

**R**equirements **A**nalysis and

**S**pecification **D**ocument (**RASD**)

Computer Science and Engineering (CSE)

Software Engineering 2 Project

Year 2015/16



**STUDENTS:**

*Martino Andrea (****788701****)*

*Marchesani Francesco (****852444****)*

**PROFESSOR:**

*Mirandola Raffaela*

1. Introduction 3

1.1. Purpose of the requirements model 3

1.2. RASD Approach: “The world and the machine” 4

1.3. myTaxiService: main goals 5

1.4. Current state of the service and future prospect 6

1.5. Definitions, acronyms and abbreviations 7

1.6. References 8

2. Overall description 8

2.1. Product perspective 8

2.2. User characteristics 9

2.3. Constraints 9

2.4. Assumptions and Dependencies 9

2.5. Future possible implementations 10

2.6. Stakeholders identification 10

3. Specific requirements 11

3.1. External interface requirements 11

3.2. Functional Requirements 11

3.3 Scenarios 13

3.4 Non-functional requirements 15

# Introduction

## Purpose of the requirements model

The main purpose of this **RASD** (*Requirements Analysis and Specification Document*) is to examine in depth the phases of analysis and specification of the project requirements.

The project name is *myTaxiDriver*, which is the **Software Engineering 2 project** of year 2015/16 at **Politecnico di Milano**.

The reference model used in this project is **IEEE/ANSI 830-1998**. This is one of the most widely known requirements document standard. It is important to underline that the specifications of this document may evolve in the future (this may occurs for several causes).

Anyway, we will try to maintain coherence with this document in the next steps as much as possible.

## RASD Approach: “The world and the machine”

Identify the right requirements may be a difficult thing to do if the approach is not good enough. The main thing to understand is the link between what happens in the real world (*The World*) and the software technologies (*The Machine*). This link is Requirements Engineering.

**Requirements Engineering**

**Software**

**Technologies**

**Real World**

**Demands**

The approach followed in this document is known as “*The world and the machine*”. This one is the approach defined by Michael Jackson and Pamela Dave. There are two main entities in this approach:

* **The World**: part of the real World that interfaces with the software to be and which is influenced by him.
* **The Machine:** part of the software to be. That is the union of the developed software and the hardware where software will be executed.

***D****-omain properties/assumptions*

**Shared phenomena**

***R****-equirements*

***G****-oals*

**The Machine**

**The World**

## myTaxiService: main goals

According to “*The World and The Machine*” model, *myTaxiService* project has the following goals:

* **[G1]** Allow a user to make a taxi reservation using the mobile application.
* **[G2]** Allow a user to make a taxi reservation using the web application.
* **[G3]** Grant the possibility to delete a reservation using the mobile application.
* **[G4]** Grant the possibility to delete a reservation using the web application.
* **[G5]** Reduce costumer waiting time. // revise
* **[G5']** Maximize taxi driver down-time. // revise
* **[G6]** Give the possibility of taxi reservation to the user <<- GOAL //->>SONO REQUISITI, DA GESTIRE LA POSSIBILITA' DI PRENOTARE IN ANTICIPO (feature dei gruppi da 2 persone, come me e te), MA ANCHE @LIVE after the specification of the start and the end of the ride (two hours or more before the ride).
* **[G7]** Allow customer to pay the ride to TAXISPA.
* **[G8]** Allow the taxi driver to reach the right costumer destination.
* **[G9]** Allow the taxi driver to reach the costumer position.
* **[G10]** Notify the taxi driver when there's a ride possiblity.
* **[G11]** Allow the user to check the current price of a ride.
* **[G12]** Allow the taxi driver to see costustumers' requests of ride.
* **[G13]** Notify a costumer with the taxi driver's response.
* **[G14]** Allow a customer to see his/her current position in a ride.

## Current state of the service and future prospect

### SYSTEM AS IS

Taxi drivers are equipped with a cellphone and an earpiece to be able to answer calls during driving.

Taxis are equipped with a proprietary device placed on the cockpit that periodically sends GPS information to TAXISPA using GSM connection and acts as a taximeter.

