



**POLITECNICO**  
MILANO 1863

13 November 2016

# **PowerEnJoy**

## Requirements Analysis and Specifications Document

Blanco	Federica	875487
Casasopra	Fabiola	864412

*Software Engineering 2 Project*  
2016/2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Actors . . . . .	2
1.4	Goals . . . . .	2
1.5	Definitions, acronyms, abbreviations . . . . .	3
1.5.1	Definitions . . . . .	3
1.5.2	Acronyms . . . . .	4
1.5.3	Abbreviations . . . . .	4
1.6	Overview . . . . .	4
1.7	Reference Documents . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product perspective . . . . .	6
2.2	Product functions . . . . .	6
2.3	User characteristics . . . . .	6
2.4	Constraints . . . . .	6
2.5	Assumptions and Dependencies . . . . .	7
<b>3</b>	<b>Specific Requirements</b>	<b>8</b>
3.1	External Interface Requirements . . . . .	8
3.1.1	User Interface . . . . .	8
3.1.2	Hardware Interfaces . . . . .	8
3.1.3	Software Interfaces . . . . .	8
3.1.4	Communication Interfaces . . . . .	8
3.2	Functional Requirements . . . . .	8
3.3	Performance Requirements . . . . .	9
3.4	Design Constraints . . . . .	9
3.4.1	Standards compliance . . . . .	9
3.4.2	Hardware limitations . . . . .	9
3.4.3	etc... . . . .	9
3.5	Software System Attributes . . . . .	9
3.5.1	Reliability . . . . .	9
3.5.2	Availability . . . . .	9
3.5.3	Security . . . . .	9
3.5.4	Maintainability . . . . .	9
3.5.5	Portability . . . . .	9
3.6	Other Requirements . . . . .	9



## List of Figures

# 1 Introduction

In this chapter there are the main purpose of this document, the project's scope and its actors and goals, some definitions who help to understand the content of the document and some reference documents that are used to make this one.

## 1.1 Purpose

This document is called *Requirements Analysis and Specifications Document*, also known as the acronym RASD. Its purpose is to communicate to customers what is understood about functional and not-functional requirements based, the limitations and obstacles for implementing this system, the constraints founded and for modeling the customer's need. This document is also addressed to developers and programmers who have to implement all the requirements then it must be more complete and correct than possible. It is a contract with customers therefore it must show use cases to allow everyone to understand what the system will do and in what domain it can be used. Project manager can use this document to make an evaluation of costs and size of the project.

## 1.2 Scope

The aim of the project called PowerEnJoy is to provide a car-sharing service that involves only electric cars. All people who want to share a car must be able to register at the system using credentials such as name, surname, e-mail, nickname and giving a valid information payment (number of credit card) that is needed to pay for the service. When the user receives the password to log in, he can find available cars in a specific location and, if he wants, he can reserve it. The system unlocks the car as soon as the user is nearby and keeps informed of the amount of the service with a screen on the car; when the car is in one of the safe areas, indicated in a list that can be consulted on-line, the system locks it after the person exits from it. The project has also the purpose to encourage people to leave at home their pollutant cars and take with other people the electric car: in fact if there are at least two passengers with the driver he has a 10 per cent discount on the ride. If the user leaves the car in the safe area with more than half of the battery he has a 20 per cent discount on the ride and if he recharged the car he will have 30 per cent discount. But if the user leaves the car far away from a safe area or with more than 80 per cent of empty battery, the system charges 30 per cent more on the ride.

### 1.3 Actors

Here there are a list of the actors who can operate with the system.

- > VISITOR: the person who visits the systems but that is not log-in in the site, he can only see the home page and the page with the form for the registration, where he must provide all the requested information. Moreover, he has the possibility to log-in with the password given by the system when the registration is successfully committed.
- > USER: the person who has successfully log-in. He can do all the operations provided by the system through the user interface such as reserve a car or consult the list of available cars or say to the system that he is nearby the reserved car.

### 1.4 Goals

In this section we want to explain exhaustively what are our system's goal.

- G1** The system must be able to allow visitor to register.
- G2** The system must be able to give a password to a visitor successfully registered.
- G3** The system must be able to allow visitor to log in.
- G4** The system must be able to provide to the user the list of available cars near his position or a specific location.
- G5** The system must be able to allow user to reserve a car up to one hour before it is picked up.
- G6** If the user is near his reserved car the system must unlock it.
- G7** The system must be able to charge the user.
- G8** The system must be able to keep informed the user about the charge through a display in the car.
- G9** If the user not pick up the car after one hour from the registration, the system must be able to delete the registration giving a fee of 1 euro to the user.
- G10** The system must allow user to delete a reservation before one hour is passed.
- G11** If a registration is deleted the system must be able to make the car available.
- G12** The system must lock a car when it is left in a safe area.

