

# Power EnJoy

Emanuele Ghelfi  
Emiliano Gagliardi

October 29, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Description of the given problem . . . . .	3
1.2	Glossary and abbreviations . . . . .	3
1.3	Actors . . . . .	3
1.4	Assumptions . . . . .	3
1.5	Domain assumptions . . . . .	4
1.6	Goals . . . . .	4
1.7	Document overview . . . . .	5
<b>2</b>	<b>Overall description</b>	<b>6</b>
2.1	Product perspective . . . . .	6
2.1.1	Payment methods . . . . .	6
2.1.2	Car technology . . . . .	6
2.1.3	Operators . . . . .	6
<b>3</b>	<b>Requirements</b>	<b>7</b>
3.1	Functional Requirements . . . . .	9
3.2	Non-functional Requirements . . . . .	9
3.2.1	User Interface . . . . .	9
<b>4</b>	<b>Scenario Identifying</b>	<b>10</b>
4.1	Scenario 1 . . . . .	10
4.2	Scenario 2 . . . . .	10
<b>5</b>	<b>UML Models</b>	<b>11</b>
5.1	Use Case Diagram . . . . .	11
5.2	Sequence Diagram . . . . .	11
5.2.1	Login . . . . .	11
5.2.2	Car Reservation . . . . .	11

# 1 Introduction

## 1.1 Description of the given problem

The purpose of this document is to support the development of a software that manages a car sharing services. The software that will be developed has the aim to make the reservation of cars simple and quick. So the system should provide users with real time information about availability of cars, and their positions. Cars will be parked in some pre-defined parking area, where user can directly get them after the reservation. The service will be provided only to user that have made the registration, giving some personal information and data needed to the payment. The price of the ride is computed with a fixed amount of minutes, displayed by the car, and finally charged. To avoid useless reservation, where a user doesn't pick up the car, the system should be able to detect that after a fixed time, re-tag the car available, and charge with a fee the user. Cars must be locked in the safe areas, and unlockable only by users that has made a reservation on them.

## 1.2 Glossary and abbreviations

- VIRTUOUS BEHAVIOUR: are behaviour that makes simple the maintenance of the cars and reservation areas, but also helps to reduce the traffic impact.

## 1.3 Actors

These are the people that will be involved in the use cases of the software to be developed:

- GUEST: a person that hasn't made the registration to the service.
- USER: a person that has already registered to the system.
- OPERATOR: an employee of the company that takes maintenance of cars and charging area.

## 1.4 Assumptions

There are few points that aren't well specified in the assignment document. This section's purpose is to formalize some facts that complete the specification.

- Also GUESTs are able to search cars, in order to induce them to create an account.
- There are no information about cars' maintenance, so in the document is made the assumption that cars need OPERATOR presence to keep an acceptable status.
- USERS can't delete reservations.

- The system relies on external payment methods. The accepted methods are Mastercard, Visa, PayPal, PostePay.
- If a payment fails for a insufficient money availability of the user, the USER is suspended from the service until the payment is done.

## 1.5 Domain assumptions

- [D1] The car positions provided by GPS are accurate,
- [D2] The USER and GUEST position provided by GPS are accurate.
- [D3] The cars provide remote communication API
- /\*se un utente apre sale, va dove vuole andare e non si schianta\*/
- /\*non danno in giro la password\*/

## 1.6 Goals

The objectives of the software to be developed are the following:

- [G1] USERS must be able to reserve a car and use it.
  - [G1.1] USER should be able to reserve a car.
  - [G1.2] USER should be able to unlock reserved car when they are close to it.
  - [G1.3] USER should be aware of how much they are going to pay during the ride.
- [G2] The reservation of two (or more) cars at a time must be forbidden.
- [G3] USER and GUEST must be able to search cars.
  - [G3.1] USER should be able to search car near his position.
  - [G3.2] USER should be able to search car near a selected position.
- [G4] Induce USER to keep a VIRTUOUS BEHAVIOUR.
  - [G4.1] A discount of 10% should be applied to rides with at least two passengers.
  - [G4.2] If a car is left with no more than 50% battery empty, a discount of 20% should be applied to the last ride.
  - [G4.3] If a car is left at more than 3 km from the nearest charging area, the system should charge 30% more on the last ride.
  - [G4.4] If a car is left with more than 80% battery empty, the system should charge 30% more on the last ride.

- [G4.5] If a car is left in a charging area and the user plugs the car into the power grid, a discount of 30% should be applied to the last ride.
- [G4.6] If a car is not picked up within one hour from the reservation, the user pays a fee of 1 EUR.
- [G4.7] If a payment fails for a insufficient money availability of the USER, the USER should be suspended from the service until the payment is done.
- [G5] USER have to pay an amount of money based on the ride's duration.
- [G5] OPERATOR must be able to do car maintenance, knowing their position and status.