Currently if a user wants to use a taxi he/her must call TAXISPA phone number and provide his/her position with accuracy, if possible. Every call is redirected from a switchboard system to an available employer that takes care of the customer. The employer watches a computer screen that shows every taxi driver location over a map to decide which one could be available to take the call.

Then the employer puts on hold the customer and calls the driver to check his/her availability and report the response to the customer.

Currently is not possible to reserve a taxi before the very same day.

### SYSTEM TO BE

*myTaxiService* will be built from scratch. This new product is not a specific evolution of the existing system.

*myTaxiService* aim to provide new ways of organizing work efficiently to ensure an always growing customer-base and quality of service, reduce the total operating costs of TAXISPA and put TAXISPA in a stronger competitive position. myTaxiService will use mobile and web technologies and will add new features.

## Definitions, acronyms and abbreviations

* **RSA**: one of the first practical *public-key cryptosystems*. It is widely used for secure data transmission. It takes the name from the algorithm’s inventors (Rivest, Shamir, Adleman).
* **TAXISPA**: big taxi society that wants to develop *myTaxiService*.
* **QoS**: *Quality of Service* is the overall performance of a service (especially from the users’ point of view).
* **2PL**: *Two-Phase Locking* is a concurrency control method used in the most recent databases with transaction processing.
* **Timestamp**: A *timestamp* is a sequence of characters or encoded information identifying when a certain event occurred. It can be used with the 2PL in DBMS to improve the concurrency control efficiency.
* **DBMS**: A *DataBase Management* System is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data.
* **GSM:** Global System for Mobile Communications, default global standard for mobile communications.
* **GPS:** Global Positioning System, satellite-based navigation system.
* **ADS**: Also known as Advertising is a form of marketing communication used to promote or sell something, usually a business's product or service.
* **Queueing Theory**: is the mathematical study of waiting lines, or queues.
* **M/D/k**: In *Queueing theory*, a discipline within the mathematical theory of probability, an M/D/k queue represents the queue length in a system having k servers, where a Poisson process determines arrivals and job service times are fixed.
* **UI:** User Interface, the space where interactions between humans and machines occur.
* **Internal functionality:** Functionality is not directly available to the final user but only to system administrator.
* **User**: an abstract class of interest with two subclasses: *Customer* and *Taxi Driver*.

## References

This document was produced by faithfully following the directives contained in the **IEEE/ANSII 830-1998** (as we said in the chapter *1.1*).

It also revealed and very useful to consult some of the RASD presented over the previous academic years, trying to identify critical issues, patterns and isolate sections developed in an accurate, thorough and organic way.

Here are the documents used as reference:

* M*. Jackson, P. Zave, "Deriving Specifications from Requirements: An Example", Proceedings of ICSE 95, 1995*
* *M. Jackson, P. Zave, "Four Dark Corners of Requirements Engineering", TOSEM, 1997*
* *B. Nuseibeh, S. Easterbrook, "Requirements Engineering: A Roadmap", Proceedings ICSE 2000*
* *M. Jackson, Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices, ACM Press Books, 1995*
* *830-1998 IEEE/ANSII Recommended Practice for Software Requirements Specifications,*

*http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=720574&tag=1&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs\_all.jsp%3Farnumber%3D720574%26tag%3D1*

* *Various projects of the past years (from the* ***Beep*** *platform)*
* *Slides of the course by Prof. Raffaela Mirandola*

# Overall description

## Product perspective

Both the mobile application and the web application of *myTaxiService* will be new products of *TAXISPA*. In fact, there will not be an integration with *TAXISPA*’s legacy systems. There will be two different version of *myTaxiService*: the first one for the user and the other for the taxi driver. These versions will have different features and views, as it is possible to imagine.

*Note*: the taxi driver will only have access to the mobile app service (in his case, the web application version is almost useless).

## User characteristics

The use of *myTaxiService* will be easy enough to allow a big number of people to use it. In fact, no special skills are requested. Users must be only able to use the service via mobile application or web application. There is not a “target age” of users: everyone is a potential user. Anyway, there is a limitation for children under 16 years (without a special permission of the parents/tutors).

## Constraints

### Regulatory policies

People under 16 years can use the service only with a special authorization from their parents or tutors.

### Hardware limitations

*myTaxiService* doesn’t have hardware limitations.

