myTaxiService
Requirement Analysis and Specification Document
Politecnico di Milano

Cattaneo Michela Gaia, matr. 791685
Barlocco Mattia, matr. 792735

November 6, 2015

Contents

1 Introduction

1.1 Purpose

This document represent the Requirement Analysis and Specification Document (RASD). This document aims at describing the system in terms of functional and non-functional requirements, modeling the system according to the real needs of the customer and showing the constraints and the limit of the software. This document serves as a contractual basis between the customer and the developer.

1.2 Actual system

The government of a large city has commissioned a system that is able to manage and optimize the taxi service.

1.3 Scope

The system aims at simplifying the passengers access and guaranteeing a fair management of taxi queues. In order to manage the queue optimization, the city is divided in zones, each one associated to a taxi queue, and the system computes the taxi distribution basing on their GPSs information.

The system allows the user of the application to request a taxi in an area of the city, even without logging in. As soon as a taxi driver takes the request, the application notifies the user of the ID of the incoming taxi and the waiting time. In addition, the user can register, by filling in a form, and make reservations: he only needs to indicate the origin and destination of the ride and the exact time (at least 2 hours in advance) and the system reports that a taxi will be there to pick him up.

On the other hand, the taxi driver always has to notify his availability and let the system know if he is going to take a certain request by a passenger. If the driver is not able to confirm, the system will forward the request to the second in queue and move this taxi driver in the last position of the queue.

1.4 Actors

The actors participating in the system are:

- Visitor: a person who can see the home page of the application, he can register, look up for information about the system and request a taxi, after giving name and phone number.

- Passenger: a regular registered user who wants to use the application in order to request a taxi in a specific zone or make a reservation for a given time.

- Taxi Driver: a registered user who logged in with a specific ID. This area is reserved to taxi drivers who needs to use the application in order to inform the system of their availability or about what request they are going to take care of.

1.5 Goals

A user should be able to:

- Request a taxi.
- Payment method.
- Check the waiting time.
- Visualize the information about the system.
- Sign up into the system.

A passenger should be able to:

- Reserve a taxi by specifying the origin and the destination of the ride, at least two hour in advance.

The taxi driver should be logged in with a specific ID and should be able to:

- Change his state from available to unavailable and vice versa.
- Visualize the queue of his area.
- Answer positively or negatively to the request that the system forwards to him.
- Visualize the position of the user he needs to reach.

The system should be able to:

- Assign each taxi to the queue of the designated area, based on GPS position.
- Forward the request to the first taxi in the queue.
- If the first taxi of the queue doesn't take the request, the system will send the request to the second taxi of the queue and, at the same time the first taxi will be moved in the last position of the queue.
- If the passenger wants to reserve a taxi, the system will forward the request to the first available taxi in the area ten minutes before the ride.

1.6 Domain properties

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

- We assume that there are not previous implementations managing this service.
- The visitor is expected to be where he made the request to the system when the taxi driver arrives.
- The passenger is expected to be where and when he made the request to the system when the taxi driver arrives.

- The passenger is not supposed to cancel the reservation less than 2 hours earlier he has submitted it.

- A taxi driver always inform the system that he is available only when he actually is.

- A taxi driver, who notifies that he is going to take a certain request, always carries it out.

- When a taxi driver finishes a ride, he instantly changes his state from unavailable to available.

- There is always an available taxi driver in an area.

- If a user requests a service from the system, it will be within the borders of the city.

2 Overall Description

2.1 Product perspective

The product consists in a web application and a mobile app both based on the web. There is a server part which manages all the requests of the users along with the answers and the availability of the taxi drivers, computing their zone distribution. There are several clients (visitors, passengers or taxi drivers) interacting with the server by using a graphical user interface, but there is no implementation of interfaces for administration, in fact the application is only user-based.

2.2 User characteristics

The user can be a person who wants to request a taxi or access the application in order to use additional functionalities, such as make a reservation or easily request a taxi without having to give his information every time he exploits the service. The application is used also by the taxi drivers who needs to receive the requests of the customers and to interact with the system, responding to the calls and updating their availability and their GPS position . The user must have access to the Internet.

