

Politecnico di Milano  
A.A. 2016–2017  
Software Engineering 2: “PowerEnJoy”  
*Requirements Analysis and Specification*  
*Document*

Pietro Ferretti, Nicole Gervasoni, Danilo Labanca

November 6, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Scope . . . . .	4
1.3	Definitions, Acronyms, Abbreviations . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Acronyms . . . . .	4
1.3.3	Abbreviations . . . . .	4
1.4	Reference Documents . . . . .	4
1.5	Document Overview . . . . .	4
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	Product Perspective . . . . .	5
2.2	Product Functions . . . . .	5
2.3	User Characteristics . . . . .	5
2.4	Constraints . . . . .	5
2.5	Assumptions and Dependencies . . . . .	5
<b>3</b>	<b>Specific Requirements</b>	<b>6</b>
3.1	External Interface Requirements . . . . .	6
3.1.1	User Interfaces . . . . .	6
3.1.2	Hardware Interfaces . . . . .	6
3.1.3	Software Interfaces (API) . . . . .	6
3.1.4	Communication Interfaces . . . . .	6
3.2	Functional Requirements . . . . .	7
3.2.1	Requisiti . . . . .	7
3.2.2	A guest registers to PowerEnJoy . . . . .	12
3.2.3	A user logs in the PowerEnjoy application . . . . .	13
3.2.4	A user searches an available car near his position . . . . .	13
3.2.5	A user searches an available car in a specific position . . . . .	14
3.2.6	A user reserves a car . . . . .	14
3.2.7	A user unlocks the car with the QR code printed on the car . . . . .	15
3.2.8	A user unlocks the car using his position . . . . .	15
3.2.9	A user parks the car without end the ride . . . . .	15
3.2.10	The system suggests to the user a PGS to park the car and save money . . . . .	16
3.3	Performance Requirements . . . . .	16
3.4	Design Constraints . . . . .	16
3.5	Software System Attributes . . . . .	16
3.5.1	Reliability . . . . .	16
3.5.2	Availability . . . . .	16
3.5.3	Security . . . . .	16
3.5.4	Maintainability . . . . .	16
3.5.5	Portability . . . . .	16
3.6	(Scenarios?) . . . . .	16

3.7 Alloy Model . . . . .	16
<b>4 Appendix</b>	<b>17</b>
4.1 Software and tools used . . . . .	17
4.2 Hours of work . . . . .	17
<b>5 Revisions</b>	<b>18</b>
5.1 Changelog . . . . .	18

# **1 Introduction**

## **1.1 Purpose**

This document is the Requirements Analysis and Specification Document (hereinafter abbreviated as RASD). The aim of the RASD is to give a complete and robust description of the requirements that our system has to fulfil adopting the SO/IEC/IEEE 29148 dated Dec 2011 standard. It also introduces the functional and non-functional requirements via UML diagrams. In the last part of this document it presents the formal model of the specification using Alloy analysis (language).

The information in this document are intended for the customer and the developers of the project. For the customer this document represents a description useful to understand the project development and is the equivalent of an agreement legally binding; meanwhile for the developers it's an useful way to coordinate the development and show the matching between the stakeholders' requests and the developed solution.

