



Requirements Analysis and Specification Document

Software Engineering 2: “myTaxiService”

Fioratto Raffaele, Longoni Nicolò
Politecnico di Milano

A.Y. 2015/2016

October 22, 2015

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Actual System	4
1.3	Scope	4
1.4	Actors	4
1.5	Goals	5
1.6	Definitions, Acronyms, Abbreviations	5
1.6.1	Definitions	5
1.6.2	Acronyms	5
1.6.3	Abbreviations	5
1.7	Identify Stakeholders	6
1.8	Reference Documents	6
1.9	Document overview	6
2	Overall Description	7
2.1	Product perspective	7
2.2	User characteristics	7
2.3	Constraints	7
2.3.1	Regulatory policies	7
2.3.2	Hardware policies	7
2.3.3	Interfaces to other applications	7
2.3.4	Parallel operation	8
2.3.5	Documents related	8
2.4	Assumptions and Dependencies	8
2.4.1	Assumptions	8
2.5	Future possible implementation	8
3	Specific Requirements	10
3.1	External Interface Requirements	10
3.2	Functional Requirements	10
3.2.1	[G1] Allow unregistered passengers to sign up for the service . . .	10
3.2.2	[G2] Allow registered passengers to request a taxi	10
3.2.3	[G3] Allow registered passengers to reserve a taxi	10
3.2.4	[G4] Allow registered passengers to cancel a reservation	11
3.2.5	[G5] Guarantee a fair management of the taxi queues	11
3.2.6	[G6] Allow taxi drivers to manage their workday	11
3.2.7	[G7] Allow the system administrator to update the information about drivers and taxis	11
3.2.8	[G8] Enforce the validity of a taxi driver's license	11
3.2.9	[G9] Enforce the validity of all the documents belonging to a taxi	11
3.3	The World and the machine	11
3.4	Scenarios	11
3.4.1	Scenario 1	11
3.4.2	Scenario 2	11
3.4.3	Scenario 3	12
3.5	UML models	13
3.5.1	Use case	13

	3.5.1.1	Unregistered Passenger signs up into myTaxiService . . .	14
3.6		Non Functional Requirements	14
	3.6.1	Performance Requirements	14
	3.6.2	Design Constraints	14
	3.6.3	Software System Attributes	14
		3.6.3.1 Availability	14
		3.6.3.2 Maintainability	14
		3.6.3.3 Portability	14
	3.6.4	Security	14
		3.6.4.1 External Interface Side	14
		3.6.4.2 Application Side	14
		3.6.4.3 Server Side	14

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 limit of the software and simulate the typical use cases that will occur after the development. This document is intended to all developer and programmer who have to implement the requirements, to system analyst who want to integrate other system with this one, and could be used as a contractual basis between the customer and the developer.

1.2 Actual System

The software house wants to offer a new service for optimizing an existing taxi service of a large city. We suppose that until now nothing has been created and we have to create the entire application without using or modify a previous system.

1.3 Scope

The aim of the project is to create a brand new system that optimizes an existing taxi service. The system will be capable of automatizing and/or simplifying certain processes during the request or reservation of a taxi. It will also guarantee a fair management of the taxi queues. New passengers can sign up for the service inserting some basic information in order to use the service's features as quickly as possible. A passenger can request a taxi using a web service or through a mobile application after registration. The system will be able to localize precisely the position of the passenger and thus determine the taxis that are available near him/her. The system will select a taxi and then will forward the request the the driver. Upon confirmation, the system will notify the customer about the operation completed successfully and the ETA of the taxi. A passenger can also reserve a taxi by specifying the time and date when he/she will catch the ride. The passenger that wants to reserve should do it with at least two hours of premonition, cancellation it is also permitted. The system will actually process the request ten minutes before the time specified during reservation in the same way as described previously.

1.4 Actors

- Unregistered Passenger: a visitor, i.e. a person that is not yet registered to the system thus the only page that can view is the registration one.
- Registered Passenger: all customers that have a valid username and password and that are able to access all the features available to the customers, they can request or reserve a taxi, they can also cancel these operations with some constraints.
- Taxi Driver: a person who is able to drive a taxi, owns a valid professional driving license. The driver informs the system about the availability of the taxi that belongs to him/her logging in into a specific mobile application installed in his/her smartphone, and informs the system when the ride ends.

