myTaxiService
Requirements Analysis and Specification Document

Angelo Gallarello, Giacomo Locci, Edoardo Longo

November 2, 2015

Contents

Abstract

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 Introduction

This section introduces the *system-to-be*, and provides a high level description of its functioning, together with the main functionalities that the system will offer.

1.1 Purpose

The system is a software named *myTaxiService.* It is a taxi service that will operate in a big city; the main purpose is to simplify the access of passengers to the service and to guarantee a fair management of the taxi queues.

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 *Support 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 *Support 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.

Support 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.

Taxi Zone
Area of the city of 2km^2 .

Admissible Paths
Two paths are admissible if the destination point of path is included in the other path and the starting points belong to the same taxi zone.

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 MySQL 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

- MySQL API
  - Mnemonic : MySQL
  - Version : 5.0
  - Source : `https://dev.mysql.com/doc/refman/5.0/en/c-api.html`
- Java EE API
  - Mnemonic : Java
  - Version : 6
  - Source : `http://docs.oracle.com/javaee/6/api/`
- Google Maps API
  - Mnemonic : Google Maps API
  - Version Number : V3
  - Source : `https://developers.google.com/maps/documentation/javascript/`

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.
Functions are here described in high level language to give a general overview of the products.
Detailed requirements are listed in section 3.2 on page 18

2.2.1 Web Application and Mobile Application (*User*)

[**F1**] The app must allow the user to register and login

[**F2**] The app must allow the user to book a taxi on the spot.

[**F3**] The app must allow the user to reserve a taxi ride

[**F4**] 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*)

[F5] The app must allow the taxi driver to login

[F6] The app must provide position updates to the *back-end application*

[F7] The app must allow the taxi driver handle ride requests

[F8] The app must allow the taxi driver to report issues that could delay his arrival (traffic jam, car faults, etc ...)

[F9] The app must allow the taxi driver to overview a history of his last rides.

[F10] The app must allow the taxi driver to report a absent user.

2.2.3 Back-End Application

[F11] The application must provide an interface for the registration process of users and taxi drivers

[F12] The application must provide an interface for the data modification process of the users

[F13] The application must implement a password recovery system

[F14] The application must handle the request queue using a fair policy

[F15] The application must provide an accurate estimate of the total cost and time of a ride

[F16] The application must provide an interface to allow an operator to perform all basics CRUD operations

2.3 User Characteristics

This section highlights the main characteristics of the *actors* defined in section 1.3 on page 6.

User The user has to be at least 14 years old and own a valid ID.
The user to be familiar with standard mobile applications interfaces.

Taxi Driver The driver has to be a regularly licensed taxi driver.
The driver has to be familiar with standard mobile applications interfaces.

Technician The technician must be able to debug and identify issues in a full stack environment. The technician 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 `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.

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 of most used browser [1]

[1] `http://www.w3schools.com/browsers/browsers_stats.asp`

2.4.3 Interfaces to other applications

Mobile Application (*User*) The developer of the **Mobile Application (*User*)** must follow the design guideline [2] [3] 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 [4]

Web Application The application must interface with Google's Maps API.

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

Back-End Application The application must interface with Google's Maps API.

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] `http://developer.android.com/design/index.html`

[3] `https://developer.apple.com/design/`

[4] See section 2.1.3 on page 9

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.

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 [5] (personal or provided by the project client) with a working GPS module.
We assume that our system has access to an internal database of valid taxi licenses of the city.

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.

2.5.4 Taxi Driver Assumptions

We assume that the taxi driver has to send his personal information (license number, name, surname, C.F) via email to the support address to get valid

[5] See section 2.4 on page 13

login credentials.
We assume that every taxi driver handles all his rides with our application.

3 Specific Rquirements

3.1 External interfaces requirements

This section will examine the requirements needed for the *interfaces* introduced in section 2.1 on page 8.

3.1.1 User Interfaces

Here is a listing of all the requirements concerning the User Interfaces. They are divided into the different applications described in section 1.2.2 on page 5.

Mobile Application (*User*)
The user interface of the Mobile Application (*User*) has the following requirements:

[**IR1**] When the *User* inputs his/her position, the system must output whether a taxi is available or not;

[**IR2**] When the *User* inputs a destination, the system must output whether it is possible to share a ride with another *User* or not;

[**IR3**] When the *User* has requested a reservation the system must output either a confirmation or a refusal;

[**IR4**] When the *User* has modified his/her personal information the system must output either a confirmation or a refusal.

Mobile Application (*Taxi Driver*)
The user interface of the Mobile Application (*Taxi Driver*) has the following requirements:

[IR5] When the *Taxi Driver* accepts a taxi request the system must output either a confirmation or a refusal;

[IR6] When the *Taxi Driver* logs in the system must show him/her the all the pending requests that he/she can accept;

[IR7] The system must offer a way to the *Taxi Driver* to refresh the list of pending requests that he/she can accept;

[IR8] When the *Taxi Driver* signals any kind of emergency the system must output either a confirmation or a refusal;

[IR9] When the *Taxi Driver* has modified his/her personal information the system must output either a confirmation or a refusal.

