

myTaxiService
Requirements Analysis and Specification
Document

Belluschi Marco, Cerri Stefano, Di Febbo Francesco

November 1, 2015

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	4
1.3.3	Abbreviations	4
1.4	Actors	4
1.5	Identifying stakeholders	4
1.6	Reference documents	5
1.7	Overview	5
2	Overall Description	6
2.1	Product perspective	6
2.2	Product functions	6
2.3	User characteristics	6
2.4	Constraints	7
2.4.1	Regulatory policies	7
2.4.2	Hardware limitations	7
2.4.3	Interfaces to other applications	8
2.4.4	Parallel operation	8
2.4.5	Reliability requirements	8
2.4.6	Criticality of the application	8
2.4.7	Safety and security considerations	8
2.5	Assumptions and Dependencies	8
2.5.1	Assumption	8
2.6	Future implementation	8
3	Specific Requirements	9
3.1	External Interface Requirements	9
3.1.1	User interfaces	9
3.1.2	API interfaces	16
3.1.3	Hardware interfaces	16
3.1.4	Software interfaces	16

3.1.5	Communication interfaces	17
3.2	Functional Requirements	17
3.3	The world and the machine	17
3.4	Scenarios	17
3.5	UML Models	17
3.6	Non Functional Requirements	17
3.6.1	Performance Requirements	17
3.6.2	Software System attribute	17
3.6.3	Security	17

Chapter 1

Introduction

1.1 Purpose

This document represent the Requirement Analysis and Specication 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 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 Scope

The system described in this document is a taxi service for large cities. The main goals of the system are: 1) simplify the access of passangers to the service 2) guarantee a fair management of taxi queues. The system is composed by a web application, a mobile application and a web server.

There are three types of actors that can use the system: visitors, taxi drivers and passengers. Visitors have only two operations allowed: log in or sign in. Passengers can use both the web application and the mobile application to request a taxi. Taxi drivers use only the mobile application to modify their status and to confirm to the system that they are going to take care of a certain request from a certain passenger.

The system, when a passenger request a taxi, informs an available taxi driver (FIFO mode) about the current position of that passenger. At this time the taxi driver has two options:

- accept : the system sends a notification to the passenger with the estimated waiting time

- reject : the system searches for another available taxi driver

The system allows also a passenger to:

- reserve a taxi by specifying the origin and the destination of the ride
- share a taxi with others (if possible) by specifying all the rides that he/she wants to share. In this case the system defines the cost of the ride for each passenger

Besides the specific user interfaces for passengers and taxi drivers, the system offers also APIs to enable the development of additional services on top of the basic one.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- User: person that uses the service applications.
- Visitor: user that has not registered nor logged in.
- Registered user: user that has registered to the service.
- Passenger: passenger registered to the service.
- Taxi driver: taxi driver registered to the service.
- System: the union of software and hardware to be developed and implemented.

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 Actors

There are three types of actors that use MyTaxiService:

- visitors: they can only log in or sign up
- passengers: they can request, reserve or share a taxi
- taxi drivers: they can accept/deny a ride and modify their status

1.5 Identifying stakeholders

The main stakeholder of the project is the government of a large city. The government, with the help of the transport council, decided to improve the actual taxi service with MyTaxiService. With MyTaxiService the stakeholders want to:

- symplify the access of passangers to the service
- guarantee a fair management of taxi queues
- give the possibility to the passengers to reserve a taxi
- give the possibility to a passenger to share a taxi with other passengers

1.6 Reference documents

- Software Engineering 2 Project AA 2015/2016: Project Description And Rules
- Software Engineering 2 Project AA 2015/2016: Assignments 1 and 2 (RASD and DD)
- Software Engineering 2 Project AA 2015/2016: RASD-meteocal-example1
- Software Engineering 2 Project AA 2015/2016: RASD-meteocal-example2

1.7 Overview

This document is essentially structured in three parts:

- Section 1: Introduction: it gives a description of the document and some basical information about the system. It also identifying the stakeholders and the actors involved.
- Section 2: Overall Description: it gives general information about the software and hardware product, constraints and assumptions.
- Section 3: Specific Requirements: this is the core of the document. It describes the functional and non-functional requirements combined with some scenarios. There is also a class diagram that gives an overall representation of the system.