- G13** The system must be able to apply discount if is verified one case.
- G14** The system must be able to apply an increase to the amount of a ride if is verified a specific case.

## 1.5 Definitions, acronyms, abbreviations

### 1.5.1 Definitions

This section is necessary for avoid ambiguity or misunderstanding during the reading of this document.

- **VISITOR:** he is a person that is not register in the system, he can only see the homepage and go to the form for the registration.
- **USER:** he is a person that is registered in the system; he is identified by a name, surname, nickname, e-mail, password (given by the system at the end of the registration), telephone number, address and all the payment information such as number of credit card and card's deadline that must be verified by the system. He can do all the services that are provided by "*PowerEnJoy*".
- **CAR:** we intend an electric car that can be shared trough the system. It can be available or reserved and the system will lock and unlock it when is necessary. It is parked in a safe area.
- **RIDE:** with this term we intend all the route that a user accomplishes with the same car from the moment when he pick up the car to the moment when I left it in a safe area.
- **CHARGE:** we intend the debt that the user must pay at the end of a ride; it is calculate from a specific amount of money per minute which start from the begin of the ride to the end of it. That can be also modified by some discount, if is verified a particular situation, or a fee. It exists also a charge if the user reserve a car but he don't pick it up after one hour from the reservation.

### 1.5.2 Acronyms

Here there is the acronims list:

<b>RASD</b>	Requirements Analysis and Specifications Document
<b>UML</b>	Unified Modeling Language
<b>API</b>	Application Programming Interface
<b>DB</b>	DataBase

### 1.5.3 Abbreviations

- **Gn** : indicates the goal's number
- **Rn** : indicates the requirement's number for a specific goal

## 1.6 Overview

This document is structured in four parts:

- Part 1** In the first part there is an introduction with this document's purpose, the scope of this project and its aim, the actor that can use the system and in which way, some definitions to avoid misunderstanding during the reading of the document and, most important, the goal of our project described all in brief but comprehensive way.
- Part 2** In the second part there are more specifications about the requirements, the interfaces with external agents, constraints and assumptions.
- Part 3** The third part is very important because there are all the models for the requirement: they are modeled using UML diagrams such as *Use Case* and *Sequence Diagram* and using Alloy. They are very important to understand all the functionality of the system.
- Part 4** In this part there is the Appendix with other informations and the hour spent by all of us to make this document.

## 1.7 Reference Documents

- Specification document: Assignments AA 2016-2017.pdf
- Standard for RASD: IEEE Std 29148-2011
- API information: <https://developers.google.com/maps/documentation/geolocation/intro>



## 2 Overall Description

In this chapter we explain the product prospective and major functionality and the characteristic of the user that we image will use this application. We also write all the constraints we have found and all the assumptions that we have made because the requirements are not very clear in all their parts.

### 2.1 Product perspective

Our application is a self-contained product based on a client interface that helps the user to use this applications. It uses an API for the geolocation provided by *Google Maps* because we need to know the car and user's position. It is supported by common browser such as "*Chrome*". It is also closely linked with a DB that contains all user and car data and the list of safe areas. Because of our assumptions, the system must collaborate with the motorization's system to check user's driving license.

### 2.2 Product functions

### 2.3 User characteristics

The user we expect will use our application is a person that want to share a car every time near his position, in an efficient and rapid way; he is also a person who is interested in the problem of pollution because our system involves only electric car. Our application doesn't require a particular ability in using IT equipment: in fact, he must only be able to insert his credential for registration and log in and set the position from where he wants to share the car. Thanks to the position, the system will do all the operation to lock and unlock the car and charge the user.

### 2.4 Constraints

Our software must be used at the same time by a lot of user; it mustn't consume a lot of battery and memory. It is very important that is followed a policy of privacy: in fact, if the user doesn't sign and approve the warranty of privacy, the system can't be used. The privacy's policy is very important and is a constraint for the system develop because for the use of the application we must know the position of the user and that can't be done without the approval of the user. Another constraint is that, since we use a Google API, we must follow its regulation and developer guide.