## **1.2 Scope**

## **1.3 Definitions, Acronyms, Abbreviations**

### **1.3.1 Definitions**

### **1.3.2 Acronyms**

### **1.3.3 Abbreviations**

## **1.4 Reference Documents**

## **1.5 Document Overview**

## **2 Overall Description**

### **2.1 Product Perspective**

### **2.2 Product Functions**

### **2.3 User Characteristics**

### **2.4 Constraints**

### **2.5 Assumptions and Dependencies**

## **3 Specific Requirements**

### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

#### **3.1.2 Hardware Interfaces**

#### **3.1.3 Software Interfaces (API)**

#### **3.1.4 Communication Interfaces**

## 3.2 Functional Requirements

### 3.2.1 Requisites

- [G1] Guests must be able to register as a user by choosing a username and providing their personal data, driving license and payment information. They will receive a password at the email address they specified.
  - [RE.1.1] The system must offer an interface (e.g. a form on a web page or on the mobile application) where the user can enter and submit the data needed for registration: personal data (first name, last name, date of birth [more?]), a driving license ID or a photo of one, and a payment method
  - [RE.1.2] The system must verify that the data provided by a user on registration is valid [...]
  - [RE.1.3] When a user registers the system must randomly generate a safe password and associate it to the user's account
  - [RE.1.4] After a password is generated when a user registers the system must send the password to the user via email at the email address the user provided
- [G2] Users must be able to log in with the username/email they submitted and the password received on registration.
  - [RE.2.1] The system must offer an interface (e.g. a form on a web page or on the mobile application) where the user can enter and submit his credentials (username/email and password) to log in.
  - [RE.2.2] The system must compare the credentials submitted by the user and compare them with the ones saved in the database
  - [RE.2.3] If the credentials submitted on login are valid, the system authenticates the user and allows them to use the available functionalities
  - [RE.2.4] If the credentials submitted on login are not valid, the system denies the user access and notifies the user of the error [+ offre di riprovare?]
- [G3a] Users must be able to find the locations of available cars within a certain distance from their current location.
  - [RE.3a.1] The system must offer an interface where the user can (activate the option) to look for nearby available cars if the user's location is available (ex. gps)
  - [RE.3a.2] When the user activates the option to find nearby available cars, the system must look in the database for every available car near the the user's location

- [RE.3a.3] The system must show the results of the search for nearby available cars on a suitable \*interface\* (ex. a map)
- [G3b] Users must be able to find the locations of available cars in a specified location.
  - [RE.3b.1] The system must offer an interface where the user can submit an address and search for available cars near that location
  - [RE.3b.2] When the user starts a search for available cars near a certain address, the system must look in the database for every available car close to the address provided by the user
  - [RE.3b.3] The system must show the results of the search for available cars on a suitable \*interface\* (e.g. a map)
- [G4] Users must be able to reserve a car for up to one hour before they pick it up.
  - [RE.4.1] The system must offer an interface that shows to the user the cars available for reservation (e.g. the search results specified in RE.3a.3) and allow the user to select a car
  - [RE.4.2] When a user selects a car for reservation the system must mark the car as reserved by that user
  - [RE.4.3] If the car has not been unlocked for an hour after reservation, the system must mark the car as available and charge the user a fixed amount
- [G5] A user that reaches a reserved car must have a way to tell the system they're nearby, so that the system unlocks the car and the user may enter.
  - [RE.5.1] The system must offer an interface to unlock the car, with their position or \*car id\*/qr code
  - [RE.5.2] If the user is within a certain distance from the car, their device location is available and they choose from the interface to unlock the car with the position \*option\*, the system must unlock the car
  - [RE.5.3] If the user chooses from the interface to unlock the car with its \*id\* and the user \*submits the right id\* corresponding to the car, the system must unlock the car -> sbloccare con la posizione disponibile solo quando è vicino?
- [G6] The system must charge the user who reserved the car from the moment the engine is ignited after unlocking it. The amount charged depends on the distance traveled and time elapsed.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.