Chapter 2

Overall Description

2.1 Product perspective

The system is composed by a web application, a mobile application and a web server. The web application runs on most common browsers, namely Chrome, Internet Explorer, Firefox, Safari. It needs a web server that supports PHP. The mobile application needs a platform supporting Android, iOS or Windows Phone. Both applications interact with a DBMS. Additional functionalities are provided through the use of APIs or interfaces, i.e. taxi reservation and the taxi sharing option.

2.2 Product functions

The system allows different kinds of user to perform different actions. In particular:

- Visitors can simply register or log in, thus becoming either a passenger or a taxi driver user.
- Passengers can request, reserve and share taxi rides.
- Taxi drivers can modify their availability status and respond (accept/refuse) to impending ride requests.

2.3 User characteristics

Registered users can be either passengers or taxi drivers.

The system wants to give both an easy way to interact, thus optimizing the taxi service. To do so, passengers must be able to install and use the mobile application, or use the web application. On the contrary, taxi drivers can only install and use the mobile app; besides, their cellphone must be provided with a GPS. All users must have access to the Internet.

2.4 Constraints

2.4.1 Regulatory policies

myTaxiService is a service provided by the public company **ATM** responsible for public transportation in **Milan** city. The user, who reaches this service by web or mobile application, has to agree to License Agreement rather than Privacy policy and Terms of use at registration.

The user access and use of the services constitutes his/her agreement to be bound by these Terms, which establishes a contractual relationship between him/her and myTaxiService. If user does not agree to these Terms, he/she may not access or use the services. myTaxiService may immediately terminate these Terms or any services with respect to him/her, or generally cease offering or deny access to the Services or any portion thereof, at any time for any reason.

myTaxiService collects the information provided by the user, for example when creating or making changes to services on demand, through contact with customer service or during other communications. This information may include: name, email, phone number, mailing address, profile picture, payment method, products required (for service delivery), delivery receipts and other information user choose to provide. The personal data will be used only to provide the services requested.

User is responsible for obtaining the data network access necessary to use the services. User mobile network's data and messaging rates and fees may apply if he/she accesses or uses the services from a wireless-enabled device. User is responsible for acquiring and updating compatible hardware or devices necessary to access and use the service and applications and any updates thereto. myTaxiService does not guarantee that the services, or any portion thereof, will function on any particular hardware or devices. In addition, the services may be subject to malfunctions and delays inherent in the use of the Internet and electronic communications.

2.4.2 Hardware limitations

myTaxiService defines the minimum requirements for using web and mobile applications.

- *Web application*
Supported browsers: Chrome, Internet Explorer, Firefox, Safari. Other browsers may also work
Web access at the minimum speed of 1Mbps
- *Mobile application*
Operating system: Android, iOS, Windows Phone
Memory: 512MB RAM
Hard drive: 50MB of free space
GPS navigation system (only for taxi drivers)
Web access at the minimum speed of 1Mbps

2.4.3 Interfaces to other applications

myTaxiService does not have to meet any interface to other applications.

2.4.4 Parallel operation

myTaxiService supports parallel operations cause of the nature of service. Many users can access to the service at same time thus system and database have to work with parallel requests.

2.4.5 Reliability requirements

2.4.6 Criticality of the application

2.4.7 Safety and security considerations

2.5 Assumptions and Dependencies

2.5.1 Assumption

- Passenger requests a ride from applications
- Passenger sets a correct meeting point
- Passenger sets a correct destination
- Taxi driver reaches the meeting point
- Taxi driver picks up the correct passenger
- Accurate taxi driver's locations are known by GPS
- Taxi driver reports correctly his availability
- The city is divided in taxi zones
- The taxi queue in a zone contains only taxi drivers available in that zone
- Taxi driver confirms either denies a passenger request call

2.6 Future implementation

Chapter 3

Specific Requirements

3.1 External Interface Requirements

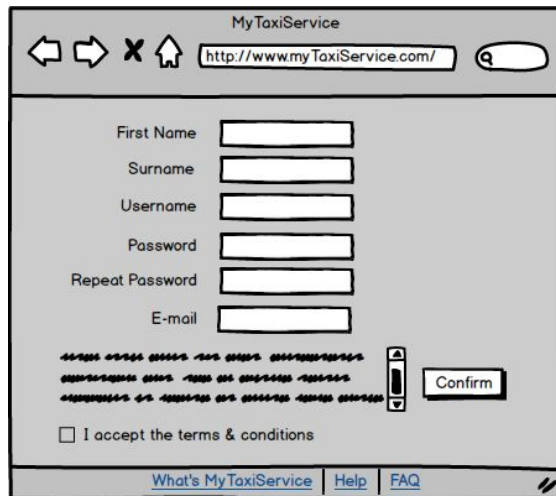
3.1.1 User interfaces

The interface of MyTaxiService can be for web application and mobile application. Here will be presented some of the most important pages and screens of MyTaxiService.

Log in: In the figure below is shown the MyTaxiService's homepage



Registration: Here the visitor can sig up to the application



MyTaxiService

http://www.myTaxiService.com/

First Name

Surname

Username

Password

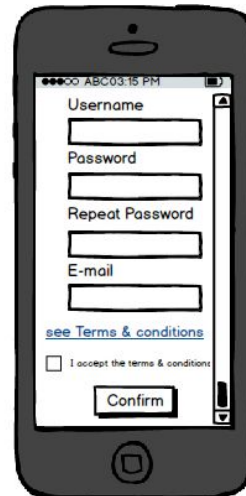
Repeat Password

E-mail

☐ I accept the terms & conditions

[What's MyTaxiService](#) [Help](#) [FAQ](#)

Confirm



ABC03:15 PM

Username

Password

Repeat Password

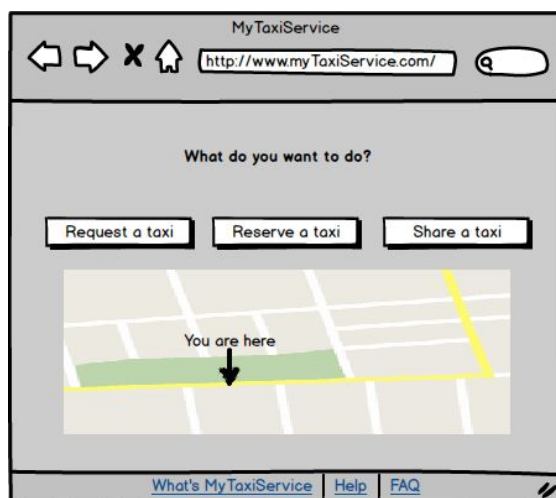
E-mail

[see Terms & conditions](#)

☐ I accept the terms & conditions

Confirm

Passenger view: View of the passenger



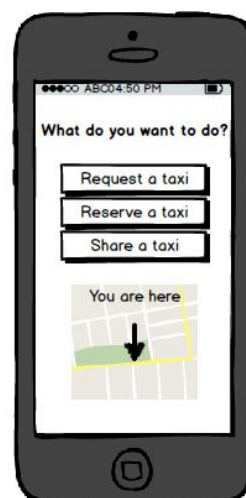
MyTaxiService

http://www.myTaxiService.com/

What do you want to do?

