

Luca Scannapieco -

Andrea Pasquali -

Emanuele Torelli - 876210

13/11/2016

PowerEnJoy

Requirement Analysis and Specification Document

Table of contents

[1. Introduction 3](#_Toc466716025)

[1.1. General description of the problem 3](#_Toc466716026)

[1.2. Stakeholders identifying 3](#_Toc466716027)

[1.3. Actors identifying 3](#_Toc466716028)

[1.4. Goals 3](#_Toc466716029)

[1.5. Text assumptions 4](#_Toc466716030)

[1.6. Domain assumptions 5](#_Toc466716031)

[1.7 Glossary 5](#_Toc466716032)

[1.7. Constraints 6](#_Toc466716033)

[1.7.1. Regulatory policies 6](#_Toc466716034)

[1.7.2. Interfaces 6](#_Toc466716035)

[1.7.3. Others 7](#_Toc466716036)

[2. Requirements 7](#_Toc466716037)

[2.1. Functional requirements 7](#_Toc466716038)

[2.1.1. Guests 7](#_Toc466716039)

[2.1.2. Users 7](#_Toc466716040)

[2.1.3. Assistance coordinators 9](#_Toc466716041)

[2.2. Non-functional requirements 10](#_Toc466716042)

[3. Functional Modeling 10](#_Toc466716043)

[3.1. Possible scenarios 10](#_Toc466716044)

[3.1.1. Scenario 1 10](#_Toc466716045)

[3.1.2. Scenario 2 10](#_Toc466716046)

[3.1.3. Scenario 3 11](#_Toc466716047)

[3.1.4. Scenario 4 11](#_Toc466716048)

[3.1.5. Scenario 5 11](#_Toc466716049)

[3.2. Use case diagram 11](#_Toc466716050)

[3.3. Use case description 12](#_Toc466716051)

[3.3.1. Sign up 12](#_Toc466716052)

[3.3.2. Login 13](#_Toc466716053)

[3.3.3. See available cars 13](#_Toc466716054)

[3.3.4. Report car accidents or damages 13](#_Toc466716055)

[3.3.5. Make a reservation 13](#_Toc466716056)

[3.3.6. Unlock the car 14](#_Toc466716057)

[3.3.7. See all the information about a ride 14](#_Toc466716058)

[3.3.8. Activate the money saving option 14](#_Toc466716059)

[3.3.9. Finish a ride 15](#_Toc466716060)

[3.3.10. See position and battery of cars 15](#_Toc466716061)

[3.3.11. Retrieve information about damaged cars 15](#_Toc466716062)

[4. Object Modeling 16](#_Toc466716063)

[4.1. Class diagram 16](#_Toc466716064)

[4.2. Statechart diagram 16](#_Toc466716065)

[5. Dynamic Modeling 17](#_Toc466716066)

[5.1. Sequence diagrams 17](#_Toc466716067)

[5.1.1. Login 17](#_Toc466716068)

[5.1.2. Reserve a car 18](#_Toc466716069)

[5.1.3. Unlock the car 19](#_Toc466716070)

[5.1.4. Activate the money saving option 20](#_Toc466716071)

[5.1.5. Report/collect information about damaged cars 21](#_Toc466716072)

[5.2. Activity diagrams 22](#_Toc466716073)

[5.2.1. Process of unlocking a car 22](#_Toc466716074)

[5.2.2. Process of finishing a ride 23](#_Toc466716075)

[6. Alloy modelling 23](#_Toc466716076)

[6.1. Model 23](#_Toc466716077)

[6.2. Alloy tool results 23](#_Toc466716078)

[6.3. World generated 23](#_Toc466716079)

[7. Other info 24](#_Toc466716080)

[7.1. Future development 24](#_Toc466716081)

[7.2. Reference documents 24](#_Toc466716082)

[7.3. Used tools 24](#_Toc466716083)

[7.4. Hours of work 24](#_Toc466716084)

# 

# 1. Introduction

## 1.1. General description of the problem

We will project PowerEnJoy, a car sharing service that allows people to reserve and drive electrical cars in Milano.