Back-End Application

The user interface of the Back-End Application has the following requirements:

[IR10] When the *Support Operator* issues a warning to a *Taxi Driver* the system output either a confirmation or a refusal;

[IR11] When the *Support Operator* issues a warning to a *User* the system output either a confirmation or a refusal.

3.1.2 Hardware Interfaces

Here is a list of the requirements for the Hardware Interfaces defined in section 2.1.2 on page 8:

[IR12] The system must be able to interface with the GPS module;

[IR13] The system must be able to interface with the network module.

3.1.3 Software Interfaces

Here is a list of the requirements for the Software Interfaces defined in section 2.1.3 on page 9:

[IR14]

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.2.1 Web Application

Requirements for the main functions specified in section 2.2.1 on page 11.

[FR1] The application must allow every unregistered person above 13 years old to register providing email, ID number, name, surname, date of birth.

[FR2] The application must allow only registered user to access the service

[FR3] The application must allow the user to choose his starting position

[FR4] The application must allow the user to select the destination position

[FR5] The application must check that only valid coordinates and addresses are selected by the user

[FR6] The application must allow the user to book a taxi only if there are not other reservations

[FR7] The application must allow the user to reserve a taxi for a chosen date and position

[FR8] The application must allow the user to active the sharing option to allow other users to share the ride

[FR9] The application must allow the user to find sharing rides available in his path.

[FR10] The application must allow the user to modify his personal information (except for the ID number)

3.2.2 Mobile Application (*User*)

Requirements for the main functions specified in section 2.2.1 on page 11.

[**FR11**] The application must allow every unregistered person above 13 years old to register providing email, ID number, name, surname, date of birth.

[**FR12**] The application must allow only registered user to access the service

[**FR13**] The application must allow the user to choose his starting position

[**FR14**] The application must allow the user to use the GPS interface to select the starting position

[**FR15**] The application must allow the user to select the destination position

[**FR16**] The application must check that only valid coordinates and addresses are selected by the user

[**FR17**] The application must allow the user to book a taxi only if there are not other reservations

[**FR18**] The application must allow the user to reserve a taxi for a chosen date and position

[**FR19**] The application must allow the user to active the sharing option to allow other users to share the ride

[**FR20**] The application must allow the user to find sharing rides available in his path.

[**FR21**] The application must allow the user to modify his personal information (except for the ID number)

3.2.3 Mobile Application (*Taxi Driver*)

Requirements for the main functions specified in section 2.2.2 on page 12.

[**FR22**] The application must allow only registered drivers [6] to access the service

[**FR23**] The application must send every 2 minutes a position update to the Back-End Application.

[6] See section 2.5.4 on page 15

[**FR24**] The application must allow the driver to accept a ride request in just one tap

[**FR25**] The application must allow the driver to refuse a ride request in just one tap

[**FR26**] The application must allow the driver to report most common issues using a drop-down

[**FR27**] The application must show a list of the last 30 rides, providing information about time, passenger and cost.

[**FR28**] The application must allow the driver to report an absent user selecting the appropriate ride request.

[**FR29**] The application must notify the driver when the sharing option is active

[**FR30**] The application must show the driver all the starting and arrival position when the sharing option is active

[**FR31**] The application must dim the screen light during late hours and when a gallery is detected by the GPS

3.2.4 Back-End Application

Requirements for the main functions specified in section 2.2.3 on page 12.

[**FR32**] The application must accept registration request from the Mobile Application (*User*) and the Web Application and create a new user

[**FR33**] The application must provide a control panel to allow the operator to register a new taxi driver

[**FR34**] The control panel must be accessible only with a secret password

[**FR35**] The application must generate automatically login credentials for new taxi driver

[**FR36**] The application must record every ride and store all the data in the database

[**FR37**] The application must handle one queue for every city zone

[**FR38**] The application must assign a driver to a queue if and only if the driver is in the zone of the corresponding queue

[**FR39**] The application must sort the queue in a descending order using the driver idle time as ordering parameter

[**FR40**] The application must calculate an estimated time and cost for every ride request using Google's Maps API

[**FR41**] The application must allow two users to share a taxi ride only if their path are admissible[7]

[**FR42**] The application must update the database after a user request some data modification

[**FR43**] The application must send a password reset email when the user signals that he has lost his password

[**FR44**] The application must ban a user if he cancel a request that has been already processed

[**FR45**] The application must ban a user if the taxi driver report the user as absent

[**FR46**] The application must accept a ride request from a user only if the user is not banned and if the user has not other pending request

[**FR47**] The application must be developed in order to be extensible [8]

[7] See section 1.3 on page 7

[8] The application developer should follows these guidelines `https://docs.oracle.com/javase/tutorial/ext/basics/spi.html`