­



Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “MyTaxi”

Requirements Analysis and Specifications

Document

Manzi Giuseppe (mat. 854470) &

Nicolini Alessandro (mat. 858858)

CONTENTS

1. Introduction 3

1.1 Purpose 3

1.2 Actual system 3

1.3 Scope 3

1.4 Actors 3

1.5 Goals 4

1.6 Definition, Acronyms and Abbreviations 4

1.6.1 Definition 4

1.6.2 Acronyms 4

1.6.3 Abbreviations 4

1.7 Reference Documents 4

2. Overall Description 5

2.1 Product perspective 5

2.2 User characteristics 5

2.3 Constraints 5

 *2.3.1 Regulatory policies* 5

 *2.3.2 Hardware limitations* 5

 *2.3.3 Interfaces to other applications* 5

 *2.3.4 Parallel operation* 5

 *2.3.5 Documents related* 5

2.4 Assumptions and Dependencies 5

2.4.1 Assumption 5

2.5 Possible future implementations 6

3. Specific requirement 6

3.1 External Interface Requirements 6

3.1.1 User Interfaces 6

# 1. Introduction

## 1.1 Purpose

Software Requirements Specification Document (RASD) is an unambiguous and complete specification document, helping *Clients* to describe their wishes and developers to understand what the *Client* wants. This document also shows constraints and the limit of the software and simulates the typical use cases that will occur after the development. RASD 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 *Client* and the developer.

## 1.2 Actual system

Till last year there were different taxi companies providing the service of managing taxi requests for the whole city. Any taxi driver could become member of one of these companies. When a costumer phoned the company, the system of the company forward the call to the right taxi driver. At the beginning of this year the government of the city decided to unify all the companies and provide a unique service for all the taxies.

## 1.3 Scope

The aim of the project is to create a new system to improve the quality of the service. The passenger can send his request either through a web application or a mobile app, in order to simplify the access to the service thanks to a user-friendly interface and simple functions. The use of mobile app will also be an advantage for taxi drivers, who can install it on their own private device or rent one provided by the government of the city, sign in and receive notification of requests they have to manage.

To guarantee a fair management of both requests and taxi queues, the city will be divided in taxi zones (approximately 2 km2 each). The oldest request coming from a costumer must be the first managed in his zone. A request must be forwarded to the taxi that is free and has been waiting in that zone for more time.

## 1.4 Actors

**Guest user**: He can decide either to sign up or log in (whether already registered) or to request a taxi without signing in.

**Registered costumer**: He can reserve/request a taxi or see the list of all the rides he took or he has reserved. Reserved ride can be modified if it’s more than two hours before the reservation time or deleted at any time.

**Employee**: He can register taxi drivers, after checking their identity.

**Taxi driver**: He is notified when a request/reservation is assigned to him. He can accept or decline the assignment. On his main screen he can see the map of the city, automatically see the way to the location of the costumer he has to pick up and look for the best way to a specific address.

## 1.5 Goals

**[G1]** Allow taxi drivers to have a personal reserved area in the app/site, accessible by login.

**[G2]** Allow guest users to request a taxi to come to a specific address as soon as possible.

**[G3]** In addition to functionalities provided to guest users, a registered costumer can reserve a taxi for a specific time (at least two hour in advance), check previous requests and reservations and modify or delete them.

**[G4]** Guarantee a *fair queue management* (see 1.6.1).

**[G5]** Manage the distribution of the taxis in each zone and notify taxi drivers the need to move from a zone to another.

## 1.6 Definition, Acronyms and Abbreviations

### 1.6.1 Definition

* **Assignment**: the assignment by the system of a ride to a specific taxi.
* **Client**: the person/the company who commissioned the project.
* **Customer**: a person who makes use of the service provided by the *Client*.
  + **Registered costumer**: a *costumer* who signed up with the system and is logged in for the current session.
* **Employee of the “Mobility office”**: a person who works for the “Mobility office”.
* **Fair queue management**: a queue management that ensures that he oldest request coming from a costumer must be the first managed in his zone and the request must be forwarded to the taxi that is free and has been waiting in that zone for more time.
* **Request**: a request to take a taxi as soon as possible.
* **Reservation time**: the time at witch the taxi must be on the agreed place, according to the reservation.
* **Reservation**: a reservation of a taxi for a time that is more than two hours in the future.
* **User**: a person who uses the system.
  + **Guest user**: a *customer* who uses the system without logging in.
  + **Registered user**: either a registered *costumer* or an *employee*.