You are here

[What's MyTaxiService](#) [Help](#) [FAQ](#)

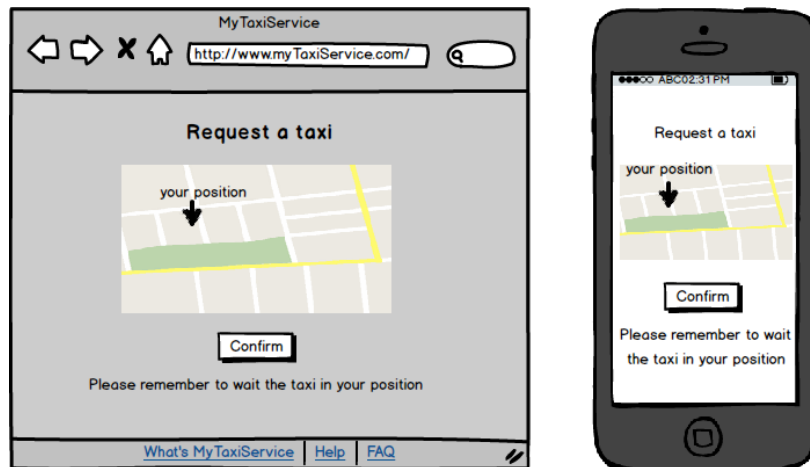


ABC04:50 PM

What do you want to do?

You are here

Request a taxi: View of the passenger when he/she requests a taxi



Reserve a taxi: View of the passenger when he/she reserves a taxi

MyTaxiService

http://www.myTaxiService.com/

Reserve a taxi

Origin [use maps](#)

Destination [use maps](#)

Time: hours/minute

Remember that you can reserve a taxi 2 hours before the ride

[Confirm](#)

[What's MyTaxiService](#) [Help](#) [FAQ](#)

ABC02:31 PM

Reserve a taxi

Origin [maps](#)

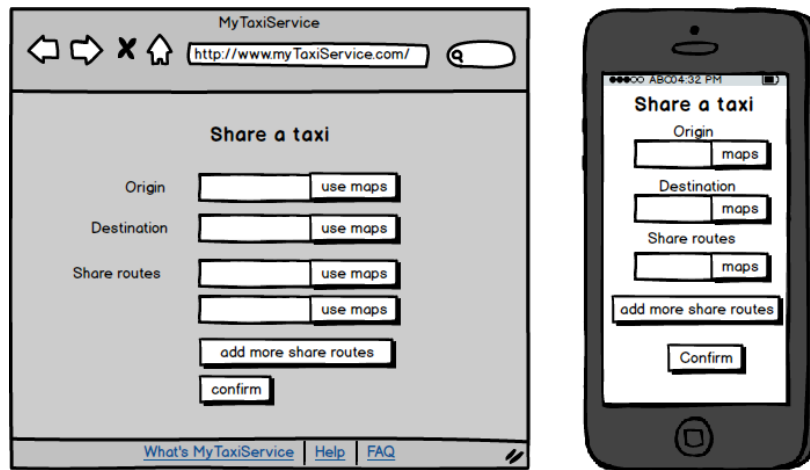
Destination [maps](#)

Time: hours/minute

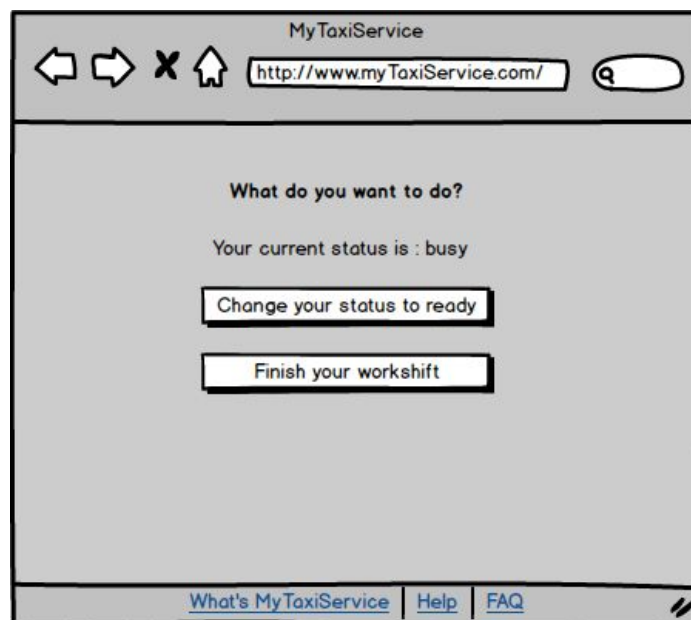
Remember that you can reserve a taxi
2 hours before the ride

