

POLITECNICO DI MILANO  
School of Industrial and Information Engineering  
Master course in Computer Science and Engineering  
DEIB Department



## Requirements Analysis and Specification Document (RASD)

Moreno SARDELLA - 859239

Academic Year 2015–2016



# Abstract

**Purpose:** this document represent the Requirement Analysis and Specification Document (RASD) of MyTaxiService project.

**Work area:** Requirements Engineering (RE).

**Scope:** explain the MyTaxiService project requirements in order to explicit the releated software behavior, according to the stakeholders needs.

**Brief summary:** the main activity concerned with collecting and analyzing the stakeholders needs, constantly communicating with them (requirements elicitation and modeling).

It was necessary to know which components are included in a RASD, therefore there was a study about the «The World & the Machine» model (M. Jackson & P. Zave, 1995) and two modeling languages: UML 2.0 (Unified Modeling Language, OMG, 2005) and Alloy (MIT, 1997).



# Contents

<b>Introduction</b>	<b>1</b>
0.1 Purpose of the system . . . . .	1
0.2 Scope of the system . . . . .	1
0.3 Definitions, acronyms and abbreviations . . . . .	2
0.3.1 Definitions . . . . .	2
0.3.2 Acronyms . . . . .	3
0.3.3 Abbreviations . . . . .	3
0.4 References . . . . .	3
0.5 Overview . . . . .	3
<b>1 Overall Description</b>	<b>5</b>
1.1 Overall Description . . . . .	5
1.1.1 Product perspective . . . . .	5
1.1.2 Actors . . . . .	5
1.1.3 Goals . . . . .	5
<b>2 Specific Requirements and Domain Assumptions</b>	<b>7</b>
<b>3 Scenarios</b>	<b>9</b>
<b>4 Modeling</b>	<b>11</b>
<b>Conclusions</b>	<b>13</b>
<b>Bibliography</b>	<b>15</b>
<b>A First appendix</b>	<b>17</b>
<b>B Second appendix</b>	<b>19</b>
<b>C Third appendix</b>	<b>21</b>



# List of Figures

1	Taxi zones . . . . .	2
---	----------------------	---





# List of Tables



# Introduction

## 0.1 Purpose of the system

This document represents the Requirement Analysis and Specification Document (RASD), describes the requirements (functional, non-functional) and the pseudo-requirements (constraints), modeling the MyTaxiService project, according to the stakeholders' needs.

The audience of this document is composed by: the customers and the users, which are interested in validating the system goals and in a high-level description of the project functionalities, the analyst developers, which define the needs to be satisfied, the programmers, which will implement the requirements, the testers, which will check if the requirements will be met.

## 0.2 Scope of the system

The government of Milan aims at optimizing its taxi service.

In particular, it wants to:

- i) simplify the access of passengers to the service
- ii) guarantee a fair management of taxi queues.

A person can register to MyTaxiService giving him/her cellphone number and personal data (also taxi license in case of taxi driver).

Passengers can request a taxi either through a web application or a mobile app.

A passenger is reached by a taxi through him/her GPS coordinates, automatically sent to the system, which 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 (figure 1).

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 an confirmation SMS to the passenger.

If not, then the system will forward the request to the second available taxi 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 an additional service, taxi sharing, on top of the basic one.

