Alessandro Pozzi 10423781

[Indirizzo posta elettronica]

Requirements Analysis and  
Specification Document

MyTaxyService

[1. Introduction 2](#_Toc434337718)

[1.1. Purpose 2](#_Toc434337719)

[1.2. Scope 2](#_Toc434337720)

[1.3. Goals 2](#_Toc434337721)

[1.4. Definitions, acronyms, abbreviations 3](#_Toc434337722)

[1.4.1. Definitions 3](#_Toc434337723)

[1.4.2. Acronyms 4](#_Toc434337724)

[1.4.3. Abbreviations 4](#_Toc434337725)

[1.5. Identify Stakeholders 4](#_Toc434337726)

[2. Overall Description 5](#_Toc434337727)

[2.1. Product perspective 5](#_Toc434337728)

[2.2. Product functions 5](#_Toc434337729)

[2.3. User characteristics 5](#_Toc434337730)

[2.4. Constraints 5](#_Toc434337731)

[2.5. Assumptions and Dependencies 5](#_Toc434337732)

[3. Specific Requirements 6](#_Toc434337733)

[3.1. The world and the machine 6](#_Toc434337734)

[3.2. Functional requirements 6](#_Toc434337735)

[3.3. Non Functional requirements 6](#_Toc434337736)

[4. Appendices 7](#_Toc434337737)

# Introduction

## Purpose

This document represents the Requirements Analysis and Specification Document (RASD). Its main aim is to describe the system to be developed in terms of what the software will do and how it will be expected to perform. Functional and non-functional requirements will be comprehensively defined, as well as constraints, assumptions and the system limitations. It will also include a set of use cases and graphical mockups that describe the interactions the users will have with the application, so that the commissioner will have a clear idea of the aspect and functionality of the final product. This document may also be used by developers and programmers in order to extend this system or integrate it with existing ones.

## Scope

The aim of this project is to model a software application whose intent is to provide an easy, comfortable and intuitive access to a city’s taxi service. The name of the application is MyTaxiService (MTS).

The front-end layer of the system is composed by a web application, dedicated exclusively to people requesting the service, and a mobile application, dedicated both to taxi drivers and customers. Customers and taxi drivers have access to different functionalities. Access to the actual service on both sides is granted only to registered users, meaning that a registration is required. The only functionality available for guests is to visualize the homepage.

The back-end of the application manages the taxis’ distribution around the city via GPS information and the forwarding of incoming requests to near taxis via a queue policy based on a taxi zones division. It also manages the forwarding of notifications to the users.

*MTS functionalities on the customer side:*

MTS allows customers to request a taxi in a specific location. In such case, the user is given a code and a waiting time estimated by the system. MyTaxiService also allows customers to make a taxi ride reservation for a specific time, origin and destination. The reservation must be done at least two hours before the given time, otherwise the system will send a notification about the rejected reservation’s request.

*MTS functionalities on the taxi driver side:*

MTS allows the user to communicate his availability. If the user is available, the system can send him notifications about incoming requests. The user may or may not confirm his will to take care of the request.

## Goals

[G1] Allow customers to access the system’s taxi service in any moment, whether they are at home or anywhere else in the city.

[G2] Allow customers to request a taxi ride from an arranged location.

[G3] Allow customers to reserve a taxi ride at a specific time with a given origin and destination.

[G4] Allow taxi drivers to answer a ride request and take care of customers.

[G5] A ride request should always be satisfied within a considerable short amount of time, 15 minutes on average.

[G6] Allow customers to be notified of any relevant update connected to their requests and reservations.

[G7] Customers and taxi drivers must be able to contact each other after the system has paired them.

[G8] Allow customers to cancel requests and reservations.

[G9] Administrators must be able to assign an account to the taxi drivers hired by the company.

## Definitions, acronyms, abbreviations

### Definitions

*Customer* –Registered user that may require a ride

*Taxi* *Driver* – Employee of the government’s taxi service

*Reservation* – A ride that has been reserved by a customer through the web or mobile application.

*Request* – A customer’s demand to be picked up by a taxi, sent using the web or mobile application.

*Taxi* *zones* –2km2 areas in which the city’s territory is divided.

*Taxi driver availability* – The status of a taxi driver: if he is available then he can receive a ride request, otherwise no requests can be sent.

*Customer and taxi driver pair* – Sometimes taxi drivers and customers are said to be “paired”: this happens only after a requests or 10 minutes before a reservation, i.e. when the system choose a taxi driver to serve a customer. This association ends when the taxi drops the customer to the requested destination.