2.3 Constraints

2.3.1 Regulatory policies

The system must ensure the privacy of the users, without publishing their personal information.

2.3.2 Hardware limitations

The system doesn't work if the smartphone is not powerful enough or it has not the GPS detector.

2.3.3 Interfaces to other applications

The system interfaces with two applications:

- Google maps to determine the area of a taxi or the position of a user.

- Facebook if a user wants to sign in with his Facebook account.

2.3.4 Parallel operations

The system supports parallel operations of different clients accessing the application and making requests or reservations of taxis.

2.3.5 Documents related

- Requirements and Analysis Specification Document (RASD).

- Design Document (DD).

- User's Manual.

- Testing report.

2.4   Assumptions

- User can only make one request or reservation at a time.

- The system changes the state of a taxi driver from available to unavailable if he responds positively to a request, but not vice versa.

- If the passenger makes a reservation less then two hours in advance the system will not accept it.

- The system is able to manage the taxi queues in a way that there is always a taxi driver available for the reservation within ten minutes before the ride.

- The taxi driver has two minutes to respond to the request.

- The profile of the passengers are private, no other users can visualize it.

- The taxi driver receives the phone number and the position of the user if he take the request.

2.5   Future possible implementations

The application must be open to future implementations. Future possible implementations are:

- Taxi sharing: this is an additional functionality that allows different passengers to share the taxi with other passengers. The system must be notified that the passenger wants to use the taxi sharing option and it will ask him to indicate the destination of this rides. When it is all settled, the system is able to check if there are other persons in the same area that wants to reach the same destination, informing the taxi driver and the passengers. This can be a convenient solution, as the cost of the ride would be equally shared among the passengers.

- Price transparency: this is an additional functionality that allows the passenger to calculate the price of the ride in advance, inserting its origin and destination.

- Meet the driver: this is an additional functionality that lets the passenger know which taxi driver has answered his request. The system will send the passenger the profile of the designated driver with his name and data, so that the he would be able to recognize him or even call or text him if necessary.

- Give us feedback: this is an additional functionality that allows the passenger to rate the driver after the ride and leave comments about it.

- Easy to pay: this is an additional functionality that allows the passenger to automatically pay with the credit card associated with his account. The system will send him an e-mail with the receipt of payment.

3 Specific Requirements

3.1 External interfaces requirements

3.1.1 User interfaces

3.1.2 API interfaces

3.1.3 Hardware interfaces

This project does not support any hardware interfaces.

3.1.4 Software interfaces

3.1.5 Communication interfaces

3.2 Functional requirements

1. **Registration of the visitor.** The system displays the registration form to the user, who needs to fill in the mandatory fields of the form and press the "Submit" button. The user should provide his name, surname, password, e-mail address and telephone number. He can choose whether let the system know about his gender and date of birth or not.

2. **E-mail confirmation.** The system does not instantly confirm the creation of a new passenger profile, a confirmation e-mail is sent to the e-mail address provided by the user. The user can follow the link written in the e-mail and the system creates the new account.

3. **Look up information about the system.** The system displays a screen where the user can read the data of the application, in particular the description of its functionalities, the contacts of the managers of the system and information about the investors.

4. **Log in as a passenger.** The system provides an input form where the passenger can insert e-mail and password. The system checks if the account is already registered and if the password is correct, if not an error message is displayed. If this conditions hold, the passenger logs in successfully.

5. **Log in as a taxi driver.** The system provides an input form in the mobile application where the taxi driver can insert e-mail, ID and password. The system checks if the account is correctly registered, if the ID matches the e-mail address and if the password is correct, if not an error message is displayed. If this conditions hold, the taxi driver logs in successfully.

6. **Log in with social networks** The system allows the user to sign up with his Facebook or Google+ account without inserting e-mail, telephone number and the other information, if he has already inserted them on his profile.

7. **Retrieve password.** The system provide a link to recover the password of the account. The user clicks on the link and receives a recover e-mail with a link that redirects to a page where he can set his new password.

8. **Request a taxi.** The user can make taxi requests enabling the GPS of his device so that the system can determine his position and send the call to the first taxi driver of the queue of the designated area. The visitor can exploit this service in the home page of the application, inserting his e-mail and telephone number. The passenger can use this service in the passenger area, displayed after the log in, without inserting any further information.

