



## Requirement Analysis and Specification Document

Patricia Abbud  
Maddalena Andreoli Andreoni  
Paolo Cudrano

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Description of the given problem . . . . .	3
1.2	Actual system . . . . .	3
1.3	Scope . . . . .	3
1.4	Actors . . . . .	3
1.5	Goals . . . . .	3
1.6	Domain properties . . . . .	4
1.7	Glossary . . . . .	5
1.8	Assumptions . . . . .	7
1.9	Constrains . . . . .	7
1.9.1	Regulatory policies . . . . .	7
1.9.2	Hardware limitations . . . . .	7
1.9.3	Interfaces to other applications . . . . .	8
1.10	Proposed system . . . . .	8
1.11	Identifying stakeholders . . . . .	8
1.12	Reference documents . . . . .	8
<b>2</b>	<b>Actors identifying</b>	<b>9</b>
<b>3</b>	<b>Requirements</b>	<b>10</b>
3.1	Functional requirements . . . . .	10
3.2	Non-functional requirements . . . . .	13
3.2.1	User interface . . . . .	13
<b>4</b>	<b>Scenario identifying</b>	<b>19</b>
4.1	Scenario 1: Registration from the website . . . . .	19
4.2	Scenario 1: Registration from the mobile application . . . . .	19
4.3	Scenario 2: Login in the app . . . . .	19
4.4	Scenario 3: Reserving and using a car . . . . .	19
4.5	Scenario 4: Parking and regular fees . . . . .	20
4.6	Scenario 5: Power grid and discounted fees . . . . .	20
4.7	Scenario 6: Parking and sanctioned fees . . . . .	20
4.8	Scenario 7: Lost reservation . . . . .	20
4.9	Scenario 8: Two passengers + special parking area . . . . .	20
4.10	Scenario 9: Money saving option . . . . .	21
4.11	Scenario #: Left the city . . . . .	21
4.12	Scenario #: Recover a car with user input . . . . .	21
4.13	Scenario #: Recover a car without user input . . . . .	21
4.14	Scenario #: User breaking the law . . . . .	21
<b>5</b>	<b>UML models</b>	<b>22</b>
5.1	Use case diagram . . . . .	22
5.1.1	Use case 1: Reserve a car . . . . .	22
5.1.2	Use case 2: Park in known safe area . . . . .	23

5.1.3	Use case 3: Park with money saving option . . . . .	24
5.1.4	Use case 4: Park in a recharging area . . . . .	25
5.1.5	Use case 5: Retrieve a car with user input . . . . .	25
5.1.6	Use case 6: Manually assist a parked car . . . . .	27
5.1.7	Use case 7: Manage infractions . . . . .	28
5.1.8	Use case 8: Assist user after an accident . . . . .	28
5.1.9	Use case 9: Assist a user on-site after a car breakdown . .	29
5.1.10	Use case 10: Assist a user with a not on-site reparation .	31
<b>6</b>	<b>Alloy modeling</b>	<b>32</b>
<b>7</b>	<b>Appendix</b>	<b>33</b>
7.1	Used tools . . . . .	33
7.2	Hours of work . . . . .	33

# 1 Introduction

## 1.1 Description of the given problem

The system we are going to develop is a car-sharing service called *PowerEnJoy*, based on mobile application and which has a website. The system allows users to locate and reserve a car to drive in their geographical region using mobile application. Geographical region is identified using GPS on user's smartphone, he can also insert it manually. User can drive a car only inside one chosen geographical region. People in the system are not considered as registered until they fill an online form on the website or on the main page of the mobile application and until their driving license are not checked by the system as valid ones.

The main purpose of the system is to be efficient, reliable and easy to use to fulfill the will of users and the customer.

## 1.2 Actual system

There is nothing created until now, and we suppose that we have to create the entire application, depending only on the problem given by the customer.

## 1.3 Scope

## 1.4 Actors

- Guest: all guests can only see the login page of the application or of the website, where they can complete the registration form to be able to login from mobile application as registered users.
- Registered User: after logging in to the system users can "use" (change this word) the application, where they can see map with available cars, reserve one and get notifications from the system and also can see their profile and make some changes in it.

## 1.5 Goals

1. The system allows guests to register; to complete the registration procedure the system sends a password to the guest as an access key.
2. The system should enable a registered user to find the location of an available car within a certain distance from the user's location or from a specified address.
3. The system enables user to reserve a single available car in a certain geographical region for one hour before the user picks it up. If the car is not picked up by that time, the reservation expires, the system tags this car as available again and it charges the user a fine of 1 EUR.

4. The system should allow the user to employ a car in a proper and safe way.
5. The system charges the user for a predefined amount of money per minute. A screen on the car notifies the user of the current charges.
6. The system starts charging the user as soon as the car ignites. It stops charging them when the car is parked in a safe area and the user exits the car. The user must confirm the operation, otherwise the system keeps charging them.
7. The system should encourage good user behaviour through the application of discounts to the fee per minute.
8. The system should discourage bad behaviour through the application of sanctions to the fee per minute.
9. The system should provide an alternative usage mode for cars called *money saving option*. Besides aiding the user in saving money, this mode allows for a uniform distribution of cars throughout the city by suggesting the user where to park.
10. The system allows the company to assist the users in case of need and take care of the cars.
11. The admin should be able to configure some parameters of the system.