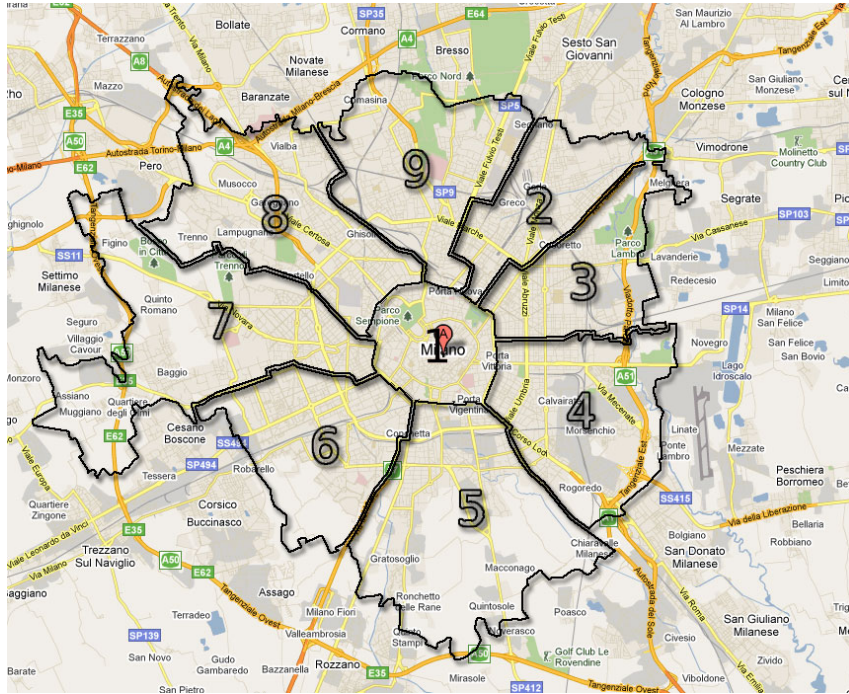


Figure 1: Detailed map of Milan Taxi zones

## 0.3 Definitions, acronyms and abbreviations

### 0.3.1 Definitions

**Notification:** a short SMS sent from the system to a user to signal something.

**Login:** the procedure through which a guest, entering his credentials, authenticates as a user.

**Logoff:** the procedure through which a user disconnects himself/herself from the system.

**System:** the whole MyTaxiService service (includes app and website).

**Sign up:** the procedure through which a guest registers to the service creating a new account.

### 0.3.2 Acronyms

**RASD:** Requirements and Specifications Document

**DD:** Design Document

**UML:** Unified Modeling Language

### 0.3.3 Abbreviations

**ETA:** Estimated Time of Arrival is a term generally used in various fields to indicate the time required for the completion of an operation. In this project it refers to the required time a taxi driver should takes to reach the passenger.

**Gn:** n-th goal

**FRn:** n-th functional requirement

**NFRn:** n-th non functional requirement

**Dn:** n-th domain assumption

## 0.4 References

Specification document: “Assignments 1 and 2”.

The structure of this document follows the standard “IEEE Std 830-1993”.

## 0.5 Overview

This document is composed by four part:

1. Overall Description: information about the software product with more focus about actors and goals.
2. Specific Requirements and Domain Assumptions: information about requirements and domain assumptions.
3. Scenarios: some frequently scenarios.
4. Modeling: UML and Alloy models.



# Chapter 1

## Overall Description

### 1.1 Overall Description

#### 1.1.1 Product perspective

The product which will be developed is a new online system whose requirements will not be subject to those of a legacy system.

The only interaction with external systems will concern with people identification and payment transactions.

#### 1.1.2 Actors

- **Guest:** a person using the service that isn't either logged in or registered.
- **User:** a guest, once logged in the system, becomes a user. A user can be a:
  - **Passenger:** a user who can request a taxi.
  - **Taxi driver:** a user who can supply a passenger request. A taxi driver must have a taxi license.

#### 1.1.3 Goals

1. Allow a guest to:
  - (a) to register
  - (b) to login
2. Allow a user to log in the system like a passenger or like a taxi driver
3. Allow a passenger to:
  - (a) request a taxi

4. Allow a taxi driver to:
  - (a) inform the system about her/his availability
  - (b) confirm/deny an incoming request provided by a passenger, visualizing her/his position
  - (c) confirm/deny the start of the trip within the passenger
5. The system automatically has to:
  - (a) assign dynamically every taxi driver to a zone queue checking her/his GPS coordinates every 15 minutes
  - (b) forward the passenger request to the first available taxi driver in the queue of the zone in which the request comes from (GPS position)
  - (c) send to a passenger a notification about her/his request answer (confirmed/denied, taxi driver ID, ETA)
  - (d) move in the last position of the queue a taxi which has denied a request. Then, 5b.



## Chapter 2

# Specific Requirements and Domain Assumptions



## Chapter 3

# Scenarios



## Chapter 4

# Modeling



# Conclusions





# Bibliography



## Appendix A

### First appendix



## Appendix B

### Second appendix



## Appendix C

### Third appendix