- [RE.6.2] The system starts to **\*\*charge the user\*\*** -> ambiguo, non paga subito the moment the reserved car's engine is ignited after unlocking the car.
- [RE.6.3] The amount charged by the system depends on the time elapsed and the distance traveled by the user, following specific rules.
- [G7] The system must allow the user to see the amount they're being charged through a screen on the car.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.
  - [RE.7.1] After the system has started keeping track of the amount that the user has to pay, the system must show it in a clear way on the car screen while the car is unlocked and the reservation hasn't ended.
- [G8] The system must stop charging the user as soon as the car is parked in a safe area and the user exits the car without notifying the system they are temporarily parking.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.
  - [RE.8.1] The moment a car is parked in a safe area, the system must ask through **\*an interface\*** on the car screen if the user wants to end the reservation or just park temporarily.
  - [RE.8.2] If a reserved car has been parked in a safe area and the user has chosen to end the reservation, then the moment the last door has been closed the system must carry out the transaction for the amount that the system calculated, with the method of payment specified by the user on registration. -> si ricorda l'assunzione che quando tutte le porte sono chiuse dopo aver terminato, allora i passeggeri sono tutti usciti
- [G9] Users must be able to leave a car in a safe area without losing the reservation by informing the system that they're only temporarily parking; in this case the system continues on charging the user.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.
  - [RE.8.1] The moment a car is parked in a safe area, the system must ask through **\*an interface\*** on the car screen if the user wants to end the reservation or just park temporarily.
  - [RE.9.1] When a car is temporarily parked and locked, the system keeps on **\*charging\*** the user but with different rules than those used in RE.6.3, this time depending only on time elapsed while parked.

- [RE.9.2] After a temporarily parked car is unlocked by the user and the engine is turned on, the system starts charging the user with the rules used in RE.6.3 again.
- [G10] The system must lock the car automatically after the user exits the car.
  - [RE.8.1] The moment a car is parked in a safe area, the system must ask through \*an interface\* on the car screen if the user wants to end the reservation or just park temporarily.
  - [RE.10.1] The system must lock the car when the last door has been closed after the user chose to end their reservation or park temporarily.
- [G11] The system must provide a money saving option to the user, offering a suitable power grid station near the user's final destination such that the user will get a discount if they park the car there and plug it in the power grid.
  - [RE.11.1] The system must offer an interface where the user can select the money saving option, (submitting their final destination)
  - [RE.11.2] When the user submits their final destination for the money saving option, the system must find a suitable power grid station near the ones close to the destination that have at least one free spot
  - [RE.11.3] The system must show on the car screen (e.g. on a map) the location of the power grid station chosen for the money saving option
  - [RE.11.4] The system must apply a discount on the amount the user has to pay if they selected the power saving option and they parked in the station the system chose for them. -> c'è un problemino se due utenti si dirigono entrambi a una stazione con 1 posto libero, risolvibile "prenotando" i posti
- [G12] The stations proposed by the system with the money saving option must be chosen in a way that ensures a uniform distribution of cars in the city.
  - [RE.12.1] To choose a station in the process specified in RE.11.2, the system selects among all the stations with free spots within a certain distance of the user's final destination (e.g. 10 minutes by foot) the one with the fewest available cars nearby.
- [G13] The system must apply a discount of 10
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.

- [RE.13.1] If the weight sensors in the car seats detect at least two passengers (excluding the driver) for at least half of the time the car has been driving, then the system must apply a discount of 10% at the end of the reservation on the amount the user has to pay.
- [G14] The system must apply a discount of 20% on the last ride if the car is left with more than 50% of remaining battery charge.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay.
  - [RE.14.1] When the car has been parked in a safe area and the user has chosen to end the reservation, the system must apply a discount of 20% on the amount the user has to pay if the car is left with more than 50% remaining battery charge.
- [G15] The system must apply a discount of 30% on the last ride if the car is left at special parking areas where they can be recharged, and the user takes care of plugging the car into the power grid.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay. se la macchina ha terminato la corsa (utente ha indicato di aver finito ed è uscito dalla macchina) ed è stata parcheggiata a una stazione con ricarica, ed è attaccata alla presa, il sistema applica il 30% di sconto sul totale da pagare alla fine del viaggio
  - [RE.15.1] When the car has been parked in a power grid station, has been plugged in the power grid and the user has chosen to end the reservation, the system must apply a discount of 30% on the amount the user has to pay.
- [G16] The system must charge 30% more if the car is left at more than 3 km from the nearest power grid station or with less than 20% remaining battery charge.
  - [RE.6.1] The system must keep track until the end of the reservation of the amount the user has to pay. se la macchina ha terminato la corsa (utente ha indicato di aver finito ed è uscito dalla macchina) e il sistema rileva che è stata parcheggiata a più di 3km di distanza da una qualsiasi stazione, oppure è rimasta meno del 20% di carica, il sistema carica 30% di più sul totale da pagare alla fine del viaggio
  - [RE.16.1]

