
PowerEnJoy
Software Engineering II

# Requirements Analysis and Specification Document

*Giovanni Scotti, Marco Trabucchi*

Document version: 1
October 29, 2016

# Contents

# Section 1

# Introduction

## 1.1 Purpose

The Requirement Analysis and Specification Document for the *PowerEnJoy* digital management system is intended to describe the system itself, the functional and non-functional requirements, its components as well as its constraints and the relationship with the real world and the users by providing several use cases and scenarios. Furthermore part of the documentation makes use of Alloy, a language for describing structures and a tool for exploring them, and gives a formal specification of some features of the system to be.

This document establishes some baselines for the project planning, its estimation and evaluation and it may be legally binding; it is mainly addressed to developers and programmers, who have to implement the requirements, testers, who have to determine whether the requirements have been met, project managers, who control the development process and - last but not least - users, who validate the system goals.

## 1.2 Scope

The product is a digital management system to support a car-sharing service that exclusively employs electric cars.

The system consists of a back-end server application that manages rental requests remotely and three front-end applications:

- A web-based application to provide the final user with a friendly interface to take advantage of the services of *PowerEnJoy*;

- An application that runs on the existing on-board computers provided on each vehicle, used to interact with the car itself, unlock it and access the GPS/sat-nav service;
- A mobile application that allows the user to easily access the service anywhere he/she needs to.

The system is intended for only one type of user: drivers, who should be allowed to register and access the system via username and password, in order to make the renting and payment processes easier and quicker to carry out. Moreover, the system aids the users by locating nearby available vehicles and keeps track of the distance driven, all while notifying them about the amount of money they are being charged. A defined Safe Area stands as a limit for the service; properly equipped charging spots are signaled by an on-board computer.

Lastly, the system aims to motivate drivers to maintain a virtuous behavior providing discounts when it detects signs of responsible and ecologic actions.

## 1.3 Goals

The goals of *PowerEnJoy* are the following:

1. Let the user register to the service and login via the provided credentials;
2. Let the driver find the location of nearby available cars;
3. Let him/her reserve a chosen car up to an hour before picking it up;
4. Improve the efficiency of the service by assuring that no car stays reserved if not actually in use;
5. Allow the user to easily access the cars by unlocking them once the driver is in proximity;
6. Actively keep track of the driver's current charges, continuously notifying it;
7. Let the user know in detail where the Safe Area begins and ends;
8. Let the user know which are the nearest free charging spots;
9. Incentivize responsible behaviors, providing discounts for the worthiest users.

## 1.4 Definitions, Acronyms and Abbreviations

**API:** Application Programming Interface. A set of tools, protocols and libraries for building software and applications.

**Back-end application:** any computer program that remains in the background and offers application logic and communication interfaces to work with the front-end counterpart. It does not involve any graphical user interface, but it can provide a data access layer.

**Charging spot:** a particular parking spot equipped with a power grid and plugs to recharge the cars.

**CAN bus:** the Controller Area Network is a vehicle bus standard designed to allow microcontrollers and other devices to communicate with each other without a host computer.

**DBMS:** Database Management System.

**Driver:** See **User**.

**Front-end application:** any application the users interact with directly. It provides the so called presentation layer.

**GPS:** Global Positioning System.

**m:** meter, *Interational System of Units (SI)* unit of measure.

**Mobile broadband:** it is the marketing term for wireless internet access delivered through mobile phone towers to any digital device using a portable modem.

**OS:** Operating System.

**Parking spot:** a generic place whitin the Safe Area where the driver can park the vehicles.

**PC:** Personal Computer.

**PIN:** Personal Identification Number.

**RASD:** Requirements Analysis and Specification Document.

**Ride:** the trip that involves the use of a *PowerEnJoy* electric car.

**Safe Area:** the predefined area where it is possible to start and end a ride; it marks the boundaries within which the user can park a vehicle and conclude the ride so that the system stops charging him/her.

**System:** The software system-to-be, in all of its entirety.

**UMTS:** Universal Mobile Telecommunications System.

**User:** Any person subscribed to the service who rents a car using *PowerEnJoy*.

**Vehicle:** Any of the electric cars provided by *PowerEnJoy*.

## 1.5 References

This document follows the guidelines provided by ISO/IEC/IEEE 29148:2011 [1] and IEEE 830-1998 [2] respectively related to the requirements engineering for systems and software products and the recommended practice for software requirements specifications.

