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

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

(use cases, con sequence/activity diagrams)

3.2.1 A guest registers to PowerEnJoy

Actor	Guest
Goal	[G1]
PreConditions	The guest had never been registered before
Execution Flow	1. The guest on the home page clicks on “register” button to start the registration process. 2. The guest fills in at least all mandatory fields with the required informations(name, surname, username, email address, DOB). 3. The guest uploads a photo of the driving license or inserts manually the informations. 4. The guest inserts the number of the credit card and the relative CVV. 5. The system verifies the correctness of the inserted data 6. The guest clicks on “confirm” button. 7. The system generates a password and provides it to the user. 8. The system will save the data in the DB. 9. The system notifies the registration and sends the user to the profile management page.
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	1. The guest is already registered. 2. The guest inserts invalid information. 3. The guest inserts a username used by another user. 4. The guest inserts an email used by another user. 5. The guest doesn’t confirm the registration. Each exception is handled warning the guest of the problem and the Execution Flow comes back to the point 2.

3.2.2 A user logs in the PowerEnjoy application

Actor	Guest
Goal	G[2]
PreConditions	The user must be registered in the system.
Execution Flow	1. The guest opens the PowerEnjoy application and presses on the login button. 2. The guest inserts the username or email and password received during registration. 3. The system checks the couple inserted by the user. 4. The guest, from now user, is redirected to the page where he can search a car.
Postconditions	The guest is now a user, he is logged in and can use all the functionality of the system.
Exceptions	1. The guest inserts invalid credentials.

3.2.3 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	1. The user presses the button to be localized on the map. 2. The system receives the user's position and checks in the DB all the available cars nearby the user. 3. The system shows on the application all the available cars. 4. The user navigates on the map to search a car.
Postconditions	The user finds a car most suitable for him.
Exceptions	1. There aren't any available cars and the system suggests to the user to search in another location.

3.2.4 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	1. The user presses the search bar to insert a location. 2. The user inserts an address (street, building, place (vorrei intendere pub, bar, discoteche)) 3. The system receives the address inserted by the user and checks in the DB all the available cars nearby the location. 4. The system shows on the application all the available cars. 5. The user navigates on the map to search a car.
Postconditions	The user finds a car most suitable for him.
Exceptions	1. The address inserted by the user doesn't exist. 2. There aren't any available cars and the system suggests to the user to search in another location.

3.2.5 A user reserves a car

Actor	User
Goal	[G4]
PreConditions	The user is logged and there is at least an available car.
Execution Flow	1. The user selects a car in the map. 2. The system shows to the user the battery remaining charge. 3. The user confirms to reserve the car.
Postconditions	The car is reserved for the user for an hour.
Exceptions	1. The car is reserved by an another user before the user confirm the reservation. 2. The user doesn't confirm the reservation (non so se è un'eccezione)

3.2.6 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	1. The user presses on the camera button and sends the QR code to the system. 2. The system identifies the car with the QR code and checks the reservation. 3. The system enables the button to unlock the car on the application. 4. The user presses the button.
Postconditions	The car is ready to be ignite.
Exceptions	1. The user sent a QR code of a car he didn't reserve.

3.2.7 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	1. The user presses on the localization button and sends to the system his position. 2. The system checks the user's position and the reservation. 3. The system enables the button to unlock the car on the application. 4. The user presses the button.
Postconditions	The car is ready to be ignite.
Exceptions	1. The user is nearby a car he didn't reserve. 2. The user is far from the car he reserved

3.2.8 A user parks the car without end the ride

Actor	User
Goal	[G9]
PreConditions	The user is employing the car.
Execution Flow	1. The user stops the car and turns it off. 2. The display shows to the user the option: a) to end the ride; b) to stand without losing the control of the car. 3. The user presses the button b). 4. The user closed the car, keeping the key.
Postconditions	The car is stopped in a parking lot, ready to be used again.
Exceptions	non me ne vengono

3.2.9 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	1. The user stops the car and turns it off. 2. The display shows to the user the option: a) to end the ride; b) to stand without losing the control of the car. 3. The user presses the button a). 4. The system obtains the position of the car. 5. The system search the nearest PGS and suggests it to the user through the display. 6. The user drives to the PGS, parks the car and ends the ride. 7. The user plugs in the car in the power grid. 8. The system detects that the car is charging. 9. The system applies a discount on the amount the user must pay.
Postconditions	The car is parked, ready to be used again and the user has a discount.
Exceptions	1. There isn't an available PGS. 2. The user decides to park the car where he is.

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

3.6 (Scenarios?)

3.7 Alloy Model

4 Appendix

4.1 Software and tools used

4.2 Hours of work

5 Revisions

5.1 Changelog