

# Requirement Analysis and Specification Document

Filippo Agalbato  
850481

Andrea Cannizzo  
790469

November 2, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.3	Definitions, acronyms and abbreviations . . . . .	2
1.4	References . . . . .	3
1.5	Overview . . . . .	3
<b>2</b>	<b>Overall description</b>	<b>4</b>
2.1	Product perspective . . . . .	4
2.2	Product functions . . . . .	4
2.3	User characteristics . . . . .	5
2.4	Constraints . . . . .	5
2.5	Assumptions and dependencies . . . . .	6
<b>3</b>	<b>Specific requirements</b>	<b>7</b>
3.1	Functional requirements . . . . .	7
3.2	Non-functional requirements . . . . .	9
<b>4</b>	<b>System model</b>	<b>10</b>
4.1	Scenarios . . . . .	10

# Chapter 1

## Introduction

### 1.1 Purpose

This is the Requirement Analysis and Specification Document for the project software My Taxi Service. The goal of this document is to describe the system in terms of goals, requirements, assumptions and design, to define the different classes of users that will interact with it, and to analyze its most typical and most critical use cases.

This document is targeted to the customer's project managers, to evaluate the project specification from a high-level point of view, and to the intended designers, developers, programmers, analysts and testers, to build the actual software and to maintain, integrate and expand it in the future.

### 1.2 Scope

The aim of this project is to build a software system able to manage and optimize taxi services in a large city, targeted both to the city public transportation management council and to the citizens as customer of the taxi service. The system will allow passengers to use the service in a smart and accessible way, and at the same time will offer a better service thanks to an enhanced management of taxi resources and allocation.

In particular, passengers and taxi drivers will be registered and remembered by the system. Passengers will be able to request a taxi by using the application and taxi drivers will be notified of these requests by the system, that will divide the city in several taxi zones each with its own queue of taxis. Taxi drivers notified of passengers requests can then accept them.

### 1.3 Definitions, acronyms and abbreviations

- Request for service (RFS): A passenger uses the mobile app or web application to send the central management system a formal and well-formed request to be served by a taxi. This request includes a departure time, an itinerary starting point and ending point and the number of people to be served.

- Real time request: TODO;
- Reservation: TODO;
- Passenger: TODO;
- Logged passenger: TODO;
- Taxi driver: TODO;
- Logged taxi driver: TODO.

## 1.4 References

- Specification Document: My Taxi Service project specification AA 2015-16 – "Assignments 1 and 2 (RASD and DD).pdf".
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

## 1.5 Overview

This document is structured as follows:

- Introduction;
- Overall description;
- Specific requirements;
- System model: Main scenarios identification, UML models, Alloy analysis.

## Chapter 2

# Overall description

### 2.1 Product perspective

MY TAXI SERVICE as a software system will take the form of a web application running on a central server with access either through a standard browser or a specific mobile application (for passengers) or simply a mobile application (for taxi drivers) that will allow bidirectional communication, for which a standard HTTPS protocol will be used. Taxi drivers will need to own a smartphone with internet access. There is no software system already in place with which MY TAXI SERVICE will need to be integrated, as this is the case of a greenfield development to modernize a system where everything was previously done "by hand". A programmatic API will be part of the system, to allow for future expansion of its capabilities, such as taxi sharing management. No other specific or apparent constraints on existing hardware and software hold, as long as the system is able to support multiple connections and handle different ROS at a time.

### 2.2 Product functions

MY TAXI SERVICE will need to meet the following project goals:

1. Taxis in the city will be divided in queues and each one will be fairly managed.
2. Passengers will be registered and remembered by the system;
3. Passengers will be able to log in to their personal account;
4. Passengers will be able to request a taxi at their current position or at a position of their choice;
5. Passengers will be able to reserve a taxi in advance;
6. Passengers will be able to access their reservation history;
7. Passengers will be able to modify or delete elements from their reservation history;

8. Passengers will be notified if their request for service is accepted and the ETA of the incoming vehicle;
9. Passengers will be notified if their request for service cannot be met for whatever reason;
10. Taxi drivers will be registered and remembered by the system;
11. Taxi drivers will be able to log in to their personal account;
12. Taxi drivers will be able to inform the system of their availability;
13. Available taxi drivers will be notified when they are called to answer a request for service;
14. Taxi drivers will be able to confirm that they are going to take care of the assigned request for service;
15. Available taxi drivers will be notified by the system that they need to move to a different city zone to ensure a total coverage of possible passenger request;

## **2.3 User characteristics**

MY TAXI SERVICE is intended for two different classes of users: passengers and taxi drivers. As such, it will also offer two different and distinct views and UIs.