Moreover it is strictly based on the specifications concerning the RASD assignment [3] for the Software Engineering II project, part of the course held by professors Luca Mottola and Elisabetta Di Nitto at the Politecnico di Milano, A.Y. 2016/17.

## 1.6 Overview

This document consists of three sections:

**Section 1: Introduction.** A general introduction and overview of the system-to-be purpose, scope and goals, along with some important information about this document.

**Section 2: Overall description.** It describes the general factors that affects the product and its requirements. The section provides a background for those requirements which are defined in detail in Section 3 and makes them easier to figure out.

**Section 3: Specific Requirements.** All the software requirements are specified to a level of detail which is sufficient to let the designers satisfy them. Both functional and non-functional requirements are mentioned.

At the end of the document are an **Appendix** and a **Bibliography**, providing additional information about the sections listed above.

# Section 2

# Overall Description

## 2.1 Product Perspective

### 2.1.1 User interfaces

The users have several ways to access the system: a web application can be executed on any personal computer while a mobile application provides flexibility, portability and can be used literally everywhere. Despite the fact that the hardware interfaces running the application are rather different, a unified and common user interface is provided. It should be user friendly and very intuitive to allow everyone to easily use it without any specific knowledge.

Moreover the users have to interact with the on-board computer installed on each electric vehicle, therefore it should offer an interface as straightforward as the one implemented by the web and mobile applications.

### 2.1.2 Hardware interfaces

The web application can be executed on any general purpose computer that complies with the minimum system requirements specified in subsection 2.4.2.

The mobile application has to exchange data with the GPS module located on any recent smart-phone. Moreover it has to access the mobile broadband in order to communicate with the main system server.

An on-board computer is already set up in each electric car and it talks to the vehicle control unit through the CAN bus and to the system server via the mobile broadband. All sensors and hardware components needed to support the functionalities of the system are already installed on the vehicles.

### 2.1.3 Software interfaces

The web based application must support the main browsers such as IE, Google Chrome and Mozilla Firefox.

The mobile application has to be compatible with iOS and Android. The server side of the application, that is the system back-end, stores data in a relational DBMS and runs on any web server supporting Java.

The on-board computers run a customized Linux-based kernel and provide adequate APIs for the developers to communicate easily with the existing vehicle systems. This is to make communication between the system-to-be and the existing software easier.

The back-end has mainly to deal with the rental service and data management, essentially aimed towards keeping track of transactions and customers information.

## 2.2 Product Functions

The system allows users to reserve available cars, to manage their requests and charge them for the rental.

The users can:

- Register to the service;
- Login with their personal account;
- Edit and manage personal information;
- Delete their existing account;
- Locate nearby available-to-rent vehicles;
- Reserve one of said cars for a fixed amount of time;
- Unlock the reserved vehicle based on their mobile device GPS position;
- Alternatively, unlock the reserved car by inserting a vehicle-specific code (found on the windshield) into the application;
- Start the engine by inserting their PIN;
- Perform payments for the used service via credit card or $PayPal^{TM}$.