### 1.6.2 Acronyms

**RASD**: *Requirements Analisys Specifications Document*.

### 1.6.3 Abbreviations

**Employee** for *Employee of the “Mobility office”****.***

## 1.7 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.

# 2. Overall Description

## 2.1 Product perspective

We well release a web site, an Android application and an iOs app. All these applications have not to be integrated with other existing system. They will not have any internal interface for administration but it will be only user based.

The application will provide new interface or API for integration with future project such as shared taxi.

## 2.2 User characteristics

A *costumer* who uses the applications is a person who wants an easy way to take or reserve a taxi. He could be interested either in taking use of the service as fast as possible or in having customized functionalities, such as the possibility to view the list of all his own reservation and to delete them. So we will provide both the possibility of reserve/request the taxi with or without logging in the system. Another kind of user is the taxi driver who need to be notified when a *costumer* is waiting for his taxi and to be lead to the *costumer*’s position by a GPS navigator system. This app is also a new way for taxi drivers to accept or decline the request of user in a few taps. Employees just need an easy way to register taxi drivers. All kind of users must be able to use a web browser or a mobile application.

## 2.3 Constraints

## *2.3.1 Regulatory policies*

MyTaxi doesn’t have to meet any regulatory policies.

## *2.3.2 Hardware limitations*

MyTaxi doesn’t have to meet any hardware limitations.

## *2.3.3 Interfaces to other applications*

MyTaxi doesn’t have to meet any interfaces to other applications.

## *2.3.4 Parallel operation*

MyTaxi must support parallel operations from different users (both taxi drivers and passengers) when working with database.

## *2.3.5 Documents related*

* Requirements and Analysis Specification Document (RASD).
* Design Document (DD).
* User’s Manual.
* Testing report.

## 2.4 Assumptions and Dependencies

### 2.4.1 Assumption

* Pre-existing system are owned by private societies, so it isn’t possible to modify them.
* There is no need of a hierarchy of user to guarantee the safety of the system.
* A *customer* can reserve as many taxies as he wants.
* The user must have installed the app.
* Through the app the user that has been registered can manage his reservation
* Users and taxi have the same app but for login two different sections.
* Notification of the place of the call, for the taxi, and the code of the taxi that accept the ride, for the user, will be shown.
* We assume it is better to allow deletions of reservation even less than two hour before, because it is better to inform the taxi driver than not be at the appointment.

