POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Computer Science and Engineering

Project of Software Engineering 2: MyTaxi Service Requirements Analysis and Specification Document

Docente del corso: Prof. Elisabetta DI NITTO

Autori:

Mattia CRIPPA matricola

Francesca GALLUZZI matricol

Marco LATTARULO matricola

Anno Accademico 2015–2016

Contents

List of Figures

Chapter 1

Introduction

1.1 Purpose

This document represent the Requirement Analysis and Specification Document (RASD). The main goal of this document is to completely describe the system in terms of functional and non functional requirements, analyse the real need of the customer to modelling the system, show the constraints and the limits of the software and simulate the typical use cases that will occur after the development. This document is intended to all developers and programmers who have to implement the requirements, to system analysts who want to integrate other systems with this one, and could be used as a contractual basis between the customer and the developer.

1.2 Description of the Problem

The aim of the project is to optimize a taxi service in order to simplify the access of passengers to the service and guarantee a fair management of taxi queues. Passengers can request a taxi either through a web application or a mobile app. The system answers to the request by informing the passenger about the code of the incoming taxi and the waiting time. Taxi drivers use a mobile application to inform the system about their availability and to confirm that they are going to take care of a certain call. The system guarantees a fair management of taxi queues. In particular, the city is divided in taxi zones (approximately 2 km2 each). Each zone is associated to a queue of taxis.

The system automatically computes the distribution of taxis in the various zones based on the GPS information it receives from each taxi. When a taxi is available, its identifier is stored in the queue of taxis in the corresponding zone. When a request arrives from a certain zone, the system forwards it to the first taxi queuing in that zone. If the taxi confirms, then the system will send a confirmation to the passenger. If not, then the system will forward the request to the second in the queue and will, at the same time, move the first taxi in the last position in the queue. Besides the specific user interfaces for passengers and taxi drivers, the system offers also programmatic interfaces to enable the development of additional services (e.g. taxi sharing) on top of the basic one.

A user can reserve a taxi by specifying the origin and the destination of the ride. The reservation has to occur at least two hours before the ride. In this case, the system confirms the reservation to the user and allocates a taxi to the request 10 minutes before the meeting time with the user.

A user can enable the taxi sharing option. This means that he/she is ready to share a taxi with others if possible, thus sharing the cost of the ride. In this case the user is required to specify the destination of all rides which he/she wants to share with others. If others are willing to start a shared ride from the same zone going in the same direction, then the system arranges the route for the taxi driver, defines the fee for all people sharing the taxi and informs the passengers and the taxi driver.

1.3 Goals

Passengers should be able to:

- Login
- Sign up
- Receive receipts after each completed ride
- Request a taxi

- Receive informations about the incoming taxi (confirmation, taxi code, waiting time)
- Reserve a taxi
- Share the ride

Taxi drivers should be able to:

- Login
- Sign up
- Signal whether they are available or not
- Receive incoming requests
- Accept incoming requests
- Visualize information about the optimal route

Developers should be able to:

- Access a specific interface of the system to add new features

1.4 Domain Properties

We suppose that the following properties hold in the analysed domain:

- Accurate taxicabs positions are known by the GPS
- Taxi drivers correctly signal their availability
- If a passenger requests or reserves a taxi, he will then take the ride from the specified origin to the specified destination
- A passenger doesn't change the origin or the destination of a ride after the reservation
- If a taxi driver confirms a ride, then he will reach the passenger location on time according to the reservation hour or the waiting time calculated by the system
- If a taxi driver confirms a ride, he will complete it

1.5 Assumptions

A1: Given the collected data about the distribution and the total number of the taxicabs, we assume that the coverage of the taxicabs among the zones is almost uniform; therefore the probability of having an empty queue is reasonably low.

A2: Given the collected data about the acceptance rate of the taxi drivers, we assume that the probability of reaching the end of the queue without any acceptance is reasonably low.

A3: We assume that the maximum number of people sharing the same ride is 3.

A4: We assume that the taxi driver will correctly follow the optimal route suggested by the system

1.6 Definitions, acronyms and abbreviations

PASSENGER: is a citizen that benefits of the taxi service

TAXI DRIVER: is the owner of a taxicab, and provides a taxi service to citizens

RIDE: is the act of make use of the taxi service (provided by a taxi driver) by a passenger

REQUEST: is the act performed by a passenger when he immediately needs a taxicab in the location in which he is

RESERVATION: is the act performed by a passenger when he needs a taxicab in a certain future time (at maximum in 2 hours from now) at a certain specific location

ROUTE: is the path through the city from the origin of the specific ride to the destination of the specific ride

1.7 Proposed System

The system will be composed by a server running all the business logic, generating dynamic web pages and managing all the accesses to the data sources, and by a number of clients implemented as mobile application deployed on Android or a web application using the JEE platform. GPS raw data will be provided by a specific GPS receiver, installed in every taxicab and provided by a specialized company.

1.8 Stakeholders Identification

- The government of the city in which the system will be used
- The citizens of the city in which the system will be used
- The taxi drivers of the city in which the system will be used
- A specific IT company which provides all the technology (GPS receivers, smartphones for taxi drivers, mainframes and computational power) in exchange of raw GPS data provided by the receiver used for data mining purposes

1.9 Document Overview

This document is essentially structured in eight parts:

Section 1 - Introduction and Overall Description: it gives a description of document and gives general information about the software product with more focus about constraints and assumptions.

Section 2 - Actors Identification: it gives a description of all actors of the system

Section 3 - Specific Requirements: this part lists all the functional and non functional requirements that are part of the system and use cases.

Section 4 - Scenarios Identification: it gives a description of typical scenarios

Section 5 - UML Models: to give an easy way to understand all functionality of this software, this section is filled with UML diagrams.

Section 6 - Alloy Modeling: contains the code used for the analysis of consistency of the Class Diagram

Section 7 - Words Generated: contains words generated by Alloy Analyzer in order to understand that the model is consistent

Section 8 - Used Tools: this part contains some information about the softwares used to realize this document

1.10 Reference Documents

- Specification Document: MyTaxiService Project AA 2015-2016.pdf.
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- IEEE Std 1016tm-2009 Standard for Information Technology - System Design - Software Design Descriptions.

Chapter 2

Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 User Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory

2.1.7 Operations

2.1.8 Site Adaptation Requirements

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements

Chapter 3

Specific Requirements

3.1 External Interfaces

3.2 Functions

3.3 Performance Requirements

3.4 Logical Database Requirements

3.5 Design Constraints

3.5.1 Standards Compliance

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements

3.7.1 System mode

3.7.2 User class

Chapter 4

Supporting Information

4.1 Appendixes