These are the 2 possibile status of a taxi driver:

*Available* – When the driver has no customer to pick up and it’s ready to accept new requests.

*Busy* – When the driver is taking care of a request or of a standard customer. Note that the taxi driver is considered “busy” even if he hasn’t pick up the customer yet, but has only accepted his request.

These are the possible status of a taxi ride (request o reservation):

*NotAssigned* – This is the default status of a taxi ride requested by customers. It simply means that the request (or reservation) has been accepted by the system but nothing else has been done yet.

*Assigned –* Appears when the system has assigned a taxi to the customer request.

Completed – When the taxi driver has brought the customer to destination and has declared himself “Available”.

*Annulled –* When the customer has deleted the request (or reservation).

*Standard customer* – A customer who is not using myTaxiService’s applications but may still require a ride with the “traditional” method.

### Acronyms

MTS - MyTaxiService

### Abbreviations

[Gn]: n-goal.

[Rn]: n-functional requirement.

[Nn]: n-non functional requirement.

## Identify Stakeholders

Our main stakeholder is Prof. Raffaella Mirandola of Politecnico di Milano, DEIB. She gave us the delivery of the project. Other hypothetical stakeholders interested in the service offered by myTaxiService are:

* The government of the city, which has come up with the idea to improve the taxi service for its citizens.
* Citizens and tourists, who are the main users of the application.
* The taxi drivers, who are a small subset of users of the application.
* The taxi company, which of course provides the taxis.
* Other entities indirectly touched by the service, such as airports, hotels etc.

# Overall Description

## Product perspective

The applications we will release is not completely independent and self-contained: it will rely mainly on the GPS system in order to retrieve the position of all its taxis. Moreover, we will promote possible future integrations and extensions by releasing a set of APIs. We hope that MyTaxyService may be exploited by other companies and developers and integrated with similar services (like journey planners or hotel booking services) to make travelling easier. APIs will also provide an easier development of additional functions, like taxi sharing or a SMS-based reservation system.

## User characteristic

MyTaxiService will be used by two types of users: taxi drivers and customers. In both cases, no particular skills or information technology knowledge are required, since our application will be easy to use and accessible by any mobile phone or computer that have access to internet.

## Constraints

* *Regulatory policies*

MyTaxiService doesn’t have to meet any regulatory policies.

* *Hardware limitations*
* *Interfaces to other applications*

MyTaxiService doesn’t have to meet any interface to other applications.

* *Parallel operations*

MyTaxiService system must support parallel operations and multiple connections in order to provide a smooth experience and a correct functioning.

## Assumptions and dependancies

* Requests and reservations can be cancelled if and only if no taxi have been assigned to the customer.
* Customers and taxi drivers access to the same mobile app. The available functions will depend by the login phase, which determines the type of user connected.
* A taxi driver account cannot be used as a customer account, and vice versa. This means that if a taxi driver wants to access the customer’s services, he will need to create a customer account.
* Taxi drivers will manually update their status (available or busy) with the mobile app every time they pick up or drop off customers.
* Taxi drivers have a fixed amount of time of 1 minute to answer a ride request. If the answer is not received, the system will consider the request as refused and act accordingly.
* Customers are allowed to perform unlimited taxi reservations. However, due to the fact that overlaps between two reservations are not predictable, the system will cancel any impracticable reservation identified at runtime.
* Customers are not allowed to request a taxi ride if any other request made by the same account has not been fulfilled yet.
* If and only if a taxi zone does not have any taxi available to answer a request, the system will search for an available taxi in adjoining zones. Worst case scenario: if there are no taxi available in the adjoined zones, the costumer should be notified and put in hold. During this period of time the costumer should be allowed to cancel the request.
* Taxis are assumed to be well distributed in the city’s taxi zones. This means that taxi zones never happen to be completely empty and the situation in which no taxi drivers are present, either available or busy, cannot occur.
* Payment and specific duties related to the taxi service are not considered and managed by the application. MyTaxiService is meant to be only an interface between customers and taxi drivers.
* The web and mobile registration is intended for customers only. Taxi drivers’ account are created by an administrator when they are hired by the taxi company. Taxi drivers will then receive their username and temporary password, which they will be able to change once logged in the application.
* The system will not accept reservation whose date is latter than a year in respect to the submission date.
* Taxi drivers can log into the web application with theirs driver’s account but cannot access to the customers’ service. The only available function will be the modification of some account information.
* Taxi’s code identifier is showed somewhere on the car, where customer can easily spot it.

