Requirements Analysis and Specification Document: myTaxiService

Chitti Eleonora, De Nicolao Pietro, Delbono Alex

October 23, 2015

Contents

Chapter 1

Introduction

1.1 Purpose

This document is the Requirement Analysis and Specification Document for the myTaxiService application. Its aim is to completely describe the system, its components, functional and non-functional requirements, constraints, and relationships with the external world and to provide typical use cases and scenarios for all the users involved. Further, this document will provide formal specification of some features of the applications.

This document is written for project managers, developers, testers and Quality Assurance. It may be useful also to users. It may be used in a contractual requirement.

1.2 Scope

The system is a taxi reservation and dispatching system for large cities. Its aim is to simplify the access of passengers to the service and to guarantee a fair management of taxi queues.

The system consists in a back-end server application (*myTaxi Server*), a web application front-end (*myTaxi Web*) and in a mobile application (*myTaxi Mobile*).

The system has 2 types of users: passengers and taxi drivers; it should allow the users to sign up and login with their credentials. The system has to know the location of both the passengers and the taxi drivers.

The system allows any passenger to request a taxi, informing him o her of the incoming taxi code and the estimated waiting time.

The system knows about the available taxi drivers and, when a request is incoming, informs one of them about the location of the available passenger; the taxi driver can either accept or deny the ride. If the taxi driver accepts the ride, the system sends a confirmation to the passenger, together with the estimated waiting time. If the taxi driver rejects the ride, the system looks for another taxi driver in the same way.

The system offers programmatic interfaces (APIs) to enable the development of additional services on top of the basic one.

The system is provided with two optional modules:

Taxi reservation allows the passenger to reserve a taxi by specifying the origin and the destination of the ride.

Taxi sharing allows the passengers to share a ride together, dividing the costs.

1.3 Definitions, acronyms, and abbreviations

RASD: Requirements Analysis and Specification Document.

System: the whole software system to be developed, comprensive of all its parts and modules.

Module: an optional software component which uses the core system APIs to provide additional features.

Passenger: the registered user who uses the service for a taxi ride.

Taxi driver: any taxi driver subscribed to the service.

1.4 References

- Project rules: "AA 2015-2016 Software Engineering 2 - Project goal, schedule and rules"
- Assignment: "Software Engineering 2 Project, AA 2015/2016 - Assignments 1 and 2"

- IEEE Std 830-1998: "IEEE Recommended Practice for Software Requirements Specifications"

1.5 Overview

This document is structured in three parts:

Chapter 1: Introduction. It provides an overall description of the system scope and purpose, together with some information on this document.

Chapter 2: Overall description. Provides a broad perspective over the principal system features, constraints, and assumptions about the users and the environment.

Chapter 3: Specific requirements. Goes into detail about functional and nonfunctional requirements. This chapter is arranged by feature.

Chapter 2

Overall description

2.1 Product perspective

The back-end stores its data in a RDBMS and can run on every platform that supports the JVM. The web applications works on any web server that supports PHP. The mobile application is supported by Android, iOS. The system provides APIs to extend its functionalities, e.g. :

- taxi reservation
- ride sharing
- online payments
- ...

2.1.1 User interfaces

2.1.2 Hardware interfaces

2.1.3 Software interfaces

2.1.4 Communications interfaces

2.2 Product functions

The system allows passengers to book a taxi and notify taxi drivers of the request.

In particular it lets users to:

- Passengers:
  - create an account
  - request a taxi
  - share a taxi with other passengers
- Taxi drivers:
  - create an account
  - accept or deny a lift request

2.3 User characteristics

The two kinds of users are passengers and taxi drivers. We suppose that both kinds of users have access to Internet.

Taxi drivers must be able to install and use the mobile application on their cellphone to answer the ride requests.

Passengers have to use the browser application or the mobile one.

2.4 Constraints

2.4.1 Regulatory policies

It's user responsibility to ensure that the use of the system complies with the local laws and policies.

The system must ask the user for the permission to acquire, store and process personal data and web cookies. The system must offer to the user the possibility to delete all the personal data.

2.4.2 Hardware limitations

the system has to run under the following conditions:

- App
  - 3G connection, at 2 Mb/s

  - 100 MB of free space
  - 2 GB of RAM

- Web application
  - Support for current version of IE, Firefox, Chrome, Safari, Opera as of 2013
  - 2 Mb/s Internet connections
  - 800x600 resolutions

2.4.3 Interfaces to other applications

2.4.4 Reliability requirements

The system must have a minimum availability of 98%.

2.4.5 Criticality of the application

The system is not employed in life-critical applications.

2.4.6 Safety and security considerations

The locations of the passenger and its destinations must be kept private unless the passenger chooses to share rides.

Only taxi drivers with a valid license must be able to use the service for security reasons.

2.5 Assumptions and dependencies

We assume that:

- the taxi is provided with a GPS navigator
- the taxi driver is able to reach the meeting point within 10 minutes from the given hour 90% of the times
- the taxi driver is able to reach the meeting point within 20 minutes from the given hour 100% of the times

- the passenger waits in the location until the taxi arrives
- the taxi driver picks up the correct passenger

2.6 Future extensions

Chapter 3

Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

3.2 System features

3.2.1 User registration

3.2.2 User login

3.2.3 Standard taxi call

3.2.4 Ride request notification to the taxi driver

3.2.5 Taxi availability handling

3.2.6 Taxi reservation

3.2.7 Ride sharing

3.2.8 User profile management

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes