myTaxiService

-

Requirements Analysis and Specification Document

Davide Cremona, Simone Deola

October 28, 2015

Contents

Chapter 1

Introduction

1.1 Purpose

This document is the R.A.S.D. (Requirement Analysis and Specification Document). The purpose of this document is the description of the "myTaxiService" system. At first, it will provide functional and non-functional requirements, a complete overview of the constraints of the system and its limits. Then it will explain in detail the dynamics of the system using real-life use cases. Finally this document will provide a base for the developers that concretely have to implement the system.

1.2 Actual System

The functionality that the new system will provide is now not supported. So the entire system must be developed without using or modifying existing system.

1.3 Scope

The objective of myTaxiService is to provide an interface between customers and taxi drivers to optimize their interaction and provide a fair management of taxi queues. The users, once registered through the mobile application or the web application, can request a taxi for their travel or reserve one, specifying the origin and the destination. The reservation can be done at least two hour before the ride; if the reservation can take place, the system will allocate a taxi 10 minutes before the meeting time. On the other side, taxi drivers can inform the system that they are waiting for a client and accept or decline a ride request. If the request has been accepted, a notification will be sent to the requesting customer with the identification number of the incoming taxi and the time he has to wait. Otherwise, if the request has been rejected it will be forwarded to the next taxi in the queue. The system has to optimize the management of customers requests giving the rides

to the taxi with the highest priority that has to be evaluated in function of avaiability and the nearness of the taxi driver.

1.4   Actors

- **Guest User:** guest users are unlogged or unregistered users. They can visit the login page or the registration forms.

- **Registered User:** this kind of user identify either a Guest User or a Taxi Driver.

- **Customer:** this kind of user is the end-user of the service. He can perform request for taxis or reserve a ride. In his personal page he can view his requests and the system responses.

- **Taxi Driver:** this kind of user is composed by the actual taxi drivers that can only see customers requests that has been forwarded by the system. He can accept or decline these requests. Also, he's considered a special kind of user because one can register as a "Taxi Driver" only if he provide a valid Taxi licence.

1.5   Goals

- **[G1]** Allow guest user to become a customer or a taxi driver.

- **[G2]** Allow registered user to log in.

- **[G3]** Allow customers to require a taxi.

- **[G4]** Allow customers to reserve a ride.

- **[G5]** Allow customers to delete a previous reservations.

- **[G6]** Allow taxi drivers to accept or decline a ride request.

- **[G7]** Allow taxi drivers signal a customer if it made a bad use of the system.

- **[G8]** Allow taxi drivers to notify their availability.

- **[G9]** After login, the system will notify the customer that his request has been accepted.

- **[G10]** After login, the system will notify the hyperref[sec:tdriver]taxi driver about the incoming requests.

1.6 Definitions, Acronyms, Abbreviations

1.6.1 Definitions

1.6.2 Actonyms

1.6.3 Abbreviations

- RASD: Requirement Analysis and Specification Documents.

- DD: Design Document.

1.7 Reference documents

- (IEE830) IEEE Recommended Practice for Software Requirements Specifications

1.8 Document overview.

Until now, we have given a general explanation about the software functionalities and a brief description about this document. Now we will describe what the rest of this RASD contains.
In Section 2 we will focus more about system constraints and assumptions.
In Section 3 we will describe requirements, typical scenarios and use-cases. In this section there is also a collection of UML diagrams that describes in particular the functionalities of the system.
//TODO SECTION 4

Chapter 2

Overall Description

2.1 Product perspective

The system will be composed of a web application and a mobile application developed for the three major OS (Apple iOS, Android, Windows 10). The system will provide some API with the purpose of a future connection with another travel planning systems.

2.2 User Characteristics

The users that we suppose will use our system are of two types. the ones who want to find a taxi for a travel in the simplest way (customers). The others are taxi drivers that want to increment their productivity. The first ones must be able to access to a web browser or download and using a mobile application, the second ones also must have a taxi license.

2.3 Constrains

2.3.1 Regulatory Policies

myTaxiService has to meet regulatory policies about taxies in the countries where it will be used.

2.3.2 Hardware Limitations

The only hardware limitation that the myTaxiService mobile application has to meet will be the mobile phones characteristics. the rest of the system will be no affected by particular hardware limitations.

2.3.3 Software Limitations

myTaxiService mobile application has to be compatible with all major mobile operating systems (Android, Apple iOS, Windows 10). Also myTaxiService web application has to be compatible with all major browser (Chrome, Safari, Firefox, Microsoft Edge).