## Future possible implementations

* The implementation of a taxi sharing service, which allows customers to share their ride with other users in order to decrease the cost of the ride.
* The implementation of a SMS system, which allows users to request or reserve a ride by sending a SMS. SMS syntax rules may be available online or may be sent by request with a SMS. This function may extend the service to those who does not have always access to an internet connection, like tourists.
* A function that estimates how much time a ride will take and how much it will cost. This function may be accessible even without requesting a ride.
* An improved taxi management system, with redistributes available taxis moving them from high density areas to low density areas.
* The possibility to show the GPS position of the customer’s taxi on an interactive map.

# Specific Requirements

## External interface requirements

Below are shown some mockups that shows the general aspect, structure and main functionalities of the mobile and web application. Note that these mockups are only indicative, and the final version of the application may differ.

### Mobile Application

*Home -* This is the layout of the main screen of the mobile application:



*Registration -* If the user taps on the “Register!” button from the application home, the following screen will appear:



In the final version some of the fields may be marked as compulsory.

*Taxi Driver Home –* If a user logs in with taxi driver’s credential from the application home, the following screen will appear:



The “See request” button, as long as the “You have a new ride request” text, may not be present if no request have been received.

*Taxi driver request details –* The following mockup shows what the user will see if he presses the “See request” button from the previous screen.



Note that the “Destination” field may be empty if the customer has not compiled it while making the request. If the user presses the “REFUSE” button, the *Taxi driver home* will appear, otherwise the following screen will be loaded:

*Taxi driver busy status –* This screen may be reached either by pressing the “ACCEPT” button in the *Taxi driver request details* or by setting the busy status in the *Taxi driver home*.



No customer information will appear if the taxi driver picks up a standard customer.

*Customer home –* This mockup shows the screen that will appear after a customer has finished the login procedure:



*Customer ride request –* This screen shows what happens when the “Request a ride” button is pressed in the *Customer home*.



The asterisk indicates that the Origin field is compulsory, while the localization symbol indicates the possibility to locate the origin using the phone’s GPS system (if present).

*Customer ride reservation -* This screen shows what happens when the “Reserve a ride” button is pressed in the *Customer home*.



*Customer history –* After pressing the “Show your history” button, the following screen will appear:



*Customer Request/Reservation Info -* By pressing the Info symbol in the *Customer history* screen, details about the request or reservation will be shown as follows.



The “Waiting time” and “Code” box will be filled with the time to wait for the taxi to arrive and the correspondent taxi code only when a taxi driver will be assigned to the customer.

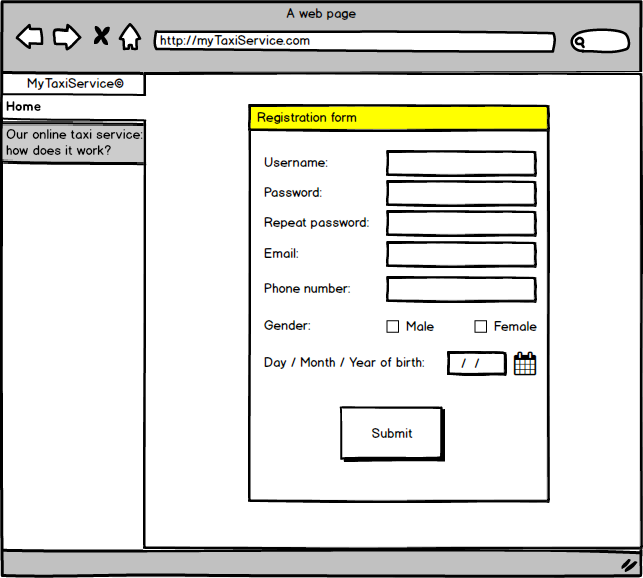
### Web Application

*Home –* This is how the home web page of MyTaxiService should appear:



On the left bar there is a possible clickable tab that may explain the service functionalities.

*Registration form* – This will appear when the “Register now” button is pressed in the home page:



*Customer home –* When the user has logged in, the following page will load:



Notice that a taxi driver user won’t have any of these functionality available, except for “Manage your account” and “log out”. “Notifications” is reserved to customer and allows them to be alerted if some events occurs.