## 1.6 Domain properties

We assume that the following properties hold in the analyzed world:

- A tablet is permanently installed in every car.
- The tablet is alimented through the car and cannot be switched off.
- All cars are equipped and located with a GPS system provided by the tablet.
- All GPS systems always give the right position.
- GPS tracking is always on.
- All cars has sensors to detect the presence of passengers in each seat.
- The system is always able to tell how many people occupy a car.
- All cars are equipped with a Bluetooth system provided by the tablet.
- The Bluetooth system is always on.
- All cars are equipped with an integrated alarm as theft deterrent.

- All cars are equipped with an accident detection system which is always able to notify the company of an accident when the driving user can't.
- A user can provide their location with their phone's GPS whenever they want.
- The safe areas are predefined and within the municipality of Milan.
- The payments of all services are managed by an external company, which guarantees fulfillment.
- The cars always ignite when they have more than 3% of power charge.
- There is no policy for time limits for a user to retain and use a reserved car.
- To complete the registration procedure, the user must send their driving license and ID as pdf documents to the company email address.
- Localization services and maps are provided by an external company, which guarantees precision.

## 1.7 Glossary

**User** We will refer to all people who are registered to the system as 'users'. All users have personal profiles which contain the following information:

- First name;
- Family name;
- Email;
- Username;
- Password;
- Payment information; this in particular includes:
  - Credit card number;
  - Credit card expiration date;
  - CVV number.
- Driving license information; this in particular includes:
  - License number;
  - Issued date;
  - Expiration date;
  - Status.

And, optionally:

- Personal photo;
- Telephone number.

Users should be able to locate, reserve and drive the cars offered by the service.

**Guest** We shall call 'guests' all people who are using the interface of the system without being registered or logged in. Guests can't access any functionality of *PowerEnJoy* except for the registration process or the log in.

**Parking areas** Also called *Safe areas*, parking areas are predefined parking slots within the municipality that are reserved for the car-sharing system *PowerEnJoy*.

**Special parking areas** , or *recharging areas* are *Parking areas* that provides a plug to recharge the car. Recharging parking areas are a subset of Parking areas (parking areas *may* be recharging areas, while the contrary doesn't apply).

**Reserved car** We will call 'reserved car' a specific car that is booked by current user, not taken by other users, and is located in the same geographical region as him.

**Available car** is the car that is not reserved by other users.

**Power grid**

**Standard price** We shall call 'standard price' the price per minute charged to the user, without any discount or sanction applied.

**Discount** A discount is a deduction of a given percentage from the standard price. It is applied every time a user has a virtuous behavior.

**Sanction** A sanction is an increase of a given percentage to the standard price. It is applied every time a user has a wasteful or incorrect behavior.

**Money saving option** An option offered to the users by the system. The user inputs their final destination and the system indicates them the best close station where to leave the car.

**Time Window**

**Standard ride** We define as *standard ride* every ride that finishes with the car being parked in a safe area and where no emergency has occurred.

## 1.8 Assumptions

The assignment document was unclear and ambiguous on some points of the specifications. Hence, we will make the following assumptions:

- Parking areas and special parking areas are two different things; however, common sense suggests that it isn't logical to charge users if they plug the car to the power grid in a recharging area but are not parked in a

safe area. Neither it makes sense to sanction them if they park it in a safe area that is 3 km far from the power grid. Having the two areas separate would lead to the consequences above, so we decided that while a safe area may not be a recharging area, recharging areas are always safe areas. Furthermore, assuming that a city has many safe areas, so that users can enjoy the service all around the city, it is reasonable to think that just a few of them have been equipped with a power grid connection, for economic and infrastructural reasons.

- The users are able to reserve an available car only from their geographical region.
- There is a "manual" way to close and open the car, i.e. that the user is allowed to temporarily park the car – while still being charged –, get out, close the car, and then get back and open it again. This means that the system can be used not only for one-way travels, but also when more stops or a round trip is needed.
- Parking areas and special parking areas are allotted and private parking spaces owned by *PowerEnJoy* and distributed throughout the urban area.
- We assume that all cars are the same model and have the same features; specifically, they are all 5-door Citroën C-Zero Micro Car with four seats, customized for the purposes of *PowerEnJoy*.
- If there is at least one passenger, the system cannot infer if the driver is actually the user or the user is the passenger and someone else is driving. There is no way of knowing that; however, we assume that upon registration the user has accepted the policy that asks them to be the only driver. If they do not comply to that and commit some infractions, the company reserves the right to take legal action.

## 1.9 Constrains

### 1.9.1 Regulatory policies

According to privacy law the system must require to User the permission to get his position and to acquire, store and process his personal data.

### 1.9.2 Hardware limitations

- Mobile application
  - Space for app package (30MB)
  - 3G connection
  - GPS
- Website



- Modern browser The Website of the service must work with all versions of Google Chrome, Opera, Firefox and Internet Explorer released after 2010.

### **1.9.3 Interfaces to other applications**

*PowerEnJoy* has

### **1.10 Proposed system**

#### **1.11 Identifying stakeholders**

Main stakeholders are the inhabitants of Milan willing to use an ecologic and comfortable mean of transport, without the need to buy and maintain a personal car. The system can be adopted in any other city, so the inhabitants of other cities too can become stakeholders. In Milan, the project is promoted and participated by the city council as minor investor.