(use cases, con sequence/activity diagrams)

### 3.2.2 A guest registers to PowerEnjoy

Actor	Guest
Goal	[G1]
PreConditions	The guest had never been registered before
Execution Flow	<ol style="list-style-type: none"><li>1. The guest on the home page clicks on “register” button to start the registration process.</li><li>2. The guest fills in at least all mandatory fields with the required informations(name, surname, username, email address, DOB).</li><li>3. The guest uploads a photo of the driving license or inserts manually the informations.</li><li>4. The guest inserts the number of the credit card and the relative CVV.</li><li>5. The system verifies the correctness of the inserted data</li><li>6. The guest clicks on “confirm” button.</li><li>7. The system generates a password and provides it to the user.</li><li>8. The system will save the data in the DB.</li><li>9. The system notifies the registration and sends the user to the profile management page.</li></ol>
Postconditions	The guest successfully ends registration process and become a user. From now on he can log in to the application using his credential and start using PowerEnjoy.
Exceptions	<ol style="list-style-type: none"><li>1. The guest is already registered.</li><li>2. The guest inserts invalid information.</li><li>3. The guest inserts a username used by another user.</li><li>4. The guest inserts an email used by another user.</li><li>5. The guest doesn’t confirm the registration.</li></ol> <p>Each exception is handled warning the guest of the problem and the Execution Flow comes back to the point 2.</p>

### 3.2.3 A user logs in the PowerEnjoy application

Actor	Guest
Goal	G[2]
PreConditions	The user must be registered in the system.
Execution Flow	<ol style="list-style-type: none"><li>1. The guest opens the PowerEnjoy application and presses on the login button.</li><li>2. The guest inserts the username or email and password received during registration.</li><li>3. The system checks the couple inserted by the user.</li><li>4. The guest, from now user, is redirected to the page where he can search a car.</li></ol>
Postconditions	The guest is now a user, he is logged in and can use all the functionality of the system.
Exceptions	<ol style="list-style-type: none"><li>1. The guest inserts invalid credentials.</li></ol>

### 3.2.4 A user searches an available car near his position

Actor	User
Goal	[G3a]
PreConditions	The user is logged in to the system and he has activated the GPS.
Execution Flow	<ol style="list-style-type: none"><li>1. The user presses the button to be localized on the map.</li><li>2. The system receives the user's position and checks in the DB all the available cars nearby the user.</li><li>3. The system shows on the application all the available cars.</li><li>4. The user navigates on the map to search a car.</li></ol>
Postconditions	The user finds a car most suitable for him.
Exceptions	<ol style="list-style-type: none"><li>1. There aren't any available cars and the system suggests to the user to search in another location.</li></ol>

### 3.2.5 A user searches an available car in a specific position

Actor	User
Goal	[G3b]
PreConditions	The user is logged in to the system
Execution Flow	<ol style="list-style-type: none"><li>1. The user presses the search bar to insert a location.</li><li>2. The user inserts an address (street, building, place (vorrei intendere pub, bar, discoteche))</li><li>3. The system receives the address inserted by the user and checks in the DB all the available cars nearby the location.</li><li>4. The system shows on the application all the available cars.</li><li>5. The user navigates on the map to search a car.</li></ol>
Postconditions	The user finds a car most suitable for him.
Exceptions	<ol style="list-style-type: none"><li>1. The address inserted by the user doesn't exist.</li><li>2. There aren't any available cars and the system suggests to the user to search in another location.</li></ol>