- System Administrator: is the person in charge of update the list of the actual cars and the list of the drivers that are enabled to work.

1.5 Goals

List of goals for myTaxiService system:

- [G1] Allow unregistered passengers to sign up for the service
- [G2] Allow registered passengers to request a taxi
- [G3] Allow registered passengers to reserve a taxi
- [G4] Allow registered passengers to cancel a reservation
- [G5] Guarantee a fair management of the taxi queues
- [G6] Allow taxi drivers to manage their workday
- [G7] Allow the system administrator to update the information about drivers and taxis
- [G8] Enforce the validity of a taxi driver's license
- [G9] Enforce the validity of all the documents belonging to a taxi

1.6 Definitions, Acronyms, Abbreviations

1.6.1 Definitions

1.6.2 Acronyms

- RASD: Requirements Analysis and Specification Document
- DB: DataBase
- DBMS: DataBase Management System
- API: Application Programming Interface
- ETA: Estimated Time of Arrival
- OS: Operating System

1.6.3 Abbreviations

- [Gn]: n-goal
- [Rn]: n-functional requirement
- [Dn]: n-domain assumption
- [Fn]: n-figure
- [An]: n-assumption

1.7 Identify Stakeholders

Our main stakeholders are the members of the government of the city that are interested in optimizing the already present taxi service. Another stakeholders can be a company that wants to enhance the service as well, but also the passengers and the taxi drivers which they want a more comfortable, intuitive and user-friendly service.

1.8 Reference Documents

- myTaxiService Specification Document
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

1.9 Document overview

The document is essentially divided into four parts:

- Section 1: Introduction, it gives a description of document and some basic information about software.
- Section 2: Overall Description, gives general information about the software product with more focus about constraints and assumptions.
- Section 3: Specific Requirements, this part list requirements, 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: Appendix, this part contains some information about the attached files and some described screenshot of software used to generate it.

2 Overall Description

2.1 Product perspective

The final product will consist of a web service and a mobile application for the customer and the taxi driver. The overall system will also have an internal interface for the system administrator. The system should also interact with existing external services, and will also provide additional APIs that allows the development of additional features in the future.

2.2 User characteristics

train the system administrator (i.e. take an existing employee and train it to manage the system)

2.3 Constraints

2.3.1 Regulatory policies

For the myTaxiService we have identified the following regulatory policies:

- Cookies: since it is mandatory to log in into the web service to interact with the system, it is necessary to use cookies in order to keep track of the session of each user, thus the system must comply with the actual Cookie Law.
- Privacy: the system needs to manage some sensible data of the customers and the taxi drivers, for instance the name, the surname, an e-mail etc. Another crucial information is the actual location detected by the GPS of both. Thus during the registration the customers must accept the Privacy Law (see login mockup), regarding the taxi driver, before being registered to the system by the administrator, the driver should sign some sort of document accepting the processing of its personal data.
- Drivers during their work are subject to the Traffic code.
- The workday of a driver is subjected to the laws about Job safety, so the system must allow the driver to take some breaks during the day in order to prevent some illnesses or damage to people or things because of tiredness or stress.

2.3.2 Hardware policies

For the myTaxiService we have identified the following regulatory policies:

- Smartphone: they have limited power and battery so care must be taken when developing the applications and also the web service (for instance, implementing a responsive design).

2.3.3 Interfaces to other applications

- Google Maps: they will be used to provide an estimation about the time of arrival of the taxi to the passenger.

- External Database containing information about the drivers' license and the data about the cars, the system will interrogate this database to check the correctness of the information supplied by the taxi driver during the registration in the system.

2.3.4 Parallel operation

myTaxiService must support parallel operations from different users when working with database and with all operation done by the user after connection.