#### **1.12 Reference documents**

## 2 Actors identifying

The system involves as main external actors:

**User** The final target of the system. Once registered, they can reserve a car and use it in respect of the agreements established with *PowerEnJoy*. They are charged for the usage of the car. It is expected

**Guest** A potential user not registered in the system yet.

The following actors have also been identified inside *PowerEnJoy* as needed by the system:

**Admin** Back-office administrators, they have access to a control panel.

**Operator** Maintenance operators who provide field-based assistance and ensures cars are always ready to be used.

**Car** The car provided to users. It interacts with the system to be develop to provide information about usage and status and to receive commands.

The system has moreover to interact with the following external service provider:

**Payment service** Provider of every service related to users payment management.

**Local police** Local law enforcement agencies such as *Polizia locale* or *Cara-binieri* that interacts with the system for legal issues.

## 3 Requirements

### 3.1 Functional requirements

We assume that all domain properties stipulated in paragraph 1.6 hold. We decided to use a *goal-driven* method to structure our requirements, meaning that we decomposed the high-level goals of paragraph 1.5 into low-level requirements. Some of these requirements stem directly from the requests of our clients, while others are born from the necessity to have a sound system.

For each goal, we derive the following requirements:

- $G[1]$  The system allows guests to register; to complete the registration procedure the system sends a password to the guest as an access key.
  - The system has to allow any person to submit only one account request.
  - The account is created when an admin validates all the necessary data.
  - The system must be able to generate passwords.
  - The system has to send a newly generated password to the user via email when the account is created.
  - The system must be able to check whether a password is correct or not.
  - The system must let the user log in only if the password is correct.
  - The system has to generate a new password and send it via email if the user asks for it.
- $G[2]$  The system should enable a registered user to find the location of an available car within a certain distance from the user's location or from a specified address.
  - The system must have the ability to locate the user.
  - The system must be able to locate any valid address.
  - The system must be able to find any of the parking areas of the company.
  - The system must be able to identify available cars inside parking areas.
  - The system must let users see whether there are available cars in a specified radius.
- $G[3]$  The system enables user to reserve a single available car in a certain geographical region for one hour before the user picks it up. If the car is not picked up by that time, the reservation expires, the system tags this car as available again and it charges the user a fine of 1 EUR.

- The system must allow users to reserve an available car.
- The cars cannot be reserved by more than one user at any given time.
- The system must keep the current reservation standing until the user has opened the car or an hour has passed.
- The system must be able to autonomously cancel reservations.
- The system must impede any user with an expired license to reserve a car.
- The system must impede any banned user to reserve a car.
- *G[4]* The system should allow the user to employ a car in a proper and safe way.
  - The system must be able to locate any car at any given time.
  - The system must be able to detect whether there are passengers inside a car, and how many.
  - The system must be able to collect data about the power charge of any of its cars.
  - The system must be able to detect when a severe accident has occurred to a car.
  - The system must be able to detect when a user is near a car.
  - The system must be able to tell when a car is parked in a safe area.
  - The system must be able to detect when a car is plugged to the power grid.
  - The system must be able to detect whether the driver is still in the car.
  - The system must be able to automatically unlock a car when the user that has reserved is nearby.
  - The system must be able to automatically lock a car when the user has exited it inside a safe area.
  - The system must allow the user to lock and unlock their car manually when outside a safe area.
  - The system must provide a finite time window that begins when the user exits the car inside a safe area. The time window must either end when the allotted time is finished or when another user reserves the same car.
  - The system must allow the user to re-enter the car while the time window is still open.
- *G[5]* The system charges the user for a predefined amount of money per minute. A screen on the car notifies the user of the current charges.

- The system must be able to retrieve all data necessary to charge the user. This data is the duration of the ride and all the conditions for application of discounts and sanctions.
- The system should notify the user of the fee per minute he's being charged through a screen inside the car.
- The system must notify the final charges to the user after the time window for the ride has expired.
- The system must invoice the user after the time window for the ride by communicating the charges to the external payment system.
- *G[6]* The system starts charging the user as soon as the car ignites. It stops charging them when the car is parked in a safe area and the user exits the car. The user must confirm the operation, otherwise the system keeps charging them.
  - The system must charge the user with a fee per minute.
  - The system must be able to tell when the car has ignited.
  - The system should start counting the charges from the ignition onward.
  - The system must ask the user whether they wish to end the ride or not as soon as they are parked in a safe area and have exited the car.
  - If a user has chosen to end the ride, the charges must be counted until the beginning of the time window for that ride.
  - If a user keeps using the car, the time window is included in the charged time.
  - If the reservation has expired but the user is inside the car (without igniting it), the system must start charging them anyway an exceptional fee per minute.
- *G[7]* The system should encourage good user behaviour through the application of discounts to the fee per minute.
  - The system must apply all discounts to the fee per minute.
  - If the conditions for a discount only count for a limited period of time during the ride, the discount should be applied only to that time period.
  - The system must apply a discount every time a user brings two or more passengers in the car with them.
  - The system must apply a discount every time the car has more than 50% of power charge by the end of a standard ride.
  - The system must be able to apply a discount every time the user plugs the car in the power grid while their time window is still open.