### Interfaces to other applications

There are not interfaces between *myTaxiService* and other applications.

### Parallel operation

Parallelism is very important for *myTaxiService* service. We attend many requests: so parallel processing and dynamic queue management are crucial.

Because of this fact, the system supports parallelism and simultaneous transactions according to the latest technologies in this field. For example, the **DBMS** uses *2PL + Timestamp* for the concurrency control.

## Assumptions and Dependencies

**Assumptions** and **Dependencies** of *myTaxiService* are in the following list:

* There are only two types of account: standard user account and taxi driver account.
* Now there is not dependency between different users. Otherwise, taxi sharing may be an idea for the future.
* A user can have also access to the web application version using a browser (like *Chrome, Safari, Firefox*…).
* A user can do limitless reservations for future events.
* If the system is down for problems or updates there is a special alert message in the app.
* If a user deletes a reservation, he can do again the same reservation in the future if necessary without particular problems.
* There are not ads both in the web and in the mobile application.
* There are not relevant differences from the functional point of view between the mobile application and the web application.
* The queue management model is based on an efficient *M/D/k* application of *Queuing Theory*.
* When a user is in the taxi cannot delete a reservation.
* According to a user request, a taxi driver can change destination while he is driving.

## Future possible implementations

* At launch mTS mobile application will only be available for iOS and Android. We will try our best to release Windows Phone version before the end of the year.
* Public API will be accessible to the public 2 months after the release of the product. Documentation and SDK/emulator will be available at launch for independent developers.
* mTS mobile application will allow customers to pay through phone credit and In-App Purchase (iOS/Android).
* We will create mobile version of the app to reach old mobile devices which can’t support our app.

## Stakeholders identification

It is possible to distinguish between two categories of stakeholders of *myTaxiService*:

**Internal Stakeholders**

* Taxi drivers of *TAXISPA*
* Mangers of *TAXISPA*
* Employees of *TAXISPA*
* Other personnel of *TAXISPA*

**External Stakeholders**

* Costumers
* Sponsors
* Direct Competitors
* Un-direct Competitors
* App developers
* App testers

# Specific requirements

Identify the right requirements may be a difficult thing to do if the approach is not good enough. The main thing to understand is the link between what happens in the real world (*The World*) and the software technologies (*The Machine*). This link is Requirements Engineering.

## External interface requirements

To maintain this document as clean as possible we have decided to add UI mockups in another file bleb le ble

AGGIUNGERE FILE

## Functional Requirements

// Maybe with the Goal-Reqs structure,DISCUSS

### Registration of a user to the system

* The system has to provide sign up functionality to every customer.
* The system has to provide system admins of an internal sign up functionality

to register taxi drivers.

### Login of a user to the system

* The system has to provide a login mechanism for user.
* The system has to provide system admins of an internal login mechanism to

login drivers.

### Make financial transactions

* The system has to provide a mechanism to allow customers to send their money

to TaxiSPA.

.

### Route taxi driver to certain location

* The system must be able to send driving directions (to a certain location) to taxi drivers.

### Create/delete/update taxi reservations

* The system has to provide a functionality to allow a customer to create a taxi reservation.
* The system has to provide a functionality to allow customer view taxi reservations.
* The system has to provide a functionality to allow a customer to delete taxi reservations.
* The system has to provide a functionality to allow a customer to update taxi reservations.

### Send information about travels/reservations to a customer

* The system will be able to notify users about their reservations.
* The system will be able to update users during their travels in mTS taxis.

### Accept/Decline system request to take care of a customer

* The system will allow taxi driver to accept a new customer.
* The system will allow taxi driver to decline a new customer.

### Request/acquire user GPS position

* The system will provide a way to acquire periodically taxi drivers GPS position.
* The system will provide a way to request customers GPS position (privacy concerns, user can deny).

## Scenarios

### SCENARIO 1

Sergey Brin is planning to hang out with other colleagues of Google this Friday after work. However, there is a problem: Friday nights are famous for the traffic jams. He is a very precise person and he prefers to book a taxi in advance to reach the destination. Therefore, he decides to login to his account at MyTaxiService website and insert all the data for the taxi reservation. He selects one taxi, only for him, near Google Maps research center in Mountain View at 8.30 pm. After the submission of the data, the system confirms Sergey’s reservation without problems. The request is stored in the system’s database.

