POLYTECHNIC UNIVERSITY OF MILAN

School of Industrial and Information Engineering

Computer Science and Engineering

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

Course Professor: Prof. Elisabetta DI NITTO

Authors:
Mattia CRIPPA 000000
Francesca GALLUZZI 000000
Marco LATTARULO 000000

Academic Year 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, analyze the real need of the customer to modeling 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 Scope

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 km^2 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 Domain Properties

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

D01 : Accurate taxicabs positions are known by the GPS

D02 : Taxi drivers correctly signal their availability

D03 : If a passenger requests or reserves a taxi, he will then take the ride from the specified origin to the specified destination

D04 : A passenger doesn't change the origin or the destination of a ride after the reservation

D05 : 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

D06 : If a taxi driver confirms a ride, he will complete it

1.4 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.

1.5 Actors

The actors of our system are:

- ***Visitor***: unregistered user that access the application interface in order to sign up or Log In as passenger, taxi driver or developer and start interacting with the system.
- ***Passenger***: this user, after successful Log In, depending on his/her needs, access the request interface or the reservation interface. He/she has also access to an information page about all his/her accepted reservations or requests and has access to a page with the chronology of his/her receipts.
- ***Taxi driver***: this user, after successful Log In, is enabled to receive reservations or requests from passengers and can decide whether to accept them or not. He/she can access at any time after the acceptance a summary page about the specific ride. This user is also able to signal at any moment his/her availability to accept new reservations or requests.

- ***Developer***: this user, after successful Log In, can access a specific interface through which he/she can introduce modifications to the system or access privileged information about the system for maintenance scopes.

1.6 Goals

Visitors should be able to:

G01 : Sign up

G02 : Log in

Passengers should be able to:

G03 : Request a taxi

G04 : Receive information about the incoming taxi (confirmation, taxi code, waiting time)

G05 : Reserve a taxi

G06 : Share the ride

G07 : Receive receipts after each completed ride

Taxi drivers should be able to:

G08 : Signal whether they are available or not

G09 : Receive incoming requests

G10 : Accept or decline incoming requests

G11 : Visualize information about the optimal route

Developers should be able to:

G12 : Access a specific interface of the system

G13 : Add new features to the system

1.7 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

1.8 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

1.9 References

- 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.

1.10 Overview

This document is essentially structured in eight parts:

- **Section 1** → Introduction: it gives a description of document and gives general information about the software product.

- **Section 2** → Overall Description: it gives a description of the software product with more focus about constraints and assumptions.

- **Section 3** → Specific Requirements: this part lists all the functional and non functional requirements that are part of the system, typical scenarios and use cases. To give an easy way to understand all functionality of this software, this section is filled with UML diagrams.

- **Section 4** → Supporting Information: this part contains some information about the attached .als file and some described screen-shot of software used to generate it.

Chapter 2

Overall Description

2.1 Product Perspective

The product is supposed to be an application with a web version and a mobile ones, that simplify the access of the users to the service and is integrated with an other existing system which guarantee a fairy management of the taxi queue. That application will have an internal interface for developers to enable the possibility of adding features.

2.2 User Characteristics

It is considered that the user do have the basic knowledge of operating the internet and to have access to it.

2.3 Constraints

2.3.1 Regulatory Policies

The system should only collects and process raw location information for the scopes related to the specific service provided (creating routes, monitor the distribution of the taxicabs, estimate arriving times, calculating fees). All the personal information should be automatically managed by the system and not shown to any developer or system administrator. All data about location should be aggregated by the system only for statistical purposes. Any other relevant data needed for the optimal behavior of the entire system should be collected, used and managed automatically by the system itself according to all the existing privacy laws and regulations.

2.3.2 Hardware Limitations

The application should be able to receive data from the GPS receiver using its specific communication protocol via Bluetooth interface.

2.4 Assumptions and Dependencies

A01 : 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.

A02 : 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.

A03 : We assume that the maximum number of people sharing the same ride is 3.

A04 : We assume that the taxi driver will correctly follow the optimal route suggested by the system.

Chapter 3

Specific Requirements

3.1 Functional Requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs.

G1 → To allow a visitor to sign up, the system has to:

- check that the personal information and email provided by the visitor when filling in the registration form does not correspond to any other existing user
- check that the username chosen by the visitor has not been already taken
- provide a sign up functionality including a registration form

G2 → To allow a visitor to login, the system has to:

- check if username and password provided by the visitor correspond to an existing user
- provide a login functionality

G3 → To allow a passenger to request a taxi, the system has to:

–

G4 → To allow a passenger to receive information about the incoming taxi the system has to:

- check the position of the taxi driver, calculate the expected waiting time and show this information to the passenger

G5 → To allow a passenger to reserve a taxi, the system has to:

- provide a reservation functionality
- check if the information provided by the passenger (origin, destination, time) are consistent and complete (locations do exist inside the area in which the service is allocated and the time specified does not exceeds the limit of two hours)

G6 → To allow a passenger to share a ride, the system has to:

- provide a sharing functionality

G7 → To allow a passenger to receive receipts after each completed ride, the system has to:

–

G8 → To allow a taxi driver to signal whether he is available or not, the system has to:

- provide an available functionality through with the taxi driver could be put in the taxi queue

G9 → To allow a taxi driver to receive incoming requests, the system has to:

–

G10 → To allow a taxi driver to accept incoming requests, the system has to:

- provide an accept/decline functionality through with the taxi driver is able to accept or decline the incoming request

G11 → To allow a taxi driver to visualize information about the optimal route, the system has to:

- check the position of the passenger, check the destination, calculate the optimal route between the origin and the destination and show this information to the taxi driver

G12 → To allow a developer to access a specific interface of the system, the system has to:

-

G13 → To allow a developer to add new features, the system has to:

-

3.2 External Interfaces

Hardware Interface:
Device should be enabled with Internet and GPS receiver.
Software interface:
The user's browser should be HTML5 compatible and the resolution should be at least 1280x720 for a satisfactory user experience.

Here are presented some mockup that represent an idea of the structure of the application pages:

3.3 Performance Requirements

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays. We assume the response time of the system is close to zero, so the performance are essentially bounded by users internet connection.

3.4 Software System Attributes

3.4.1 Reliability

3.4.2 Availability

If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

3.4.3 Security

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. Information transmission should be securely transmitted without any changes in information.

3.4.4 Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

3.5 Scenarios Identification

Here are some possible scenarios for the application:

1. Visitor V wants to sign up into the system
2. Passenger A requires ride R and first taxi driver D of the queue accepts
3. Passenger A reserves ride R at time T. The first taxi driver D1 of the queue refuses the request but second taxi driver D2 of the queue accepts
4. Passenger A1 requires ride R enabling sharing option and another passenger A2 willing to share a ride is added by the system to the same ride R. The first taxi driver D of the queue accepts the ride.
5. After a ride, a passenger wants to see the receipt

6. Passenger A1 requires a ride R and the queue is empty. The system starts exploring the queues of all the 8 adjacent zones starting from the one positioned right above and proceeding clockwise. The first taxi driver D of the first explored zone, accepts the request.

3.6 UML Models

Chapter 4

Supporting Information

4.1 Appendixes