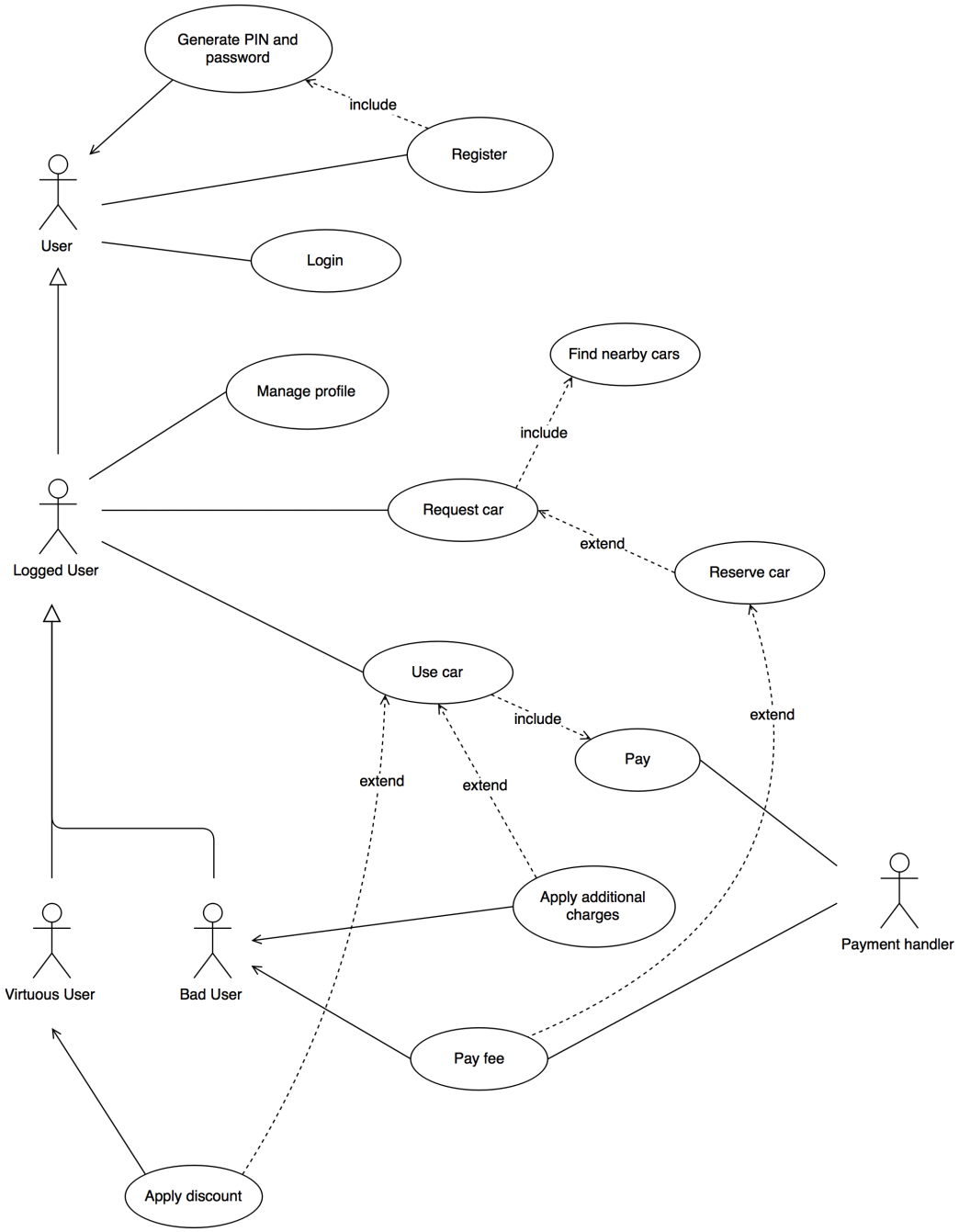


**Figure 1:** The comprehensive use-case diagram of all the functionalities provided by the system

## 2.3 User Characteristics

Target of the service are *drivers* that wish to rent a car.

The base assumption is that every registered user is in possess of a driving license which is valid in the country where he/she makes use of the service.

Moreover, the user must have the means of accessing the service from anywhere: this includes having the possibility to use a stable internet connection both from a mobile device and a stationary one such as a laptop or desktop PC.

Lastly, the user should provide the system a valid payment method upon the act of registration to the service.

## 2.4 Constraints

### 2.4.1 Regulatory policies

The system must be allowed by the user to collect, process and store personal data. Furthermore, the system is capable to delete all personal data upon user request and to keep track of each payment.

The user has the responsibility to use the system properly in order to comply with the local laws and policies.

### 2.4.2 Hardware limitations

The system should comply to these following minimum hardware requirements:

- Mobile application:
  - 3G UMTS connection at its maximum speed of 2 Mb/s
  - 50 MB of available space
  - 1 GB of RAM
  - GPS module
- Web application:
  - Internet connection at 7 Mb/s
  - 800x600 screen resolution

### 2.4.3 Parallel operation

The system must support parallel operations from different users and the DBMS relies heavily on concurrent transactions.

### 2.4.4 Reliability requirements

The system reliability, that is the probability to operate without a failure for a specific period of time, must be 99%.

### 2.4.5 Criticality of the application

Life-critical applications do not concern the system to be developed.

### 2.4.6 Safety and security consideration

The user must have a valid driving license in order to take advantage of the car sharing service. The license number is asked and stored by the system for security reason. The locations and the travels performed by the users must be tracked but kept private.