9. **Reserve a taxi.** The system provides a "Make a reservation" button in the passenger area. The passenger presses this button and an input form is displayed, where he should indicate the origin, destination and time of the ride. In the first time the system retains the reservation, then forwards it to the first taxi driver in the queue of the designated area, ten minutes earlier the specified time.

10. **Choose payment method.** The system, after the passenger submits a request or a reservation, displays a panel to him, where he can choose whether to use cash or credit card as payment method. If the passenger decides to pay by cash, the taxi driver will take care of the payment. Instead the passenger needs to add his credit card data only if this is the first time he chooses this payment method, in fact the system adds the credit card data to his profile after the insertion and does not ask anymore. The system deducts the cost of the ride from the credit card only when the ride is over, sending also the receipt of the payment to the e-mail address indicated in the passenger profile.

11. **View the profile.** The passenger can see his profile by pressing the "See profile" button. The system displays a screen with his personal data already inserted and his profile picture.

12. **Edit the profile.** The passenger can edit his profile by pressing the "Edit profile" button and the system displays a screen where he can change his name, surname, e-mail address, password, phone number and profile picture. He can edit or add his gender, date of birth and credit card data. In case the passenger modifies his e-mail address, the system will send an e-mail confirmation.

13. **See waiting time.** The system displays to the user who has made a request the waiting time panel, from the time a taxi driver is found until he arrives. The user can see the panel to know how much time he has to wait till the taxi arrival.

14. **Change availability.** The taxi driver can switch his state from available to unavailable and vice versa. The system does not consider him in the queue as long as his state is unavailable and, if he is available, the system shows him his queue positioning and keeps him in consideration for the requests of the users. Moreover the taxi driver doesn't need to set his state unavailable when he takes a call, as it is already managed by the system.

15. **Notification of the requests.** The taxi driver can see the requests by a user as notifications in the notification panel. The system sends him

a notification with the position of the user, the taxi driver can decide to accept the request or refuse it.

16. **See position in the queue.** The system provides the taxi driver his position in the queue in his area, if his state is set to available, so that he is always able to know if there can be any incoming requests. The system constantly updates (every minute) the position of all drivers of all areas.

3.3 Non-functional requirements

3.3.1 Performance requirements

The system is composed of a server side hosting the database, where all the data of the passenger and of the taxi driver are stored, and it handles all HTTP request. There is not a great amount of data to manage, there are only profile information of the passengers, of the taxi drivers and the log of the activities, then the size of the database is not a constraint for the system.
To supply a suitable service, the system has to be reactive and able to answer to a high number of simultaneous request.
There are no time constraints that depends on the system, the time to process a transaction is less then one second. The only time constraint would be the reply message of the taxi driver that depends on his reflexes.

3.3.2 Software system attributes

1. Reliability: the server must remain up twenty-four hours a day in order to guarantee a suitable service. The system allows the administrators to fix any problem without compromising the functionality of the system.

2. Availability: in order to guarantee the continuous availability of the data, it has been arranged one backup system on a secondary database and the other on an external storage.

3. Security: privacy of the data exchanged between the server and the clients is guaranteed by a SSL encryption. The passwords and the sensitive data stored on the database are protected by MD5 encryption.

4. Portability: the client side of the system is compatible with the major mobile platforms on the market (e.g. Android, iOS) and any device (e.g. PC with Windows, Apple, etc.) that supports a browser and a GPS system.

4 Scenarios Identifying

4.1 Scenario 1: Registration and log in

USER1 is new to the city and is not used to use public transport, so he searches for the best taxi services in various forums. After a while, he decides to download the new application "myTaxiService", because it seems to offer the most interesting functionalities. USER1 registers in the application, without even filling the form, accessing with his Facebook account and adds his favorite photo with his girlfriend. Now he is able to log in, to make taxi reservations choosing the payment method he prefers and to request a taxi without having to insert his information every time he needs it.