## 2.5 Possible future implementations

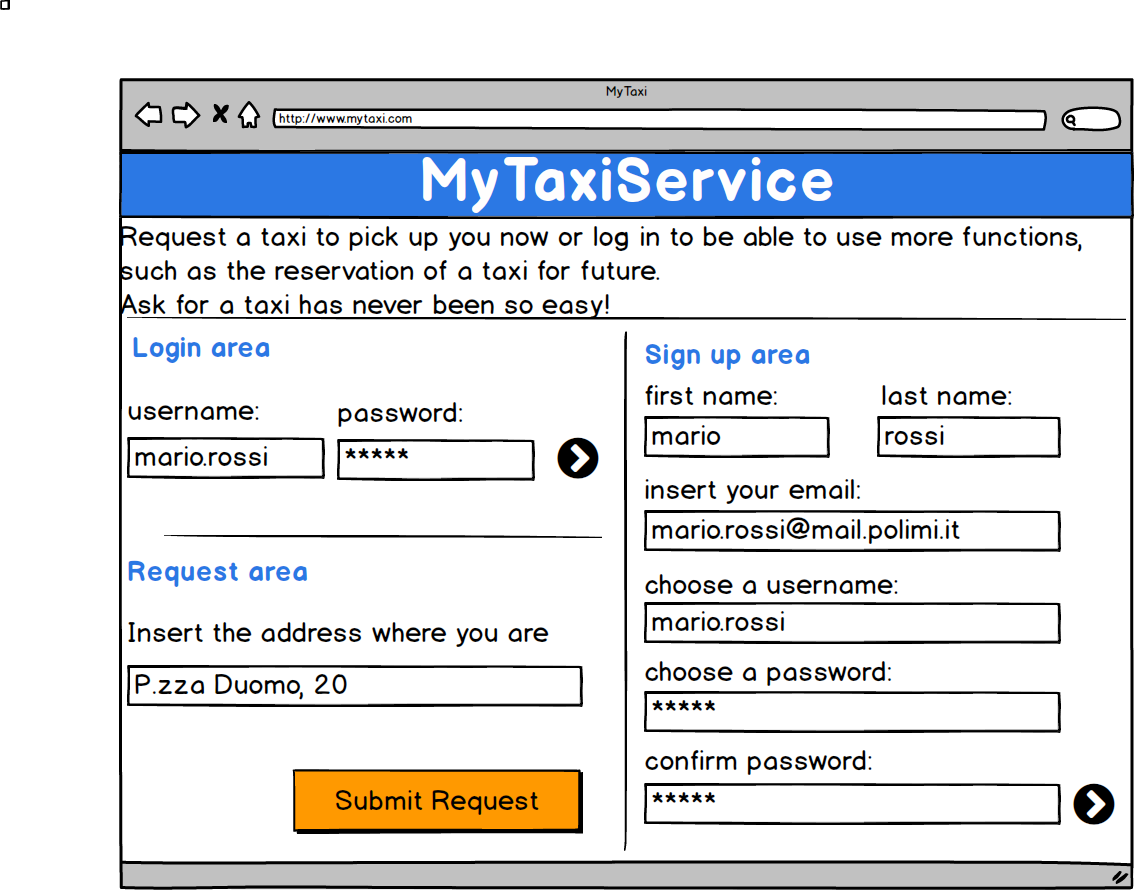
* Possibility to reserve by phoning to a call center.
* Taxi sharing: Possibility for the costumer to decide to share a taxi with others if possible, thus sharing the cost of the ride. In this case the user is required to specify the destination of all rides which he/she wants to share with others. If 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 persons sharing the taxi and informs the passengers and the taxi driver.
* In case of too many abuses, stakeholders could decide to remove the possibility for costumers to send request without signing in.