## 2.5 Assumptions and Dependencies

For the rest of the RASD, the following assumptions are given for granted:

- The GPS location of users and cars is always functioning and accurate, with an uncertainty of $\pm 1$ m;
- All users can access a reliable and stable internet connection;
- The users' mobile devices feature a working GPS;
- All users are always charged the correct amount after the ride;
- The on-board computers always notify the correct charge to the driver during the ride;
- All cars unlock properly upon insertion of the code by the user who reserved them or in case he/she is detected to be in proximity;
- The equipment of the cars always gives a correct reading of the number of passengers, driver included, for the current ride;

- The number of passengers never changes for the duration of the ride;
- Whenever a car is left with more than 80% of the battery empty, it is refueled before becoming available again;
- If a user reserves and uses a car he/she is the one who drives it and is responsible for the associated trip;

## 2.6 Future Extensions

The system implementation will open the possibility of including new features. For example:

- The possibility of including more categories of vehicles, such as motorcycles and bycicles;
- The possibility of receiving active feedback from the users, mainly concerning car conditions;
- The possibility for everyone, registered and unregistered people, of notifying *PowerEnJoy* of incorrect or unsafe situations involving their vehicles.

# Section 3

# Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User interface

The user interfaces must be strongly user-friendly, in order to provide an easy and intuitive way to access the system. The following constraints have to be satisfied by the web and mobile applications:

- The first page must always ask the user to login or register to the service;
- The main page must show a map with a marker for all the locations of all nearby available cars inside the Safe Area;
- The interface must offer the possibility to choose between a set of different languages;
- A toolbar must be visible in every screen except the first page;
- The toolbar must offer a quick and simple overview of the main system functionalities (reserve a car, reservations history, modify personal information...);
- The user interface must dynamically adapt to the screen size;
- In order to simplify the user experience, the web application and the mobile one must use the same graphical elements and layouts.

With respect to the on-board computer application, the restrictions are the following:

- After the car has been unlocked, the display of the on-board computer must turn on and show a screen thats allows the user to input his/her PIN in order to start the ride;
- Once the ride starts, the screen must clearly show the current charges;
- Similarly, during the ride, the screen must clearly show the nearby charging spots and visually signal the Safe Area boundaries;
- When the car is available, the screen must always be off.

Some mock-ups that give an overall idea of the interface structure of the applications are provided in the next section (RIFERIMENTO a FUNCTIONAL REQUIREMENTS). In addition to that several application-dependent constraints are provided:

- Web application

  All the web pages must abide by the W3C standards to ensure a long-term growth of the application and take the full advantage of HTML5 markup language and CSS.

- Mobile application

  Both the Android and iOS versions must follow the design guidelines provided by the respective platform manufacturers.

- Server application

  The server application must be easily configurable via a XML file and use the APIs provided by the payment handler to interface with its information system.

- On-board computers application

  The application mounted on on-board computers must use the APIs provided by the vehicle manufacturer to interface with the cars.

### 3.1.2 Hardware interface

The most relevant aspect to highlight, in addition to section (INSERIRE RIFERIMENTO ad hardware interface, section 2), is the ability of the

system-to-be to communicate with the car systems via the embedded computer located in the car itself that takes the advantage of the on-board computer application to be developed. The on-board computer can communicate with the mobile broadband thanks to a SIM and provides a GPS module exploitable by the car-side of the application.

### 3.1.3 Software interface

The server side of the application, that represents the back-end of the system-to-be, requires the following software products:

- Java SE 8 - `http://www.oracle.com/technetwork/java/javase/overview/index.html`
- MySQL 5.7 - `http://dev.mysql.com/`

The embedded on-board computer located in each car provides a Linux OS that is capable to run C/C++ programs. Moreover the car-side of the application can use specific software libraries provided by the manufacturer to access the car systems the computer is connected with.

### 3.1.4 Communication interface

The clients communicate with the server via HTTPS protocol by using TCP and default port 443. Moreover a safe and stable connection is also established between the server and the payment handler whenever the user performs a payment.

## 3.2 Functional Requirements

### 3.2.1 Login

#### Purpose

The main purpose of the login feature is to grant the access of the *PowerEnJoy* service to any registered user. The system requires a valid e-mail address and password to login without generating any error.

In addition to that, the login screen offers the chance to recover a forgotten password. The user clicks on "forgot password?", a new one is sent to his e-mail address immediately and he/she can carry out the login procedure with the new password provided by the system.

### Scenario 1

Mike would like to rent a car for having a ride in a beautiful sunny day. In order to do that he needs to access the system by means of his credentials. He opens the *PowerEnJoy* home page and enters his e-mail address and his password. Then Mike clicks on "login" and, due to the fact everything is correct, he obtains the access as a logged user.

### Scenario 2

Bill wants to take the advantage of the *PowerEnJoy* service. He opens the home page and he is asked for his e-mail address and password. He is aware of his e-mail address, but he does not remember the password. After some time spent trying to recall his personal password, Bill inputs his e-mail address and decides to click on "forgot password?". Right away the system sends a new password to the specified e-mail address. Bill can now enter the system using the new password.