The system allows users to reserve a car via mobile app or via web app, using a GPS system to identify the position of the user and the position of the available cars near him/her.

Users can drive a car everywhere but they must park within safe areas accurately defined by the company.

Registration is mandatory before using the service, this allow to collect all the needed information about people who want to drive PowerEnJoy cars.

The system provides user some eventual discounts, for example if a user shares the car with at least two other people or if he/she charges the car at least at 80% of the power at the end of the ride.

The society has also hired some operators to deal with bad behaviors of the users, like cars left around the city with drain batteries.

## 1.2. Stakeholders identifying

Our Stakeholder is ElectricEngine Inc., a company that has been producing electric cars since 1999 and has decided to invest in car-sharing service in our city.

The company wants to provide a service completely eco-friendly using its model of electric car called “Volta”; its CEO is our Prof Luca Mottola.

However, we can adapt this system to accomplish other requests from other enterprises with the same type of cars.

## 1.3. Actors identifying

* Guests: people who haven’t registered to the service yet, they only can read a description of the service or sign up.
* Users: people who have already signed up so that the system has given them the password that can be used to access the system.
* Assistance coordinator: is a company employee in charge of manage the operators when a car needs assistance (e.g. battery replacement).

## 1.4. Goals

We want to build a system which is capable to reach the following goals:

1. Allow guests to sign up.
2. Allow users to sign in.
3. Allow users to see the available cars (and their battery level) near them or near to a given address.
4. Allow users to reserve an available car for up to one hour and to know if their reservation went successfully and eventually fine them if the hour expires.
5. Allow users to unlock and have access to a car if and only if they are close to that car and the car is reserved by them.
6. Allow users to end a ride if and only if the car is in a safe area or the car has run totally out of battery or an accident happens.
7. Allow users to receive a 10% discount from the total fee if they carry more than two people.
8. Apply a fine of 30% of the total cost to users if the car has been parked more than 3 km from the nearest power grid station or with less than 20% of battery.
9. Reward users with a 20% of discount if they leave the car with more than 50% of the battery.
10. Reward users with a 30% of discount if they leave the car charging into a power grid station.
11. Allow users to use the money saving option (see glossary)
12. Allow users to know in real time all the information (cost, car’s battery level, safe areas’ location) about their ride.
13. Allow users to report accidents or car damages.
14. Allow assistance coordinator to sign in.
15. Allow assistance coordinator to see the GPS position of all the available cars and their battery level in order to identify the cars in need of battery replacement.
16. Allow assistance coordinator to collect information from users about damaged cars.

## 1.5. Text assumptions

In this paragraph we are going to describe the assumptions we made according to the given specifications in order to build the clearest and most complete model possible.

* We though that is realistic to assume that a company which wants to invest in a car sharing service has to build the system starting from scratch.
* Safe areas are intended as a set of zones around the city instead of a boundary that surrounds the whole Milano.
* We assumed that safe areas are the only parking zones allowed by the company. So, unless involvements in accidents or battery drains, users can’t finish a ride if they haven’t park their PowerEnJoy car in a safe area.
* In case of users who break the road rules, they will be fined according to the law, so we don’t have to deal with this issue.
* Power stations are composed by a single plug where to attach the car, a set of power station is called power grid station.
* We assumed that the company has hired some operators which are meant to retrieve cars in need of assistance, for example cars out of battery (<20% of battery level) or cars involved in accidents.
* In order to speed up the process of retrieving cars with drain battery, we assumed that it’s better to send operators to replace batteries on site without taking the car to a power station.
* Operators are managed by an assistant coordinator, in addition only the assistance coordinator has access to the system.
* In our study we overlooked the actual payment phase, because often those kind of transactions are managed by banks using complex systems. We only deal with the process of calculating and showing to users the final fee of a ride, assuming that the real payment transaction will be applied automatically by the bank at the end of the ride.
* According to the assumption just mentioned, users can only use credit cards provided previously in the registration for payment.
* The company already knows how to deal with users in trouble with payments (e.g. users with not enough money on their credit card), so we do not have to deal with this issue.
* Users can’t cancel neither cancel their reservation nor reserve multiple car, in order to prevent bad behaviors that can reduce the service efficiency.
* We assumed that discounts that users can get during a ride are not cumulative, in order to prevent overdone reduction of the fee, so only the biggest one will be applied. In addition to that, if a user gets fined, the system won’t apply any discount.
* We though it’s reasonable to think that the discount related to the money saving option is the biggest discount a user can achieve, so we decided to fix it at 50% of total cost.

