that every taxi driver will have a compatible device (personal or provided by the project client) [rif needed] with a working GPS module.

2.5.3 User Assumptions

We assume that every user will have a device with an enabled Internet connection.
We also assume that the location provided by the user (manually or automatically via GPS) is always correct.