## 1.7 Document overview

/\*here we will describe the document contents organization\*/

## **2 Overall description**

### **2.1 Product perspective**

In this section the system boundaries are described, like the interfaces provided by external systems that the software to be developed will use in order to provide the required functionalities.

#### **2.1.1 Payment methods**

In order to manage in a good way the payment process, and to ensure a well managing of exceptional situations like users unavailability of money, the software to be developed relies on external transactional system. So the system doesn't care about how the payments are done, but only that the USER has provided one of the payment methods defined in the assumption section.

#### **2.1.2 Car technology**

#### **2.1.3 Operators**

### 3 Requirements

In order to satisfy goals in section [Section Goals] under domain assumptions in Section [Section domain Assumption] we've derived requirements for our system.

- [G1.1] USER should be able to reserve a car:
  - [R1] The system can modify car's status (available, occupied).
  - [R2] The system shall reserve a car only if the car is available.
  - [R3] The system shall be able to associate a car to the USER who has reserved it.
- [G1.2] USER should be able to unlock reserved car when they are close to it.
  - [R4] The system shall know cars' location according to cars' GPS.
  - [R5] The system shall know user's location according to his GPS.
  - [R6] The system shall provide a functionality to unlock the car.
  - [D1]
  - [D2]
  - [D3]
- [G2] The reservation of two (or more) cars at a time must be forbidden.
  - [R7] The system shall reserve a car only if the user hasn't already reserved another car.
  - [R3]
- [G5] USER have to pay an amount of money based on the ride's duration.
  - [R7] The system shall be able to calculate ride's cost based on duration.
  - [R8] The system shall commission the payment to external payment service.
  - [R9]
  - [D4]
- [G4.7] If a payment fails for a insufficient money availability of the USER, the USER should be suspended from the service until the payment is done.
  - [R9] The system shall take care of the result of the payment.
  - [R10] The system shall suspend USER that can't afford the payment.

/\* TODO \*/

- [R1] The system shall provide Sign Up functionality.

- [R2] The system shall send the password to the email provided by the user.
  - [R3] The system shall require user’s credit card.
  - [R4] The system shall require user’s driving license information.
- [G2] Login of a registered USER:
  - [R
- [G2] Searching for cars near a selected position:
  - [R5] The system shall provide the closest cars to position provided by the USER.
  - [R6] The system shall be able to detect cars’ position.
- [G3] Searching for cars near USER’s position:
  - [R7] The system shall provide the closest cars to USER’s position.
  - [R8] The system shall be able to detect USER’s position according to GPS.
  - [R6](3)
- [G5] USER must be sanctioned if he doesn’t pick up the car within one hour from reservation:
  - [R13] The system shall be able to charge USER by his payment method.
- [G6] Car reservation expires after one hour if the user does not pick it up:
  - [R9]
- [G7] USER that has reserved a car can unlock it if he’s sufficiently closed to it:
  - [R14] The System shall be able to manage machine hardware communicating with car.
- [G8] USER must be aware, during the ride, of how much he’s going to pay:
  - [R15] Car provide a way to show ride’s price (is this a domain assumption?)
  - [R16] System shall be able to compute ride’s cost according to information received from the car.



### 3.1 Functional Requirements

After having defined main features of our system we can identify some functional requirements grouped under each defined actor:

- GUEST, he can:
  - Sign up.
  - Viewing cars' map.
- USER, he can:
  - Log in.
  - Reserve car.
  - Use a reserved car (unlocking it).
  - View his profile (also his discounts).
  - Modify his profile.
  - View his recent movement.
  - Search car near to his location.
  - Search car near a selected location.
- OPERATOR, he can:
  - View car's status (battery and location).
  - Set a car as available or under maintainance.

### 3.2 Non-functional Requirements

#### 3.2.1 User Interface

In order to make PowerEnJoy available for many people as possible a Cross-Platform application has to be developed, with a Framework like Xamarin. In this way could be developed a single application and then could be deployed for Windows Phone, Android and iOS. A Web Application has also to be developed in order to allow people to use our system in many way as possible.

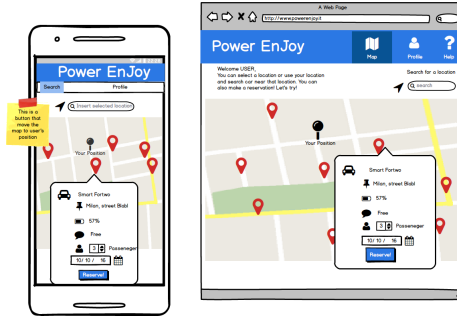


Figure 1: Map UI

## 4 Scenario Identifying

### 4.1 Scenario 1

Paul, a businessman, has to go to Milan by train in order to participate to an important meeting. As the office where they have to meet is 5 km far (away?) from the railway station and taxis are too much expensive, he decides to use *Power EnJoy* from his smartphone. He notices, thanks to the map on the app, that there is a parking area next to the railway station with a lot of available cars. He then reserves one of these electric cars so that, as soon as he arrives there, he'll be able to reach the office very quickly and in a green way.