- *G[8]* The system should discourage bad behaviour through the application of sanctions to the fee per minute.
  - The system must apply a sanction every time a car is returned in a safe area at more than 3 km from the nearest recharging safe area.
  - The system must apply a sanction every time a car is returned with less than 20% of power charge.
- *G[9]* The system should provide an alternative usage mode for cars called *money saving option*. Besides aiding the user in saving money, this mode allows for a uniform distribution of cars throughout the city by suggesting the user where to park.
  - The system must always allow a user to select the *money saving option* at any point during their ride.
  - The system must apply a discount every time the car is returned in the suggested safe area with the *money saving option*.
  - The system must select the safe area suggested to the user based on consideration of availability of power plugs and of distribution of the cars. This safe area must be as close as possible to the user's destination.
- *G[10]* The system allows the company to assist the users in case of need and take care of the cars.
  - The system must always allow a user to notify the back-end administrators if they have a problem with the car.
  - The system must always be able to locate *PowerEnJoy* operators.
  - The system must allow administrators to know the status of on-site operators and dispatch them if they are available.
  - The system must keep track of every car's battery and alert the admin when it lowers below 3%.
- *G[11]* The admin should be able to configure some parameters of the system.
  - The system must allow the administrators to modify parameters that influence the usage of cars. This parameters are:
    1. The standard fee per minute
    2. The exceptional fee per minute
    3. The lost reservation fee
    4. The fine for leaving an accident site
    5. Discount percent values
    6. Sanction percent values
    7. Time window for reservation fee
    8. Time window after ending the ride for reopening and plugging the car.

## 3.2 Non-functional requirements

### 3.2.1 User interface

The interface of our application is thought to be used mainly via mobile app, with some content also displayed in a web page. The team reasoned that the main functionalities of the system (such as reserving and managing a car) make sense only in a *movable* context (meaning, such that the user can do them anywhere they have an internet connection). Those functionalities are available on the mobile app. On the other hand, operations such as signing up are more easily managed in front of a computer, so the web page allows users to register, login and manage their profiles (which they can do also via mobile app anyway).

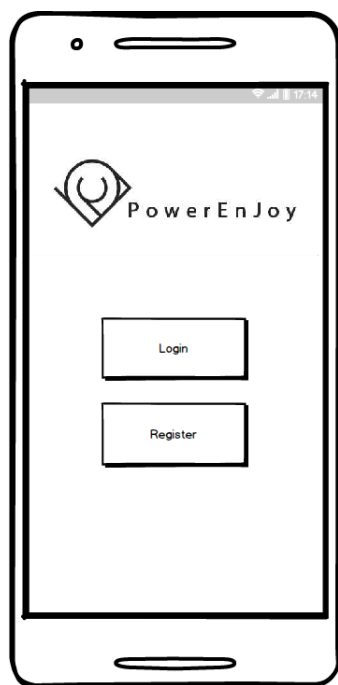
We pondered on the possibility of adding the possibility of reserving a car via web browser. However, upon consideration, we decided it is not a good mechanism: with that functionality, a user could very well be registered without having downloaded the app and reserving a car without being able to access it in any way. Hence, the reservation of *PowerEnJoy* cars can be made only via mobile app.

#### Mobile app

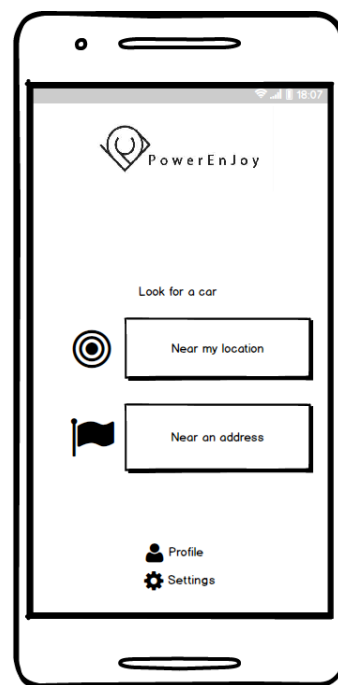
Figure 1 shows the first page that is shown when entering the app. Picture 1.a is the guest view, who has only the possibility of either logging in or registering in the system. Picture 1.b shows the user view. The user has more functionalities: they can look for cars in their vicinity or near an address, see and edit their profile, and changing settings (notification and sound settings).

Figure 2 shows the main functionalities of the *PowerEnJoy*'s app: reserving a car (2.a and 2.b) and what you can do while using a car (2.c).

Figure 3.a and 3.b shows, respectively, the website homepage from the standpoint of a guest and the sign up page. We haven't drawn the website mockup from the standpoint of a registered user since there are no more functionalities than those already present in the mobile app, as said above.