### 3.2.6 A user reserves a car

Actor	User
Goal	[G4]
PreConditions	The user is logged and there is at least an available car.
Execution Flow	<ol style="list-style-type: none"><li>1. The user selects a car in the map.</li><li>2. The system shows to the user the battery remaining charge.</li><li>3. The user confirms to reserve the car.</li></ol>
Postconditions	The car is reserved for the user for an hour.
Exceptions	<ol style="list-style-type: none"><li>1. The car is reserved by an another user before the user confirm the reservation.</li><li>2. The user doesn't confirm the reservation (non so se è un'eccezione)</li></ol>

### 3.2.7 A user unlocks the car with the QR code printed on the car

Actor	User
Goal	[G5]
PreConditions	The user is nearby the car he reserved.
Execution Flow	<ol style="list-style-type: none"><li>1. The user presses on the camera button and sends the QR code to the system.</li><li>2. The system identifies the car with the QR code and checks the reservation.</li><li>3. The system enables the button to unlock the car on the application.</li><li>4. The user presses the button.</li></ol>
Postconditions	The car is ready to be ignite.
Exceptions	<ol style="list-style-type: none"><li>1. The user sent a QR code of a car he didn't reserve.</li></ol>

### 3.2.8 A user unlocks the car using his position

Actor	User
Goal	[G5]
PreConditions	The user is nearby the car he reserved and has the localization activated.
Execution Flow	<ol style="list-style-type: none"><li>1. The user presses on the localization button and sends to the system his position.</li><li>2. The system checks the user's position and the reservation.</li><li>3. The system enables the button to unlock the car on the application.</li><li>4. The user presses the button.</li></ol>
Postconditions	The car is ready to be ignite.
Exceptions	<ol style="list-style-type: none"><li>1. The user is nearby a car he didn't reserve.</li><li>2. The user is far from the car he reserved</li></ol>

### 3.2.9 A user parks the car without end the ride

Actor	User
Goal	[G9]
PreConditions	The user is employing the car.
Execution Flow	<ol style="list-style-type: none"><li>1. The user stops the car and turns it off.</li><li>2. The display shows to the user the option:<ol style="list-style-type: none"><li>a) to end the ride;</li><li>b) to stand without losing the control of the car.</li></ol></li><li>3. The user presses the button b).</li><li>4. The user closed the car, keeping the key.</li></ol>
Postconditions	The car is stopped in a parking lot, ready to be used again.
Exceptions	non me ne vengono

### 3.2.10 The system suggests to the user a PGS to park the car and save money

Actor	System, user
Goal	[G11]
PreConditions	The user is employing the car.
Execution Flow	<ol style="list-style-type: none"><li>1. The user stops the car and turns it off.</li><li>2. The display shows to the user the option:<ol style="list-style-type: none"><li>a) to end the ride;</li><li>b) to stand without losing the control of the car.</li></ol></li><li>3. The user presses the button a).</li><li>4. The system obtains the position of the car.</li><li>5. The system search the nearest PGS and suggests it to the user through the display.</li><li>6. The user drives to the PGS, parks the car and ends the ride.</li><li>7. The user plugs in the car in the power grid.</li><li>8. The system detects that the car is charging.</li><li>9. The system applies a discount on the amount the user must pay.</li></ol>
Postconditions	The car is parked, ready to be used again and the user has a discount.
Exceptions	<ol style="list-style-type: none"><li>1. There isn't an available PGS.</li><li>2. The user decides to park the car where he is.</li></ol>

## 3.3 Performance Requirements

## 3.4 Design Constraints

## 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



## 4 Appendix

### 4.1 Alloy Model

### 4.2 Software and tools used

### 4.3 Hours of work

## 5 Revisions

### 5.1 Changelog