2.3.5 Documents related

2.4 Assumptions and Dependencies

2.4.1 Assumptions

- [A1] Only registered users can use the service, because additional information such as name, surname, an e-mail to confirm and recap the reservation are needed, thus avoid repeating the user to re-enter this information every time. Another reason is to prevent the passenger to do another request to the system while theres another one being processed.
- [A2] The position of the passenger is exactly known using the GPS available in the smartphone or by using a geolocalization API from the browser.
- [A3] The position of the taxi is exactly known using the GPS available in the smartphone of the driver
- [A4] The city is divided into zones of 2 km^2 each and for each zone a taxi queue is allocated.
- [A5] The driver can take at most 2 breaks during his workday, during the break the taxi that belongs to him is not available.
- [A6] A taxi driver must own a valid professional driving license and a taxi with all the documents in order
- [A7] Each taxi driver's license has a unique code that identifies it
- [A8] Each taxi cab has a unique code that identifies it
- [A9] external database with information about taxi drivers and their car
- [A10] The existing taxi service is available 24 hours per day, 365 days per year.
- [A11] An ETA of at most 10 ± 5 mins is considered adequate (move to domain assumptions?)

2.5 Future possible implementation

- Allow the customers to pay using an electronic payment system. During registration the passenger will provide additional information about the his/her credit card or PayPal e-mail, the system will charge a ride automatically.

- Extend the interface usable by customers to provide the ability to evaluate the service in general as a sort of feedback to improve the overall system.
- Expand the system to allow the possibility to share a taxi with other customers using the existing API that will be implemented.

3 Specific Requirements

3.1 External Interface Requirements

3.2 Functional Requirements

3.2.1 [G1] Allow unregistered passengers to sign up for the service

- [R1] The system must request an e-mail or a telephone number during the registration
- [R2] The system must reject an e-mail or a telephone number already used by another registered passenger
- [R3] After the completion of the registration process, the system must send a confirmation to the newly registered passenger
- [R4] The system shall check if the e-mail or the telephone number is well formed
- [R5] The system will consider the passenger officially registered if and only if the passenger opens an activation link in the confirmation e-mail, or inserts the code received through an SMS in the mobile application or in the web service.

3.2.2 [G2] Allow registered passengers to request a taxi

- [R1] The system must disallow unregistered passengers to perform such operation
- [R2] The system shall locate the passenger using the GPS in the smartphone or through the browser.
- [R3] After selecting the most appropriate taxi (explained in the goal [G4]), the system must inform the driver that a request is incoming.
- [R4] Upon confirmation by the driver about the request of availability, the system shall inform the passenger about the positive outcome and provide additional information, such as ETA
- [R5] In the event of unavailability of the driver contacted, the system shall pick another taxi from the most appropriate queue and repeat what described in requirement [R3]
- [R6] The system should also provide a functionality that allow the passenger to cancel the request if, for instance, the passenger doesn't consider the ETA provided adequate, the system will accept this intention if and only if it is not in contrast with assumption [A10]

3.2.3 [G3] Allow registered passengers to reserve a taxi

- [R1] The system must disallow unregistered passengers to perform such operation

- 3.2.4 [G4] Allow registered passengers to cancel a reservation
- 3.2.5 [G5] Guarantee a fair management of the taxi queues
- 3.2.6 [G6] Allow taxi drivers to manage their workday
- 3.2.7 [G7] Allow the system administrator to update the information about drivers and taxis
- 3.2.8 [G8] Enforce the validity of a taxi driver's license
- 3.2.9 [G9] Enforce the validity of all the documents belonging to a taxi