Passengers will require no special knowledge of the application to use it, for the UI must be designed to be as intuitive and easy-to-use as possible, in such a way that even a child could see it for the first time and understand how it works.

Taxi drivers will have access to different functions, but the core principle should well remain the same: there is no requirement for in-depth training (there should not be at all), for the application will simply relay informations and orders to and from the taxi driver and the system. Being a web-based application, passengers will of couse need internet access to use these services; taxi drivers, on the other hand, are required to own a smartphone with readily available and abundant internet access and, of course, the application installed.

## **2.4 Constraints**

### **2.4.1 Regulatory policies**

MY TAXI SERVICE will need to meet any and all requirements imposed by the laws and policies of the city in which it will be deployed, with special regards to the rules concerning taxi services. These will need to be studied in depth, with possible help from the customer's part, and integrated as needed into the management system.

### 2.4.2 Parallel operation

MY TAXI SERVICE central system will need to constantly monitor the situation of taxis in the whole city in order to dispatch and manage them as best as possible. At the same time, it will need to be sized appropriately to support the many different connections and operations from clients to the central server, both passengers and taxi drivers.

## 2.5 Assumptions and dependencies

1. Users profiles are private and hidden;
2. The system will minimize the risk of not being able to answer to a ROS due to lack of available taxis;
3. All taxi drivers own a suitable smartphone;
4. The mobile app will be available for each major mobile OS and will forward data to the central system in a uniformed way;
5. Taxis will be tracked by a GPS system;
6. The system will receive the GPS reading from each taxi and keep careful track of it;
7. The city will be divided in several zones;
8. All taxi drivers *must* own the MY TAXI SERVICE app and be registered users in order to work;
9. The system knows the license plates of all taxi drivers cars and their associations with the corresponding GPS tracker mounted on the vehicle.

## Chapter 3

# Specific requirements

### 3.1 Functional requirements

This section refers to system goals (specified in section 2.2). The goal referred to in the requirement description is the one it derives from.

1. TODO (Goal 1);
2. The system must provide a sign up function for passengers (Goal 2);
3. The system must store all required passenger information (Goal 2);
4. The system has to provide a log in function to access all passenger features (Goal 3);
5. The system must provide a function that allow logged passengers to require a taxi (Goal 4);
6. The system must provide a form in which the logged passenger will be able to add trip information for a request for service (starting point, destination point, number of passengers) (Goal 4);
7. The system must retrieve GPS information from the logged passenger's mobile phone (Goal 4);
8. The system must analyse the request and send a confirmation with the estimated waiting time if the request can be fulfilled (Goal 4);
9. The system must provide a reservation function (Goal 5);
10. The system must provide a form in which the logged passenger will be able to add trip information for a reservation (starting point, destination point, number of passengers, leaving time) (Goal 5);
11. The system must analyse the request and send a confirmation if the reservation can be fulfilled (Goal 5);
12. The system must finalise a reservation two hours before its requested time, making it unchangeable (Goal 5);



13. Reservation history must be stored by the system (Goal 6);
14. The system must provide a function which shows reservation history (Goal 6);
15. The system must provide a function to allow logged passengers to modify a reservation up to two hours before the requested time (Goal 7);
16. The system must provide a function to allow logged passengers to delete a reservation up to two hours before the requested time (Goal 7);
17. The system must analyse a newly-made request for service and send a confirmation with the estimated waiting time if the request can be fulfilled (Goal 8);
18. The system must analyse a newly-made reservation request and send a confirmation if the reservation can be fulfilled (Goal 8);
19. The system must notify passenger if his reservation cannot be met (Goal 9);
20. The system must delete the request in reservation history if it cannot be met (Goal 9);
21. The system must provide a sign up function for taxi drivers (Goal 10);
22. The system must store all required taxi driver information (Goal 10);
23. The system must check that a taxi driver that wants to sign up owns a valid license (Goal 10);
24. The system must check that the information needed for a taxi driver sign up points to a specific person in the company that has not already registered before confirming (Goal 10);
25. The system has to provide a log in function to access all taxi drivers features (Goal 11);
26. The system must provide a function to allow logged taxi drivers to inform the system of their availability (Goal 12);
27. The system must notify taxi drivers of an incoming request (Goal 13);
28. The system must provide all necessary information to taxi drivers (Goal 13);
29. The system must provide a function to allow taxi drivers to confirm that they are going to take care of the assigned request (Goal 14);
30. The system must provide an explicit *decline* function to allow taxi drivers to notify that they are not going to take on the assigned request (Goal 14);
31. The system must flag a taxi driver as having declined an assignment if no answer is received within 30 seconds (Goal 14);
32. The system must move a taxi driver that has declined an assignment at the end of their queue (Goal 14);
33. The system must analyse taxi locations and calculate their best possible distribution (Goal 15);

34. The system must choose which taxi drivers need to be moved to ensure total coverage of the city (Goal 15);
35. The system must notify taxi drivers in which city zone they have to move, as needed (Goal 15).