*Customer ride request –* The following form will appear when “Request a ride” is pressed:



*Customer ride reservation –* The following form will load when the “Reserve a ride button” is pressed:



*Customer history* – This page will appear when the “Show your history” button is pressed.

Notice that the “Status” fields will be explained by clicking the info buttons. “Taxi code” and “Waiting time” columns will present a value only when the status field is “Taxi assigned”. Finally, the “Delete” icon will be clickable only if a request/reservation has no taxi assigned and has not been competed yet.

## Functional requirement

*[****G1****] Allow customers to access the system’s taxi service in any moment, whether they are at home or anywhere else in the city.*

[R1] Customers should be able to access the service through both the web and the mobile application, even at the same time.

[R2] Customers must be able to register to the taxi service from the mobile or web homepage.

[R3] Only registered customers can access MyTaxiService’s services.

[R4] The system should allow the log out functionality.

*[****G2****] Allow customers to request a taxi ride from an arranged location.*

[R1] Only registered customers can request a taxi ride.

[R2] Customers must insert a valid origin location in order to request a ride.

[R3] The system will not allow more than a request if the previous one has not been accomplished yet.

*[****G3****] Allow customers to reserve a taxi ride at a specific time with a given origin and destination.*

[R1] The system should allow taxi reservations for a specific path communicated by the customer.

[R2] The system must not allow overlaps between reservations made by the same customer.

[R3] The system should allow to delete a taxi reservation made by the same customer.

[R4] The system allows reservations only 2 hours before time and date specified by the customer.

[R5] The system will assign a taxi driver for the reserved ride 10 minutes before the time and date specified by the customer.

*[****G4****] Allow taxi drivers to answer a ride request and take care of customers.*

[R1] Taxi drivers should be able to communicate their current availability state to the system.

[R2] If available, taxi drivers should be able to receive incoming requests.

[R3] After receiving an incoming request, the taxi driver should be able to either confirm or not his intention to take charge of the request.

[R4] Taxi drivers must be able to log in the mobile application with preassigned credential and be identified as drivers.

*[****G5****] A ride request should always be satisfied within a considerable short amount of time, 15 minutes on average.*

[R1] The system should always search an available taxi giving maximum priority to the taxi zone related to the request and lower priority to the immediate near zones. Any other taxi zone should be ignored.

[R2] If no taxis are available in the zones specified in the previous requirement, the system should put the request on hold and periodically check again the taxi availability.

*[****G6****] Allow customers to be notified of any relevant update connected to their requests and reservations.*

[R1] The system should send updates through email and/or in-app notification, as specified by the customer.

[R2] Absence of taxis available, reservations overlaps, taxi average waiting time and taxi assigned to customers are events that must be notified to the customer.

*[****G7****] Customers and taxi drivers must be able to contact each other after the system has paired them.*

[R1] Customers must leave a valid phone number in order to complete the registration phase.

[R2] Taxi drivers must be able to access to the customer’s phone number when the system has paired them.

[R3] Customers must receive the taxi drivers’ contact number after the system has paired them.

[R4] Customers must receive the taxi code in order to be able to recognize its driver.

*[****G8****] Allow customers to cancel requests and reservations.*

[R1] Customers can cancel a request or reservation only if it has not been assigned to a taxi driver yet.

*[****G9****] Administrators must be able to assign an account to the taxi drivers hired by the company.*