[Confirm](#)

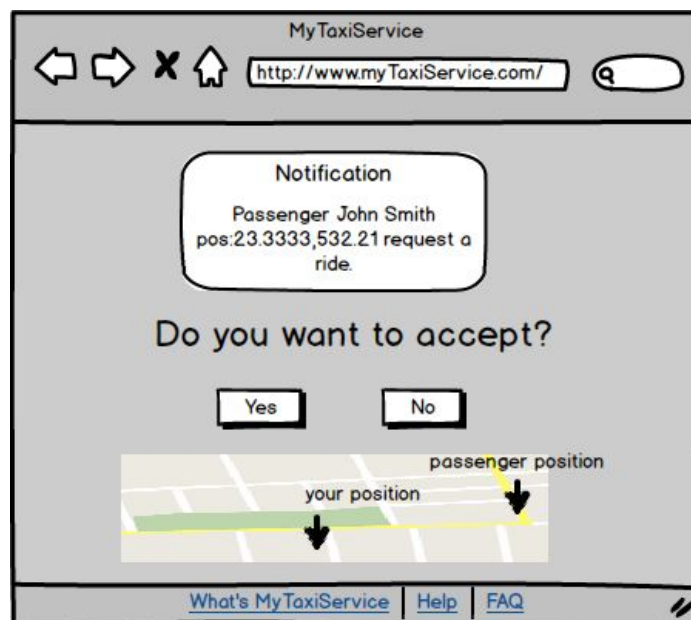
Share a taxi: View of the passenger when he/she shares the a taxi



Taxi driver view: View of the taxi driver



Taxi driver notification: Notification that the taxi driver, chosen by the system, sees when a passenger request a ride.



Passenger notification : Notification that the passenger see when a taxi accept the ride



3.1.2 API interfaces

MyTaxiService uses Google Maps APIs in order to show to the passengers and to the taxi driver their position in the city. This API is continuous update and works for all the OS and browsers web supported by MyTaxiService. More information are available on the site: <https://developers.google.com/maps/>

3.1.3 Hardware interfaces

MyTaxiService doesn't support any hardware interfaces.

3.1.4 Software interfaces

- Database Management System (DBMS):
 Name: MySQL
 Version: 5.7
 Source: <http://www.mysql.it/>
- Java Virtual Machine (JVM):
 Name: JEE
 Version: 7
 Source: <http://www.oracle.com/technetwork/java/javaee/tech/index.html>

- Application server:
Name: Glassfish
Version: 4.1.1
Source: <https://glassfish.java.net/>

3.1.5 Communication interfaces

- Protocol: TCP Service: HTTPS Port : 443
- Protocol: TCP Service: HTTP Port : 80
- Protocol: TCP Service: DBMS Port : 9247

3.2 Functional Requirements

3.3 The world and the machine

3.4 Scenarios

3.5 UML Models

3.6 Non Functional Requirements

3.6.1 Performance Requirements

MyTaxiService will perform 95% of the operations within 4 second; the total amount of the operations within 10 seconds. The system should ensure at least 2000 passangers connected and 500 taxi drivers connected.

3.6.2 Software System attribute

Availability MyTaxiService should be available 99% of the time . In order to do this, a a dedicated server will be used.

Maintainability The implementation of MyTaxiService must be documented with javadoc.

Portability The system could be used on any OS specified in Hardware Limitation Section.

3.6.3 Security

External Interface Side MytaxiService implements a login authentication to protect the information of users. Users, after the registration, have to confirm

their e-mail with the security code sent to the e-mail write in the registration form.

Application Side MyTaxiService must detect SQL injection in order to protect user information.

Server Side The server side architecture will be implemented separating data from application. Application Server must be separated from database and from the web server. All zone are divided by firewall.