# 3. Specific requirement

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

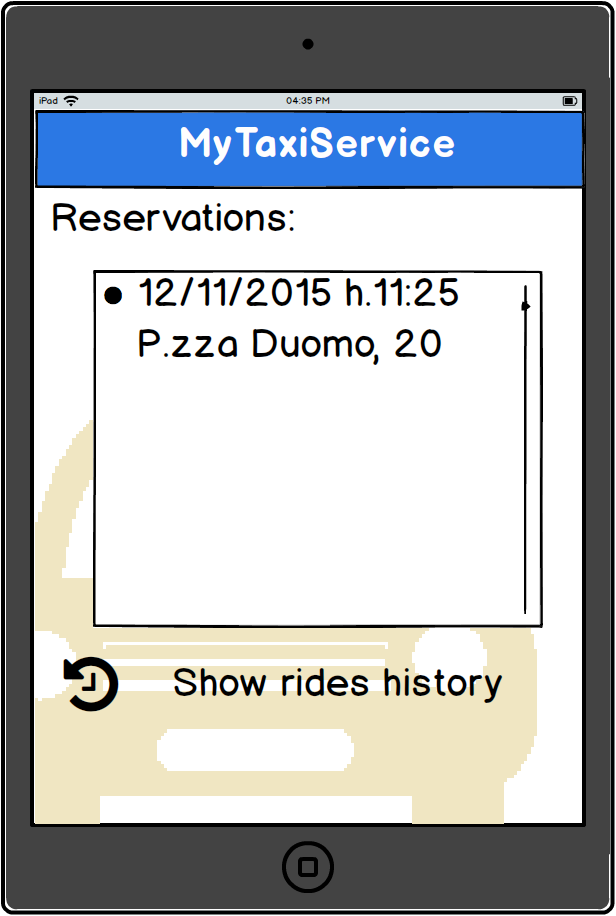
#### Site - Home page



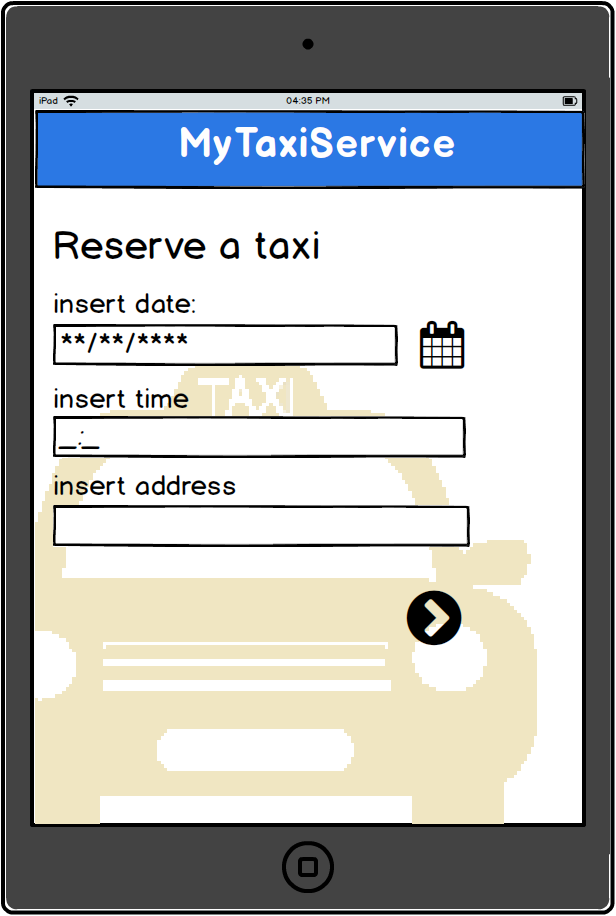
#### App - Registered costumer area - Home page

#### Macintosh HD:Users:Giuseppe:Documents:Peppe:Politecnico:magistrale:1° anno:progetto Sw eng:manzinicolini:All:mockups png:Costumer private area home page.png

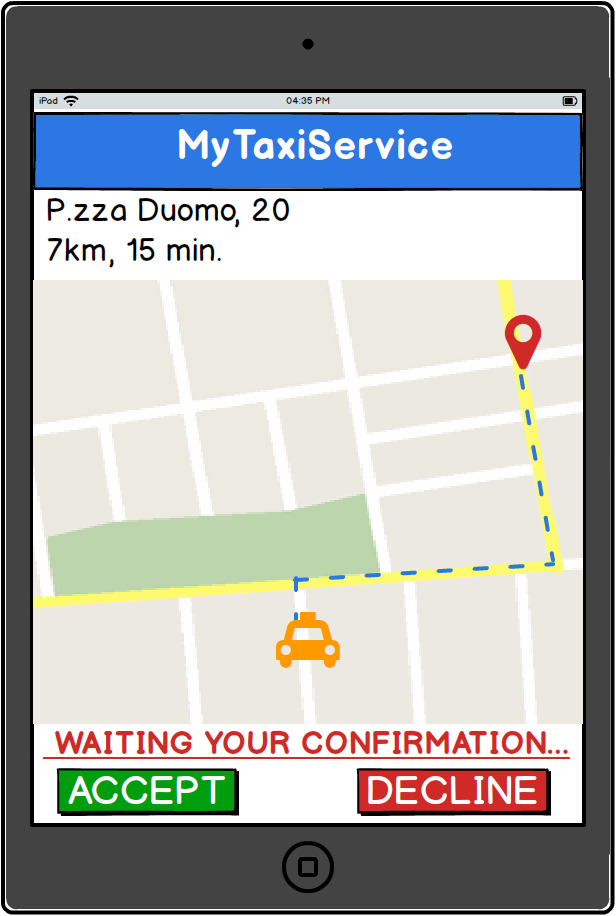
#### App - Registered costumer area - Reservations list page



#### App - Registered costumer area - New reservation page



#### App - Taxi driver area - Notification page



### 3.1.2 API interfaces

[…]

### 3.1.3 Hardware interfaces

[…]

#### 3.1.4 Software interfaces

[…]

### 3.1.5 Comunication Interfaces

[…]

#### 3.1.5 Memory

[…]

## 3.2 Functional requirements

[…]

## 3.3 The world and the machine

[…]

## 3.4 Scenarios

[…]

A taxi driver can either accept or decline the request, if he accepts the system will send taxi code to the costumer else the system will send the request to the second taxi of the queue and move the taxi that rejected the request to the last position of the queue.