## **3.2 Non-functional requirements**

### **3.2.1 Performance requirements**

- The system must support a certain number of terminals for taxi drivers that will be at first estimated and then extracted from a statistical analysis of the system usage in its first months of life;
- The system must support a certain number of terminals for passengers that will be at first estimated and then extracted from a statistical analysis of the system usage in its first months of life;
- The system must support the simultaneous use of 50% of the registered passengers;
- The system must support the simultaneous use of 100% of the registered taxi drivers;
- Taxi drivers notifications must be received in less than 5 seconds.

### **3.2.2 Software system attributes**

Reliability is an important factor due to the fact that the system manages all taxis in the city and, were it to fall, the whole taxi circulation would be paralyzed. Portability must be considered top priority as the mobile application will need to run on all major mobile Operating Systems.

Security is less critical since no confidential client information are registered on or exchanged by the system.

## Chapter 4

# System model

### 4.1 Scenarios

#### 4.1.1 Scenario 1

Bob wants to go to the theater. He downloads the MY TAXI SERVICE app, he opens it and he finds the passenger sign up page. He inserts his name and surname, his address, and his personal citizen ID. After receiving confirmation of a successful operation, he logs in by tapping on the corresponding button. He finds three options: *call a taxi now*, *reserve a taxi for later*, *view reservation history*. He decides to tap the *call a taxi now* function. He finds a form where he inserts the destination address and the number of passengers. The system notifies him to turn his GPS on if he wants to call a taxi to his current location, otherwise he can manually insert a starting address. He taps *confirm* and the system answers by confirming back and sending him the estimated waiting time.

Charlie is a taxi driver already logged in the system, who has given his availability by tapping on the *driver available* button on his personal page in the app. He receives notification that he has been assigned to picking up Bob. He accepts by tapping on the big red *accept* button on his screen, he revs up and goes to the given starting address.

After arriving at destination, Charlie sends Bob on his way and notifies the system that he is available again. The system tells him in which zone of the city he will have to move to keep general widespread taxi coverage.

#### 4.1.2 Scenario 2

Tom wants to go to the airport at 4.30 AM. Since he is a very anxious person, one week in advance he logs to MY TAXI SERVICE, chooses the reservation function from his personal page and reserves a ride for the right day and time and destination and starting address. He is relieved in seeing the system has confirmed his reservation. Three days later, his anxiety growing, he thinks he should change the reservation in order to be at the airport one hour before what he had planned. He logs again in MY TAXI SERVICE and he chooses the *reservation history* function: the system shows him all of his previous reservations, which in this case is just the one he had made three days earlier. He taps the reservation entry and it pops out the option to *modify* or *delete* it. He carefully

taps on *modify* (heavens forbid he deletes it) and he is prompted to insert again all necessary information. He sighs in relief when the system confirms again that everything went well. Unfortunately, two hours before the scheduled arrival of his taxi (Bob is well awake, least he forgets something for the journey), the system sends him notification that his request cannot be fulfilled due to unexpected events arising outside the sphere of influence of the taxi company. Bob is somewhat taken aback, but, after all, he expected nothing less.

### 4.1.3 Scenario 3

Eve is a taxi driver. She downloads the MY TAXI SERVICE app and signs up as a taxi driver as per company policy. She is prompted to insert her name and surname, her address, her personal citizen ID, her license number and her cars plate number. The system checks that she is indeed a real, active taxi driver and that she is on the company payroll. After receiving confirmation of a successful operation, she logs in by tapping on the corresponding button.

After going to work one morning, she logs and signals her availability by tapping on the corresponding button on her personal page. Since its the first time in the day, the app firstly interfaces with her cars GPS tracker, and goes then in waiting mode. After a while, her app comes to life and notifies her that she has ten minutes to pick up a passenger at a certain place. She is busy having a fat cream *bombolone* for breakfast, so she declines.

The system receives Eves reject and moves her at the end of her queue. At the same time, he notifies the next-in-line.