### Use-case

### Statechart diagram

### Functional requirements

1. The user must be already registered to the system in order to perform a successful login;
2. The user must be aware of his e-mail address and password to successfully obtain the system access;
3. The password provided by the user must correspond to the specified e-mail address;
4. The system sends a new password to the specified e-mail address if and only if the specified e-mail address is valid, registered to the system and the user clicks on "forgot password?";
5. After requesting a new password, the system must allow the user login with the new provided password;
6. After three times the entered password is wrong, the system allows a new attempt 30 seconds later.

| | |
|---|---|
| Actor | User |
| Goal | Goal 1 |
| Input Condition | The user is already registered to the system and wants to login. |
| Event Flow | 1. The user opens the *PowerEnJoy* home page or the mobile application and the system shows the login page;<br>2. The user enters his/her e-mail address and password. |
| Output Condition | The system lets the user to log in if the provided credentials are valid and loads his/her personal home page. |
| Exception | If the e-mail address provided by the user has never been registered to the system or if the password does not correspond to the entered e-mail, the system notifies the user with an error message. If the user enters wrong credentials three times in a row, the system will prevent any attempt for the following 30 seconds. |

**Table 1:** Login use-case

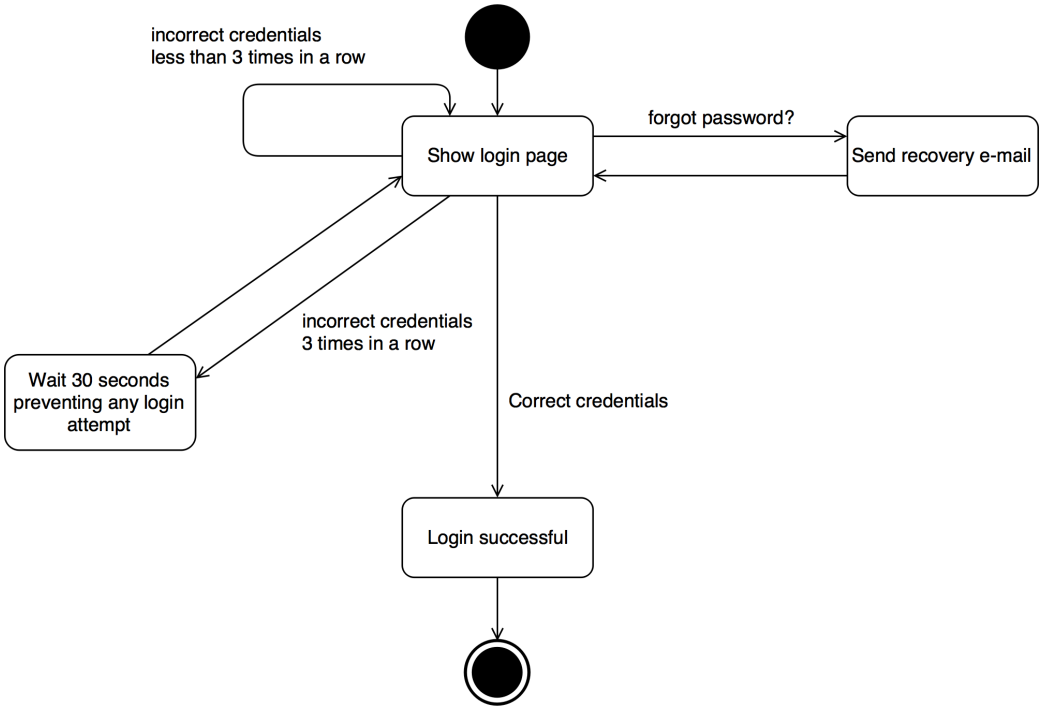


**Figure 2:** Statechart of the login process

# Appendix A

# Appendix

## A.1 Software and tools used

- LaTeX, used as typesetting system to build this document.
- draw.io - `https://www.draw.io` - used to draw diagrams and mock-ups.
- GitHub - `https://github.com` - used to manage the different versions of the document and to make the distributed work much easier.
- GitHub Desktop, the GitHub official application that offers a seamless way to contribute to projects.

## A.2 Hours of work

# Bibliography

[1] ISO/IEC/IEEE 29148:2011 *Systems and software engineering - Life cycle processes - Requirements engineering*

[2] IEEE 830:1998 *Recommended Practice for Software Requirements Specifications*

[3] AA 2016/2017 Software Engineering 2 - *Project goal, schedule and rules*