

**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*

**Index**

1. **Introduction**
   1. **Purpose of the requirements model**
   2. **RASD Approach: “The world and the machine”**
   3. **myTaxiService: main goals**
   4. **Current state of the service and future prospect**
   5. **Limitations of the product and new features possibilities**
   6. Definitions, acronyms and abbreviations
   7. References
2. **Overall description** 
   1. **Product prospective**
   2. **User characteristics**
   3. **Constraints**
   4. **Assumptions ad dependencies**
   5. **Future possible implementations**
3. **Specific requirements**
   1. **External interface requirements**
   2. **Functional requirements**
   3. **Scenarios**
   4. **UML models**
   5. **Non-functional requirements**
4. **Alloy and other information**
   1. **Alloy**
   2. **Software and Tools used**
   3. **Hours of work**

**1. Introduction**

**1.1 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.

* 1. **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**

* 1. **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]** Grant a fair management of the taxi queues.

**[G6]** Give the possibility of taxi reservation to the user, after the specification of the start and the end of the ride (two hours or more before the ride).

* 1. **Current state of the service and future prospect**

**1.4.1 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.

**1.4.2 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.

* 1. **Limitations of the product and new features possibilities**
  2. **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.
* **GSM:** Global System For Mobile Communications, default global standard for mobile communications.
* **GPS:** Global Positioning System, satellite-based navigation system.
* **UI:** User Interface, the space where interactions between humans and machines occur.

**1.7 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*
* *Some articles on Wikipedia*

**2. Overall description**

**2.1 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).

**2.2 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).

**2.3 Constraints**

**2.3.1 Regulatory policies**

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

**2.3.2 Hardware limitations**

*myTaxiService* doesn’t have hardware limitations.

**2.3.3 Interfaces to other applications**

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

**2.3.4 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.

* 1. **Non-functional 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.

**3.5.1 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.

**3.5.2 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).

**3.5.3 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.

**3.5.4 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 (*>=95%*) of devices with Android, iOS and Windows Phone (the most common operative systems in mobile devices, like smartphones and tablets).

**3.5.5 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.

**3.5.6 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).

**3.5.7 User Interface**

To maintain this document as clean as possible we have decided to show UI mockups in another document named blabla.