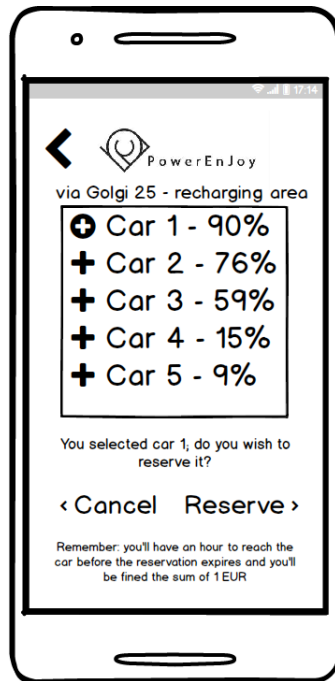
(a) Guest view



(b) User view

Figure 1: Mobile app: home page view

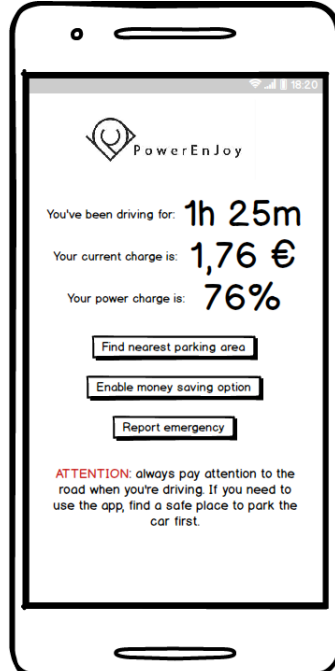




(a) View for the reservation



(b) View after having selected a parking area



(c) Display of the app while using a car

Figure 2: Mobile app: main functionalities



(a) Website homepage

(b) Sign up on website

Figure 3: Website

## **Documentation**

In order to keep track of all phases of the development process and the overall structure of the system, the team will release the following documents:

**RASD** , Requirement Analysis and Specification Document, which provides a thorough description of the system, the requirements and the specification thanks to the use of UML models.

**DD** , Design Document, which contains a more in-depth description of the functionalities of the system.

**ITPD** , Integration Test Plan Document, which describes integration tests and the team's intended plan to accomplish them.

**PP** , Project Plan, which defines a planning for the development project.

## 4 Scenario identifying

### 4.1 Scenario 1: Registration from the website

Anakin has just moved to Milan and has rented a flat; however, he couldn't afford a place close to the city centre, where he works; he also doesn't have a car, so he's been going back and forth with public transport. Because of that, he needs to wake up half an hour earlier and usually gets home very late, and he's getting tired. He then decides to look for a solution on Google, and he finds the car-sharing service of *PowerEnJoy*, which has a parking place close to his home. The *PowerEnJoy* web page has all the information readily available, pricing, features, an approximated map of the parking areas included and a link to download the application from suitable store, so Anakin decides to sign up. He completes a form, where he writes his complete name, personal information, information about his driving license, credentials and payment; the system checks Anakin's driving license and only then sends him the password, so he can login from the phone application and access the private area of the system.

### 4.2 Scenario 1: Registration from the mobile application

Chewbacca was advised the *PowerEnJoy*, so he finds the application on the mobile appstore and downloads it directly to his phone. Opening the application he sees the main page where he can register just as his friend Anakin.

### 4.3 Scenario 2: Login in the app

Padmé has registered on the *PowerEnJoy* website and now has downloaded the application on her smartphone. Opening the application, she finds a screen asking her to log in. Padmé registered with *Naboo-princess* username, and the password she received in her email is *7aKmm93s*, so she fills the login form with this information. The first time she writes the password wrong, so the login is rejected and the application asks her to try again. The second time she typed the password right, so the system accepts the login and shows the main page of the application.

### 4.4 Scenario 3: Reserving and using a car

Luke must reach his aunt and uncle for the usual sunday roast. He doesn't have a car, and usually he just takes the subway. However today the public transport workers are on strike, and the metro is out of order. Luke is a distracted kid, always with his head in the clouds, so he forgot about the strike and has walked for the fifteen minutes needed to reach the metro. He is nevertheless smart and resourceful, and so he remembers that he's signed in the *PowerEnJoy* system. He opens the app and presses the "find car" button. He has the GPS activated, so the system locates him and tells him that there's a parking area with an

available car next to the metro station. The application gives him the choice of reserving the car or cancel the operation, and Luke reserves the car.

#### **4.5 Scenario 4: Parking and regular fees**

Leia needs to get to the american consulate ASAP. She's a frequent user of the car-sharing service, so she already knows all the safe areas where she can park around the diplomatic block in the city centre, since she often needs to go there. While she drives, the smart display in her car tells her how much the system is currently charging: the fee per minute is 0,50 EUR, and she's been driving for half an hour, so her fee currently is 15 EUR. She reaches the diplomatic area after two more minutes, and she knows that the parking area is around the corner from the consulate, so she reaches there. This particular safe area is not a recharging area, and Leia left the car with a 60% battery full, so the system charges her 16 EUR. The display notifies her that she's parked in a safe area and that no sanction applies to her fee. The system also asks her if she wants to keep the car reserved and keep being charged, and she declines. She exits the car and the system locks it automatically.

#### **4.6 Scenario 5: Power grid and discounted fees**

#### **4.7 Scenario 6: Parking and sanctioned fees**

Han, Leia's husband, is also a regular user of the system. He's however less abiding to rules and is a bit of a free spirit, so he generally never gains any discount and is often sanctioned for wasteful behaviour. For example, last monday he needed to reach his bank in Porta Genova for a meeting with his broker. He was very late, so he parked right outside the bank, outside any safe area. It took him twenty minutes to get there, so the system had charged him 10 EUR. As soon as he stops the car, the display notifies him that he is in an unsafe area, so he'll keep being charged even if he isn't using the car. Han confirms and exits. The car doesn't lock automatically, so he needs to turn his bluetooth on and close it with the *PowerEnJoy* application.