4.2 Scenario 2: Password recovery

USER1 has registered in the application when he visited the city a few months ago, now he is back and needs to make a taxi reservation for the afternoon. But USER1 is really absent-minded and is not able to remember his password. He is very happy to see the button "password recovery" and he press it: this leads to another page where the system tells him that an e-mail for the recovery has been sent to his e-mail address. He checks his inbox and follows the link to set his new password. Now he can easily log in and make his reservation.

4.3 Scenario 3: Choose payment method

USER2 often uses myTaxiService for the taxi requests, as it is a fast and reliable service. He is a very lazy person and does not want to register, as the application allows visitors to easily request a taxi, but he is very tired of paying by cash, so he finally make up his mind. He registers and adds his credit card data to his profile. USER2 is happy to save his time and he can request or reserve a taxi without even worry about the money spent.

4.4 Scenario 4: Reserve a taxi

USER3 asks his friend USER3gf next door out for a date at the restaurant and then at the cinema. He really wants to make a good impression, but his car is stuck at the mechanic. So USER3 decides to go anyway by bus, even tho it is not the most elegant solution, but when he will come back there will not be any public transport. He decides to register in the myTaxiService application and to make a reservation for 00:30 pm, when the movie ends. The taxi is already there when they come out of the cinema and they are home in a few minutes, USER3 and USER3GF are very satisfied with the service and with the evening too.

4.5 Scenario 5: See requests and reservations history

USER4friend suggested USER4 the application myTaxiService. They are both usually going up and down the whole town in order to meet their clients, so they need a reliable and cheap transport. USER4 is very happy to use this application because he does not always remember the exact time of his reservation, but just accessing the passenger area is able to see it. At the end of the week,

USER4friend needs to make his boss know how many rides he did for the clients, but he has lost the Excel file where he put all the log of his work activity. Instead of depressing, USER4friend remembers that he can access the application and see his requests and even the time he has made them.

4.6 Scenario 6: Taxi driver availability

DRIVER1 is used to take a coffee after his first ride of the day. Even if he is the first of the queue, he wants to maintain this habit, so he just switch his availability button to off. Now the system knows that he cannot be called for the requests and he can set his state available whenever he wants and see his position in the queue. Finally, when DRIVER1 takes a call, he knows that he does not have to bother about his availability because the system always sets his state unavailable in the exact time he takes the request of a passenger.

4.7 Scenario 7: Taxi drivers notification of the requests

DRIVER2 is the first of the queue, so he knows that he is going to take a request soon but he is on the phone with his mother and does not see the notification within the first minute, so the system forwards it to the second taxi driver of the queue. Now he is in the last position of the area and needs to be very careful when the positions change. DRIVER2 shifts fast to the first position again and he is now able to click on the "Accept" button before the time is over. He sees the position of the request and goes there with no trouble. At the end of the day he scrolls his notifications panel and he is very satisfied with his work today, as he has missed just one request.

5 UML Models

5.1 Use Case

5.2 Class diagram

5.3 Sequence diagram

5.4 Statechart diagram

6 ALLOY Modeling

6.1 Data type

6.2 Abstract entity implementation and signature

6.3 Fact

6.4 Assert

6.5 Predicates

6.6 Result

6.7 Generated world

7 Appendix

7.1 Glossary

In order to avoid ambiguity, some words, that will be often used in this document, will be given a precise definition of what will be the meaning in the contest of this project.

- User: a person who requests a service from the system. It can be a visitor or a passenger.

- Visitor: a person who is not registered in the application.

- Passenger: a person who is registered in the application.

- Taxi driver: a taxi driver who access the application with a specific ID.

- Request: the request of a taxi in a certain zone and place of the city made by a user.

- Reservation: the reservation of a taxi in a certain zone, place and time that can be made only by passengers.

7.2 Software and tool used

- TeXstudio (http://www.texstudio.org/): to redact and to format this document.

- draw.io (https://www.draw.io/):to create Use Case Diagrams, Sequence Diagrams, Class Diagrams and Statechart Diagrams.

- Balsamiq Mockups(http://balsamiq.com/products/mockups/): to create mockups.

- Alloy Analyzer(http://alloy.mit.edu/alloy/): to create Alloy models.

7.3 Hours of work

Time spent redacting this document:

- Cattaneo Michela Gaia: ~18 hours of work.

- Barlocco Mattia: ~18 hours of work.