## 2.5 Assumptions and Dependencies

1. There isn't a privileged user: all the users can do the same things and there isn't a limitation in system's use.
2. The visitors can only see the homepage, the registration's form and the log in page.
3. There isn't dependency from users.
4. A user can't reserve two cars at the same hour at the same day: he can reserve only one car each time.
5. There is a user's page with the payment history: so, the user can see all details for a specific payment because all the operation must be clear. This is a warranty of correctness between system's owner and users.
6. When a reservation is deleted because one hour is passed, the user is informed through a notification like this: *"The reservation for car X at the hour Y for the day Z is deleted because is passed one hour. You must pay a fee of 1 euro"*.
7. In the list of available cars is specified the car model with all the details so the user can select the best one for his needs. It is also indicated whether the vehicle is suitable for the transport of disabled persons.
8. During the reservation there is the possibility to set a remainder to avoid to forget the reservation.
9. In the site there is a list of possible discount and its details.
10. During the reservation the system provides the possibility to insert credential to make invoice for tax deduction or reimbursement of expenses.
11. During the registration we ask the user to insert his driving license's number: in this way the system can verify if it is valid or not.
12. The system provides the opportunity to receive through the e-mail the password, if the user doesn't remember it.

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interface

#### 3.1.2 Hardware Interfaces

#### 3.1.3 Software Interfaces

#### 3.1.4 Communication Interfaces

### 3.2 Functional Requirements

In this section we explain all the functional requirements necessary to reach a specific goal. For each goal that was mentioned in the first section here we write its requirements.

#### G1 :

- R1- During the registration visitors can't choose an username already in use.
- R2- During the registration visitors can't choose an e-mail already in use.
- R3- During the registration visitors must insert valid credit card's information.
- R4- During the registration visitors must insert a valid driving license.
- R5- During the registration visitors must agree with the privacy conditions to use the system.

#### G2 :

- R1- The registration must be successfully completed.

#### G3 :

- R1- The user must be registered at the system.
- R2- In the log in form the user must insert the correct username.
- R3- In the log in form the user must insert the correct password.
- R4- If the visitor insert the wrong data, the system shows an error and return to the log in page.
- R5- If the user doesn't remember the password he can receive it trough a special command.

**G4 :**

- R1- The user must insert a valid address or give to the system his position.
- R2- All cars position must be knowed.
- R3- All cars are set as available or not.
- R4- The list of nearest cars is made trough calculate the distance between car and user's position.

**G5 :**

- R1- The user must be correctly logged in.
- R2- The user must select which car he wants.
- R3- The user must select only date and times valid for the reservation.

**G6 :**

- R1- The position of the user and the car must be the same.
- R2- The reservation must not be expired.
- R3- The system must recognize the correct car reserved by a specific user to unlock the correct one.

**G7 :**

- R1- The car's engine must ignites.
- R2- An amount of money per minute is set.
- R3- The final charge is based on the duration of the car use.

**G8 :**

- R1- There must be a display on the car that communicates with the system.
- R2- The charge is update every minute.

**G9 :**

- R1- It must be passed one hour from the registration.
- R2- Is applied a fee of 1 euro to the user involved.
- R3- The reservation is deleted.

**G10 :**

- R1- The user must be correctly logged in.
- R2- The user must go to the reservation page.
- R3- Mustn't be passed one hour from the reservation.

**G11 :**

- R1- The reservation is correctly deleted by a user or it is expired.
- R2- The car is set available in the list.

**G12 :**

- R1- A list of safe areas must be available to the user.
- R2- The car must be left in a safe area.
- R3- The user and all any passengers must exit from the car.

**G13 :**

- R1- The car must be used by a user.
- R2- Discounts are in a list where is defined all details.
- R3- In the car there is a sensor which counts how many people there are in the car, if they are three or more a discount is applied.
- R4- In the car there is a sensor which controls how many battery is empty, if it is no more than 50 per cent the system applied a discount.
- R5- If the user takes care to link the car to recharge the car he will have a discount.

**G14 :**

- R1- The car must have been recently used.
- R2- Most of 80 per cent of the battery is empty.
- R3- The car is left far away from a safe area.

### 3.3 Performance Requirements

### 3.4 Design Constraints

#### 3.4.1 Standards compliance

#### 3.4.2 Hardware limitations

#### 3.4.3 etc...

### 3.5 Software System Attributes

#### 3.5.1 Reliability

#### 3.5.2 Availability

#### 3.5.3 Security

#### 3.5.4 Maintainability

#### 3.5.5 Portability

### 3.6 Other Requirements

## 4 Appendix