#### **4.8 Scenario 7: Lost reservation**

Han and Leia want to go a restaurant. Han thinks that they are about to leave the house, so he decides to reserve a car while still at home to make sure they will have a car after leaving home, but it takes Leia more than an hour to get prepared. In one hour they are not near the car to open it, so the system cancels Han's reservation, notifies him about the 1 EUR sanction charged for uselessly reserving it and asks if he wants to reserve another available car.

#### **4.9 Scenario 8: Two passengers + special parking area**

Chewbacca met Han and Leia on his way, so he invited them to take one car. When they sit in the car, Chewbacca receives a notification on the display

that two other passengers are detected. When they arrive to their destination, Chewbacca sees on the application's map that there is a special parking area near them, therefore he decides to put the car there and doesn't forget to plug in the car to the power grid. At the end of the ride Chewbacca gets a discount of 40% (10% for the 2 passengers + 30% for leaving the car on the special parking area and plugging it to the power grid).

#### **4.10 Scenario 9: Money saving option**

#### **4.11 Scenario #: Left the city**

#### **4.12 Scenario #: Recover a car with user input**

Obi-Wan is driving peacefully to reach his yoga instructor, Qui-Gon Jinn, when the car breaks down; he tries to restart it or check what is wrong, but the system doesn't detect any anomaly and he isn't a mechanic, so he decides to notify the company: he opens the app on his phone and enters the *Emergency help* section. The app asks him to broadly describe the problem by filling a form. He then selects the option "car doesn't work" and the system tells him that an operator is on his way. In the meantime the system, having located the car, notifies the operator of its whereabouts. The operator – which is also a mechanic – goes there with a truck. He checks the car on site and then, since the car needs more serious repairs, he loads it on the truck. The operator informs the system that the breakdown was not Obi-Wan's fault, so Obi-Wan's charges are dropped for faulty service. The system then offers him to reserve another car in a safe area nearby.

#### **4.13 Scenario #: Recover a car without user input**

An unnamed user has left a car with 0% battery in an area far from the power grid. As soon as the emergency battery kicks in, the system locates the car and alerts an operator of the situation. In this case, two operators are dispatched on the same truck: when they reach the car, they try to charge it on site. If they manage to do so, one operator drives the car to the nearest recharging area while the other follows with the truck. If they don't, they load the car on the truck and transport it to the nearest recharging area.

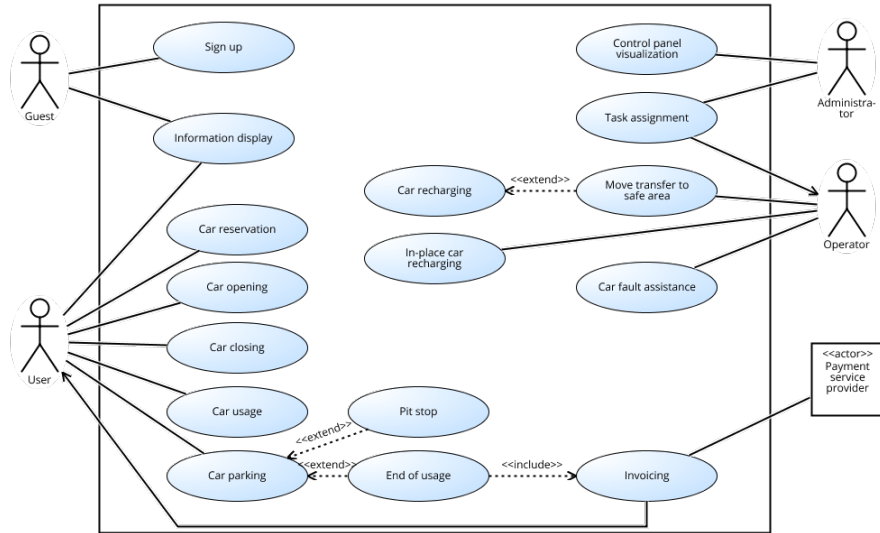
#### **4.14 Scenario #: User breaking the law**

Darth Maul has broken the law while driving a *PowerEnJoy* car: he was driving too fast in a residential area. Since the police has identified the car plate, the notification of the fine is sent to the company. The company pays the fine in advance. The system administrator then checks who was driving that specific car at that time and discovers it was him: he then charges the amount of the fine to Darth Maul through his preferred method of payment.

## 5 UML models

### 5.1 Use case diagram

A global picture of the system interaction with actors is provided here by means of use case diagrams. Following, an analysis of the most interesting use case situations derived from scenarios is presented.



#### 5.1.1 Use case 1: Reserve a car

**Name** Reserve a car

**Actors**

**User** The user who wants to reserve a car.

**Entry condition** The user decides to reserve a car to take within the next hour.

**Flow of events**

1. The user logs in into the mobile app and goes to the reservation section.
2. The system automatically retrieves and displays the location of the user, but the user can specify a different location if needed.
3. The system displays the position of the available cars close to the selected location.
4. The user selects a car and confirms the reservation.

**Exit condition** The system reserves the car for the user.

### Exceptions

- **The system is not able to locate the user automatically.** The user is required to insert a position manually.
- **The system is not able to find a position inserted manually.** The user is informed and the operation is aborted.
- **There are no available cars.** The user is informed and the operation is aborted.
- **The user cancels the operation before confirming.** The reservation process is not completed and the car remains available to other users.