### SCENARIO 2

Mark Zuckerberg is waiting his taxi in front of Facebook Park in the Silicon Valley: he is finally going to meet the famous Russian Professor Markesanskjy. However, while he was attending his guest, he receives a WhatsApp message from the Professor. The Professor is still not available for the meeting… so Mark Zuckerberg can come back in the park. While he is walking, he realizes that has to delete the taxi reservation. Therefore, he immediately opens the mobile app of *myTaxiService* and delete the reservation. The system sends him an acknowledge confirmation.

### SCENARIO 3

Bill Gates wants to buy a new big house with many rooms to earn a Guinness World Record award. In order to visit the big house, he is attending his taxi, but in the meantime a crash occurs and his booked taxi cannot reach the location. Fortunately, the system provides Bill an alternative taxi (he is an important and rich client!) and send him a notification with the announcement of the possible delay, of course!

### SCENARIO 4

Robert De Niro is a young taxi driver that has just left a passenger in Martin Scorsese Road, the destination place. Therefore, he signals again his availability to the system. The system puts him in the local area queue. While Robert is waiting for another call, he can drink something (non-alcoholic of course!) with his old girlfriend Jodie Foster.

### SCENARIO 5

Night: 11 pm. Robert De Niro is driving his taxi, carrying two passengers, on the 1st Avenue. He is going to reach the destination in about five minutes. In the meantime, the system signals him another call from the 2nd Avenue (very close to the first one, as it is possible to imagine). He should reach the 2nd Avenue in about ten minutes, because he is a very good and fast taxi driver. Therefore, he confirms with his earphones. The system sends a notification to the client. It seems that Robert is very lucky this night… he will earn a lot of money!

SCENARIO 6

Marty McFly is a student from California. At 1.30 a.m., he receives a phone call from his friend: the professor Emmet Brown alias “Doc”. The professor wants to see Marty as fast as possible: the question is very important! Thus, Marty immediately picks up his smartphone and opens *myTaxiService* mobile application. Unfortunately, no taxi is currently available in his area. How to solve this problem? Just waiting a few minutes: the service will provide a taxi as soon as possible. No worries Marty, you will see Doc soon!

## Non-functional requirements

### 3.4.1 Performance requirements

The system will send the notification of taxi availability time after the user’s reservation almost tree minute after the reservation. The taxi driver on the top of reservation’s queue will have one minute to accept the user reservation. If the taxi driver declines the reservation, the system puts him in the end of the queue and sends the notification to another taxi driver (the following in the queue). The percentage of *crash for usage* in the mobile app will be under 2% for the mobile app and under 1% for the web app.

### Availability

*myTaxiService* motto is “*Opportunity Unlimited*”, so the system will be always 24 hours per day and 7 days a week (*full time*) in standard days. In case of updates, the system will be down only form *3 am* to *5 am* (when the number of users’ requests is small). To guarantee the availability the company will also buy powerful servers (a server farm, in fact).

### Maintainability

The application’s code will follow the principles and standards of “*good programming*” (right commenting, clean and simple coding, design patterns using and so on). The full documentation of *myTaxiService* will be stored in TAXISPA. With these precautions, new developers of the service will know how the system works in detail in order to ensure an optimal maintainability.

### Portability

Thanks to the mobile application support, the service will easily run on millions of devices. In fact, the mobile application will be compatible with a big amount (*>=90%*) of devices with Android and iOS (the most common operative systems in mobile devices, like smartphones and tablets).

### Scalability

It is important to consider the scalability factor. Possibilities of new modules may be:

* *Taxi Sharing Service:* that allows two or more users to share a taxi and save a little amount of money.
* *Ride Review System:* the user has the possibility to evaluate the ride and the taxi driver (like a feedback).
* *Other modules:* to be defined in the future.

### Security

Sensible data will be stored in a database with firewall (in both hardware and software) protection. Another crucial aspect is the payment service via app: this will happen according to an asymmetric *256-bit RSA cryptography system* (in order to gain protection in the transaction).