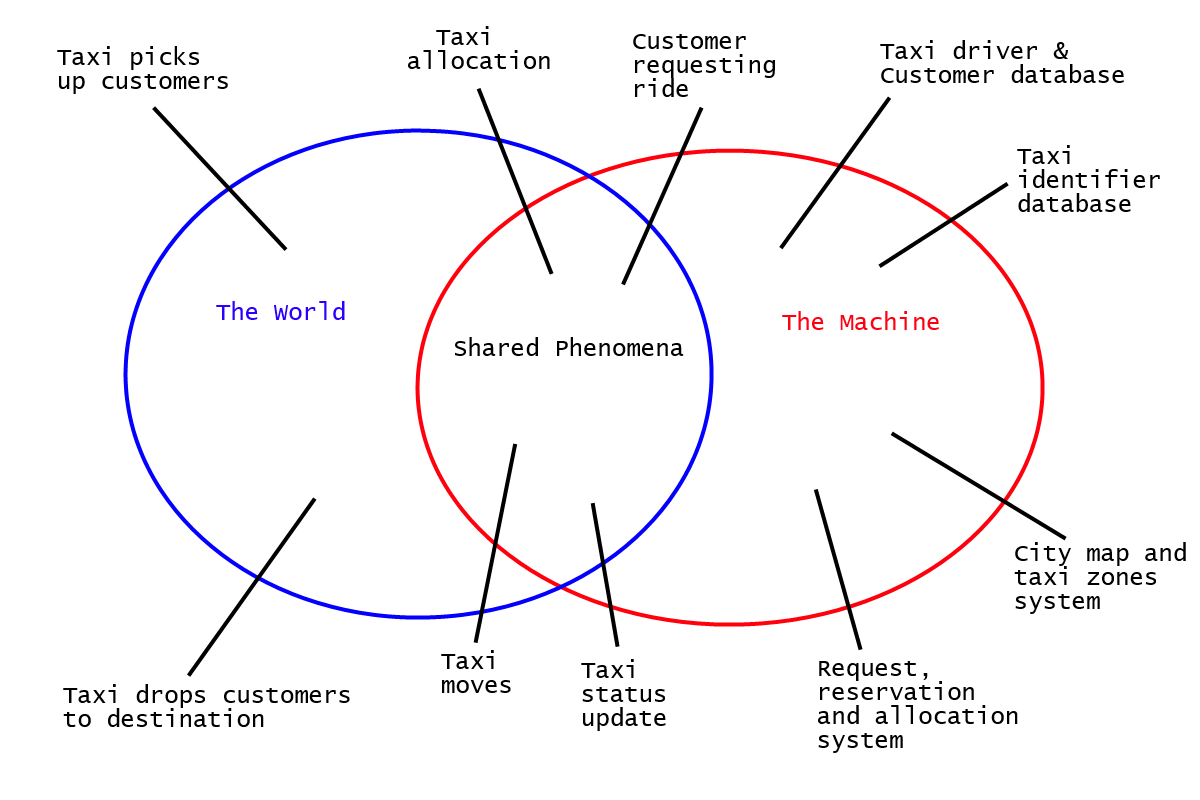
[R1] Administrators must be able to create a taxi driver’s account.

[R2] Administrators must be able to cancel a taxi driver’s account.

## The world and the machine

The model proposed by M. Kackson & P. Zave, “The World & the Machine”, allow us to analyze the domain of MyTaxyService application and its relation with the world. This model contemplate the presence of two areas: *The Machine*, which is the portion of the system to be developed, and *The World*, which is the portion of the environment that is affected by the machine. The intersection of the two is called *Shared Phenomena* and is composed by all the entities that are controller by only The World or The Machine, and observed by the other.

The following diagram shows MyTaxiService’s *The World and The Machine* analysis.



**Entities analysis:**

***The world***

“Taxi picks up customers” and “taxi drops customer to destination” are the only two phenomena that happen exclusively in the world. In fact, the machine can observe them only indirectly by monitoring the status of the taxi and its position.

***The Machine***

The machine is composed by the “Request, reservation and allocation system” which manages the main application logic, and is strictly dependent upon the “City map and taxi zones system”, the “Taxi identifier database” and the “Users database” entities.

***The Shared Phenomena***

“Taxi moves” is a shared phenomenon, which is controlled by the world and observed by the machine through the GPS system. “Taxi status update” is also a phenomenon controlled by the world (i.e. the taxi driver) and observed by the system.

“Taxi allocation” is observed by the world and controlled by the machine, which sets the taxi’s next destination accordingly to the customer requests. Instead, “Customer requesting ride” happens in the world and is only observed by the machine.

## Non Functional requirements

1. MyTaxiService’s system should be available 24/7.
2. The waiting time for a taxi to come that is showed to customers should be updated every 30 seconds.
3. Every functionality offered by the mobile or web application after the login should be reachable within three clicks.
4. Mobile applications’ button available to taxi drivers should be big enough to be easily recognized and pressed while driving.
5. Code should be well documented in order to facilitate extendibility.
6. Email sent to the user in order to confirm registration should arrive in less than 10 minutes.
7. Web and mobile applications should have a similar graphics so that the correlation between the two would be immediately identified by customers.
8. Mobile’s notifications should appear even when the user has not the app opened.
9. The description of the mobile application in the phone store should point out clearly and briefly its functions, allowing a quick comprehension from any type of user.
10. MyTaxiService’s applications should always have imperceptible response time during the navigation within its forms.

# Appendices