3.3 The World and the machine

3.4 Scenarios

3.4.1 Scenario 1

Ilaria is late for work and unfortunately there is no available public transportation to get quickly at the office. She launches the myTaxiService mobile application from her smartphone and she logs in. She wants to request a taxi so she taps the button "Request a Taxi". The smartphone is able to locate quickly Ilaria and, though the Internet, the application sends to the system her coordinates and the Ilaria's intention to request a taxi. The system looks up the taxi queue near Ilaria's position and sends a notification to the driver associated with the first available taxi in that queue that a request is incoming. The mobile application of Daniele, the taxi driver, receive the notification but unfortunately due to heavy traffic in that specific street he doesn't confirm his availability to carry out the request, and he sends a message by tapping the button "Not Available" to the system. The system acknowledges this and moves the taxi from the top of the queue to the end, and selects the second taxi (which now is the first one) from the same queue. This time Marco confirms his availability tapping the button "Available" in the application. The system receive the message and thus allocates the previously selected taxi for Ilaria's request, while sending a confirmation message to her providing additional information such as the taxi code and its ETA and she waits for the taxi to come to her position. Meanwhile Marco has received Ilaria's position from the system and it is displayed in his smartphone, thus he moves from that location. When Marco arrives he picks up Ilaria, she provides the destination to Marco and the ride starts. When the destination is reached, Marco taps the button "Destination reached" and sends the corresponding message to the system. Ilaria pays the amount due and she leaves the taxi and reaches the office. Meanwhile the system asks for the exact location of Marco's taxi by sending a message. The application answers by sending back another message with the coordinates detected by the GPS, eventually the system adds the taxi to the appropriate queue according to the location received.

3.4.2 Scenario 2

Davide has just arrived on the city, a friend suggested him about the myTaxiService, so he installs the application in his smartphone and tries to register. He provides to the application an e-mail, his name, his surname and the date of birth, he accepts the privacy and cookie policy and taps the button "Register". While checking if the e-mail

is adequate i.e. it is well formed and not already used, the system discover that the e-mail was already used and so Davide receives a notification in the application about the problem and inserts another e-mail. This time the e-mail is accepted and the system stores his information in the database, while sending a confirmation e-mail to address provided previously, containing a temporary password. To conclude the registration process Davide must log in at least one time, using the e-mail provided during the registration and the password generated by the system. As soon as he logs in, the system considers Davide a valid registered passenger and he can now use the other system's functionalities, and also change his password with another one.

3.4.3 Scenario 3

Bob has booked a flight so he decided to reserve a taxi driver that will take him to the airport. From his home, he browses to the myTaxiService website, he logs in and presses the button "Reserve a Taxi", a new page loads up with a map and a textual box. The website asks him the permission to determine his position using the browser, but he decline because he wants to set a starting position that is different than the current one. He inserts the address in the textual box and presses "Enter" on the keyboard. The map will update moving the region centering the specific position inserted previously, placing also a marker to emphasize the position detected by the system. Then Bob inserts the time and the date of the reservation of when the taxi shall arrive to the desired location and presses "Confirm". The system stores this data into the database since all the information provided are adequate, Bob receive an e-mail about the operation being successfully completed. When the date is approaching, specifically 10 minutes before the time and date specified the system process the reservation managing it in the same way as a request already described in Scenario 1.

3.5 UML models

3.5.1 Use case

3.5.1.1 Unregistered Passenger signs up into myTaxiService

3.6 Non Functional Requirements

3.6.1 Performance Requirements

The overall performance of the system must be acceptable so that a good level of usability is achieved. The response time with the respect to the system must be nearly zero. The response time of the mobile application is tricky to evaluate since there's a lot of different devices with different power processing, but in general the application should perform well on the majority of the devices available as of today in the market. Some very low level or entry level devices might show some delays in rendering the application on the screen. Nevertheless the major bound is the customer and driver's internet connection.

3.6.2 Design Constraints

3.6.3 Software System Attributes

3.6.3.1 Availability The overall system must be available anytime according to [A10]. To obtain this a dedicated server must be used. In order to achieve even more reliability, some fault tolerant mechanisms must be implemented, also to reduce the risks of high load that could eventually result in a denial of service, the server must be replicated so that the work can be done in parallel by multiple instances of the same application.

3.6.3.2 Maintainability The application will provide specific APIs that will be documented. The whole application will be documented as well.

3.6.3.3 Portability The mobile application will be developed on all the major mobile OS available in the market, also the web application is inherently portable since in general at least a browser exists for any widespread OS. The web application will be designed with a responsive layout so that it is also accessible from mobile devices.

3.6.4 Security

3.6.4.1 External Interface Side

3.6.4.2 Application Side

3.6.4.3 Server Side