### 4.2 Scenario 2

Paul is a students that goes to Politecnico University in Milan. He live in a city near Milan but unfortunately train's schedule is very prohibitive: he should wake up at 5 o'clock in the morning in order to arrive on time at the lecture that starts at 8:00.

He found out about PowerEnJoy from two of his friends in his city that attend to the same University. They decide to use together car-sharing service so they can have save much money using the two passeggers discount.

Since their city is full of electric-car they every day make a reservation of a car and go to the University together.

In this way they save money and also are much more comfortable in reaching the University.

## 5 UML Models

### 5.1 Use Case Diagram

With the Use Case Diagram are highlighted the main function of our system and the interaction between ACTORS and the related Use Case.

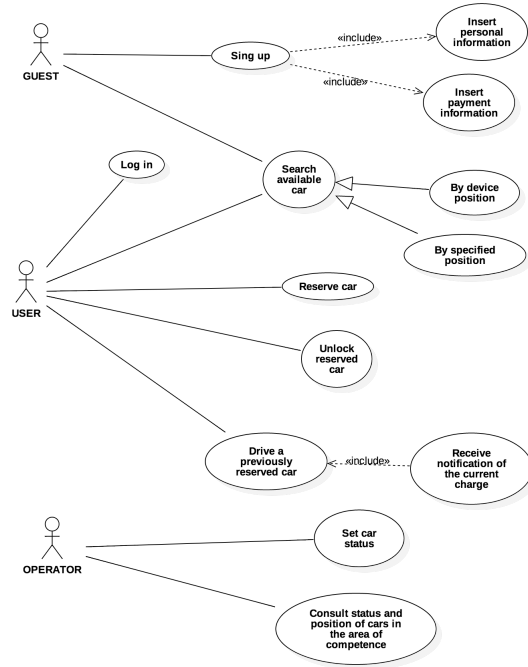


Figure 2: Use Case Diagram

### 5.2 Sequence Diagram

With Sequence Diagram are highlighted the main function of the system and the High-Level sequence of actions that focuses on the way in which happens the interaction between ACTORS and system.

#### 5.2.1 Login

#### 5.2.2 Car Reservation

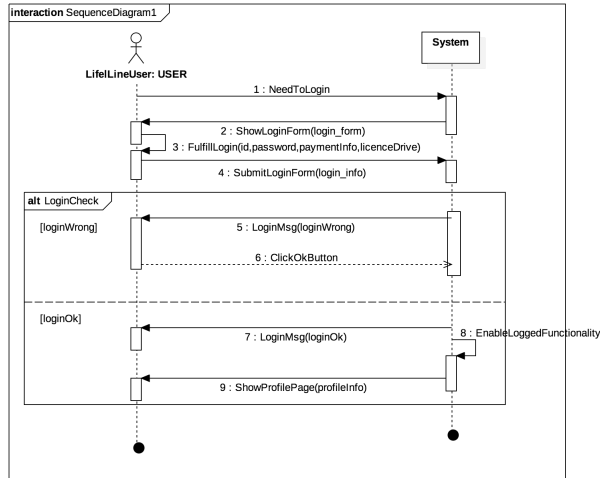


Figure 3: Login

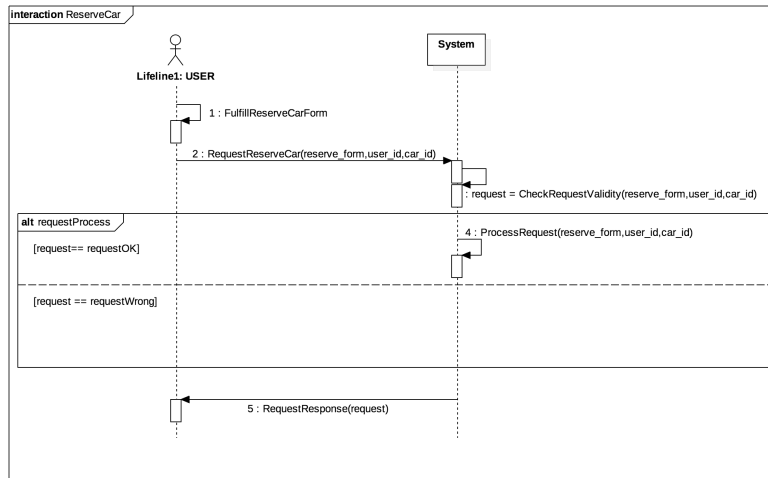


Figure 4: Car Reservation