**Special Requirements** None.

#### 5.1.2 Use case 2: Park in known safe area

**Name** Park in known safe area

### Actors

**User** The user of the car.

**Car** The car in use.

**Entry condition** The user is driving and has reached their destination. They know a safe area close to the destination.

### Flow of events

1. The safe area is free and the user parks in it.
2. As the car is turned off, the system detects it is in a safe area.
3. The system asks the user if he wants to keep the car or to end the ride.
4. The user selects to end the ride.
5. The user exits the car.
6. The system closes the car.
7. The system charges the user for the ride.

**Exit condition** The user leaves the car and the car becomes available to other users.

### Exceptions

- **The safe area is taken.** The user can't end their ride and this operation is aborted.
- **The car is badly parked.** ???
- **The user selects to keep the car when prompted.** The user keeps being charged and the car is not made available for other users.

**Special Requirements** None.



### 5.1.3 Use case 3: Park with money saving option

**Name** Park with money saving option

**Actors**

**User** The user of the car.

**Car** The car in use.

**Entry condition** The user selects the *money saving option* at some point of their ride and insert their destination.

**Flow of events**

1. The system indicates the user the suggested safe area for their destination.
2. The user parks in the suggested safe area.
3. As the car is turned off, the system detects it is in a safe area.
4. The system asks the user if he wants to keep the car or to end the ride.
5. The user selects to end the ride.
6. The user exits the car.
7. The system closes the car.
8. The system charges the user for the ride. A discount is applied for using the *money saving option*.

**Exit condition** The user leaves the car and the car becomes available to other users.

**Exceptions**

- **The suggested safe area becomes taken while the user is driving.** The system selects another safe area and notifies the user of the new suggestion.
- **The user parks in another safe area.** The system notifies the user that they will not receive a discount. If the user decides to end the ride anyway, the system charges them without applying the *money saving option* discount.
- **The destination of the user changes.** The user selects a new destination and the system indicates another suggestion.
- **The user disables the *money saving option* while driving.** The suggested safe area stops being displayed and the ride continues as normal.

**Special Requirements** The *money saving option* must be selected before stopping the car in a parking area.

#### 5.1.4 Use case 4: Park in a recharging area

**Name** Park in a recharging area

**Actors**

**User** The user of the car.

**Car** The car in use.

**Entry condition** The user is about to park in a recharging area.

**Flow of events**

1. The user parks the car in the recharging area.
2. As the car is turned off, the system detects it is in a safe area.
3. The system asks the user if he wants to keep the car or to end the ride.
4. The user selects to end the ride.
5. The user exits the car.
6. The system closes the car.
7. The system charges the user for the ride.
8. The user plugs the car into the power grid through the supply point installed in the parking space.
9. The system detects the car is recharging.
10. The system modifies the charge applied to the user for the ride. A 30% discount is applied to promote virtuous behaviors.

**Exit condition** The user leaves the car and the car becomes available to other users.

**Exceptions**

- **The user does not plug the car into the power grid.** The user is charged as if they parked in a safe area.

**Special Requirements** The user plugs in the car within 10 minutes from the moment they exits the car. Otherwise the discount is not applied.

#### 5.1.5 Use case 5: Manually assist a parked car

**Name** Manually assist a car

**Actors**

**Admin** The administrator who sends the operator.

**Operator** The operator sent to recover the car.

**Car** The car in use.

**Entry condition** The administrator is notified by the system that a parked car needs manual assistance.

**Flow of events**

1. The admin checks the issue the system is displaying. It can be one of the following:
  - the car is in a safe area without enough power charge to be used;
  - the car is in a recharging area without the plug inserted.
2. The admin assigns the maintenance work to an operator.
3. The operator accepts the assignment and the admin is notified of it.
4. When available, the operator performs the *maintenance operation* (see below).
5. The operator checks the assignment as completed.

**Exit condition** The maintenance activity has been performed and the admin is notified of the completion.

**Exceptions**

- **The operator is not able to perform the maintenance.** The assignment is marked as *not completed*, the cause is inserted into the system. The admin will be notified of it and will either assign the problem to another operator or manage it without the help of the system. The assignment will be anyway closed at the end of this process.

**Special Requirements** The operator cannot refuse an assignment if he is online and always accept it within a working day. An operator is always online when at work. The operator always marks the assignment as either *completed* or *not completed* before the end of the workday.

In the previous use case we refer to *maintenance operation* as to one of the following:

- **Issue:** the car is in a safe area without enough power charge to be used. **Maintenance operation:** the car is towed to a recharging area and plugged into the power grid once there.
- **Issue:** the car is in a recharging area without the plug inserted. **Maintenance operation:** the car is plugged into the power grid.

### 5.1.6 Use case 6: Manage infractions

**Name** Manage infractions

**Actors**

**Admin** The system administrator.

**User** The user responsible for the infraction.

**Entry conditions** The company receives by mail the notification of the infraction.

**Flow of events**

1. The company pays the fine for the infraction to the police.
2. The admin logs into the system and inserts the license plate and the time of the infraction to find the responsible user.
3. The system shows the name of the user.
4. The system charges the user with the fine.
5. The system notifies the user of the payment.