2.3.4 Parallel Operations

Our system must be able to perform parallel operations on the database to satisfy all the requests from multiple users.

2.3.5 Documents Related

- Requirements and Analysis Specification Document (RASD)

- Design Document (DD)

2.4 Assumptions

- Every taxi driver has equipped a smartphone during working hours.

- Every taxi driver has a unique taxi license.

- Every taxi has a GPS locator to send GPS information to the central server.

- Android, Apple iOS or Windows 10 is avaiable on the registered users smartphones.

- Every registered users can be connected to the Internet with a mobile device when outside.

- When a customer require a taxi, the GPS informations about his location are automatically sended to the central server.

- The reservation of a ride is made at least two hours before the ride.

- Deletion of a reservation is made at least two hours before the ride.

- Requests from customers are automatically notified to the first taxi driver in the zone queue.

- If a taxi driver declines a request he will be placed in the bottom of the zone queue.

- If a request is declined it will be forwarded to the next taxi driver in the zone queue.

- If a customer make a bad use of the taxi request system, he can be reported as a bad customer.

- If a taxi driver notifies his availability is because he is actually avaiable

- If a taxi driver notifies his availability is because he wants to be notified of customers that needs a ride.

- If a taxi driver accept a request, the requesting customer will be notified

2.5 Future possible Implementations

A possible future implementation can be a complex feedback system that permits to the customers to leave a comment about the taxi driver and vice versa. For example taxi drivers can be interested in knowing the punctuality or how is the behave of the customer that requests the ride.

Chapter 3

Specific Requirements

This chapter contains a detailed description of how the applications works and its features. It also gives a specification of the functional and quality requirements.

3.1 External Interface Requirements

This section gives a description of the various inputs and relative outputs of the system. It also gives a description of the hardware, software and communication interfaces that are necessary to make the system work. It will also provide a generic visualization of the user interface in the various user platforms.

3.1.1 User Interfaces

//TODO : Add a description of the generic mobile application and of the web application

3.1.2 Hardware Interfaces

Since mobile and web applications don't have any dedicated hardware we have not designed any hardware interfaces for our system. The interaction with the central database is performed by connections handled by the already installed operating system on the mobile devices or the users computers.

3.1.3 Software Interfaces

- The mobile application communicates with the GPS application in order to get geographical informations about the user.

- The web application communicates with the browser in order to get geographical informations about the user.

- Mobile and web applications communicates with the database through HTTP requests to the server.

3.1.4 Interfaces to Others Application

- myTaxiService web application require that at least one of these browsers is installed on the user Personal Computer:

Table 3.1: Browsers

Name	Version	Company	Source
Safari	9.0.1	Apple Inc.	Get Safari
Firefox	41.0	Mozilla	Get Firefox
Chrome	46.0.2490	Google	Get Chrome
Microsofr Edge	20.10240.16384.0	Microsoft	Get Edge

- myTaxiService mobile application require that at least one of these operating systems is installed on the user Smartphone:

Table 3.2: Mobile Operative Systems

Name	Version	Company	Source
Android	KitKat 4.4W.2 or later	Google	Android Info
iOS	9.1 or later	Apple Inc.	iOS Info
Windows 10	10.0.10572.0 or later	Microsoft	Windows 10 Info

- To give an additional LogIn method, we use also the "LogIn with Facebook" API relased by Facebook. Facebook Login for Apps is a fast and convenient way for people to create accounts and log into our system across multiple platforms. It is well described at Facebook Login API Page.

3.1.5 Communication Interfaces

The communication between system pieces is not specified because it is handled by the underlying operating systems for both the mobile application and the web portal.

In particular, the web and mobile applictaions will communicate with the server through HTTP/HTTPS requests.

- HTTP communicate through the port number 80 and is handled by the operating system.

- HTTPS communicate through the port number 443 and is handled by the operating system.

3.2 Functional Requirements

3.2.1 Functional Requirements for Guest Users

3.2.2 Functional Requirements for Customers

3.2.3 Functional Requirements for Taxi Drivers

3.3 Scenarios

3.4 UML Models

3.4.1 Use-Case Diagrams

3.4.2 Class Diagrams

3.4.3 State Machine Diagrams

3.5 Non Functional Requirements

3.5.1 Performance Requirements

3.5.2 Design Constraints

3.5.3 Software System Attributes

Avaiability

Maintainability

Portability

3.5.4 Security

External Interface Side

Application Side

Server Side

Chapter 4

Appendix

//TODO