**Exit conditions** The legal procedure has been closed and the user has paid the fine.

**Exceptions**

- **The infraction causes the user to lose their license**

**5.1.7 Use case 7: Assist user after an accident**

**Name** Assist user after an accident

**Actors**

**User** The user driving the car.

**Admin** The administrator who receives the notification from the system.

**Operator** The operator sent to assist the user.

**Entry condition** The user is driving and is involved in an accident.

**Flow of events**

1. The User notifies the Admin that an accident has happened. In case of serious accidents, it is the Accident Detection System of the car, instead of the User, that notifies the Admin. After the notification, the User waits for an Operator to arrive.
2. The Admin dispatches an Operator with a tow-truck.
3. The Operator arrives on site.
4. The Operator manages the jointly-agreed statement for insurance purposes with other drivers involved and takes care of contacting the insurance company.
5. When everything is done on site, the Operator takes the car to the company garage, to be analyzed by the insurance company if needed.

6. The user leaves the site when the Operator takes the car away.

**Exit condition** Eventually, the car is repaired and the insurance company emits a result on the accident report and refunds the company for the reparation costs of the car. The car is put back in use (the car is taken to a parking area; it can be reserved again).

#### Exceptions

- **The User leaves the site before the removal of the car.** The Operator can report it to the system, and the user will be charged of an extra. If the User is penally implied in the accident, the Operator provides their personal details to the law enforcement.
- **The insurance asserts that the accident is fault of the User and doesn't take responsibility for it.** The User is fully charged of the reparation costs.

**Special Requirements** The Operator cannot refuse to be assigned to an accident if he is online and always accept it within 10 minutes. An Operator is always online when at work. The Accident Detection System always detects an accident when the User is unconscious. In this cases, it also notifies the emergency services.

#### 5.1.8 Use case 8: Assist a user on-site after a car breakdown

**Name** Assist user after a car breakdown

##### Actors

**User** The user driving the car.

**Admin** The administrator who receives the notification from the system.

**Operator** The operator sent to assist the user.

**Entry condition** The user is driving and notices a car breakdown. An empty battery while driving is considered as a car breakdown.

##### Flow of events

1. The User notifies the Admin that a breakdown has happened. After the notification, the User waits for an Operator to arrive. The charges for the User are suspended.
2. The Admin dispatches an Operator for an *on-site reparation*.
3. The Operator arrives on site.
4. The Operator repairs the car.
5. Once the reparation has ended, the User is assigned again to the car and the charges start again.

**Exit condition** The Operator marks the assignment as *completed* and leaves. The User is back on their car.

#### Exceptions

- **The Operator decides that the reparation can't be done on-site.** The use case *Assist a user with a not on-site reparation* is invoked.
- **The User leaves the site before the reparation ends.** The Operator reports it to the system. At the end of the reparation, the User is not assigned to the car anymore and is invoiced for the ride (until the breakdown occurred). The Operator notifies the Admin that the car must be taken to a parking area and leaves the site. The Admin dispatches an Operator with a tow truck. The Operator takes the car to the closest available parking area.
- **The Operator asserts that there is no need for intervention and the User is still on site.** The User is fined (parameter). The car is assigned to the User again and the charges start again.
- **The Operator asserts that there is no need for intervention and the User has gone from the site.** The User is fined (parameter, higher than in previous exception). The Operator notifies the Admin that the car must be taken to a parking area and leaves the site. The Admin dispatches an Operator with a tow truck. The Operator takes the car to the closest available parking area.

**Special Requirements** The Operator cannot refuse an assignment if he is online and always accept it within 10 minutes. An Operator is always online when at work.

#### 5.1.9 Use case 9: Assist a user with a not on-site reparation

**Name** Assist a user with a not on-site reparation

#### Actors

**User** The user driving the car.

**Admin** The administrator who receives the notification from the system.

**Operator** The operator sent to assist the user.

**Entry condition** The operator sent to fix a breakdown asserts that the reparation can't be done on-site and notifies the Admin of the change. The notification contains a brief description of the reason.

#### Flow of events

1. The User is not assigned to the car anymore and is invoiced for the ride (until the breakdown occurred). The car cannot be reserved anymore.

2. The Admin dispatches a new Operator with a tow truck.
3. The new Operator contacts the insurance company and notifies the breakdown.
4. The new Operator takes the car to the company garage.

**Exit condition** Eventually, the car is repaired and the insurance company emits a result on the accident report and refunds the company for the reparation costs of the car. The car is put back in use (the car is taken to a parking area; it can be reserved again).

#### **Exceptions**

- **The insurance asserts that the accident is fault of the User and doesn't take responsibility for it.** The User is fully charged of the reparation costs.

**Special Requirements** The new Operator cannot refuse the assignment if he is online and always accept it within 10 minutes. An Operator is always online when at work.

## 6 Alloy modeling



## 7 Appendix

### List of Figures

1	Mobile app: home page view . . . . .	14
2	Mobile app: main functionalities . . . . .	16
3	Website . . . . .	17

### List of Tables

#### 7.1 Used tools

For this assignment, we used the following tools:

**Alloy**

**LaTeX** The group used LaTeX to structure the final document and to help with versioning.

**Github** We leaned on Github for versioning and coordinating synchronized work.

**Toggl** We used toggl to keep track of work hours.

**Slack**

#### 7.2 Hours of work