## 1.6. Domain assumptions

* Cars have a unique ID number.
* The maximum of other passengers excluding the driver during a ride is 4.
* If a sensor detects the presence of a person in the car, it means that the person is actually inside the car (sensors can’t be cheated for examples using heavy objects).
* Only the owner of a reservation will drive the car he has reserved.
* Once a user unlocks his reserved car, he will actually get in and start a ride (unless he detects damages to the car).
* All the GPSs always give the right position of the cars and must be always working.
* The company already handles the information about the operators, so we don’t have to deal with them.
* All the power grid stations are in a safe area.
* At the end of a ride, users will go away and eventually plug the car into a station in order to get a discount within 3 minutes.
* Whenever a car is left with the battery level less than 20% an operator will go to replace that battery with a fully charged one within 2 hours.
* Only operators can replace the battery of a car.
* Users never reserve a car when its battery level is at 0%.
* Only cars belonging to the company can be parked at a power station.

## 1.7 Glossary

* Guest: person who hasn’t registered yet to the service. He can only sign up to take benefits from the service
* Sign up: process that allows a guest to provide his password so that he can access to the system. In the registration process he must compile a form giving these information:
  + Name
  + Surname
  + Phone number
  + Email
  + Address
  + Birth date
  + Birth place
  + SSN
  + Zip code
  + Credit card number
  + Driving licence’s number

the system will reply sending him an email containing a password.

* User: person who has already registered and can access to the system using his username and password to reserve or unlock an available car.
* Reservation: a process thanks to which a user can reserve an available car up to one hour: since he reserves it, he has exactly one hour to reach it and unlock until the system deletes his reservation and fines him of 1 Euro.
* Ride: by calling ride we mean the time interval from the moment the user starts PowerEnJoy car to the moment he presses the button to terminate the route (after he has turned off the car as well).
* End of ride: state of the car that starts from the moment the user presses the related button after parking in a safe area and goes until the user presses the related button on the mobile app (or automatically after a maximum of 3 minutes). The screen of the car says goodbye to the user and turns off, and he can see a recap screen in his mobile app in which is showed the fee and the discount achieved (or eventually the fine).
* Available car: it’s a PowerEnJoy car not reserved by another user and no other user is driving it.
* Out of order car: an out of order car can’t be reserved because of damages, in fact it is not available until it will be fixed and the assistance coordinator tags it as available again
* Safe area: a car is parked in a safe area if it is in one of the parks belonging to the set pre-defined by the management system.
* Sharing discount: it’s a 10% discount from the total price given to users who carry at least other two people.
* Battery discount: it’s a 20% discount from the total price given to users who leave the car parked with more than 80% of battery level.
* Power station discount: it’s a 30% discount from the total price given to users who leave the car parked charging in a power station.
* Low battery fine: it’s a 30% fine from the total price given to users who leave the car parked with less than 20% of battery level.
* Money saving option: if the user enables the money saving option, he can input his final destination and the system provides information about the station where to leave the car to get a discount.
* Money saving option discount: it’s a 50% discount from the total price given to users who follow the money saving option instructions.
* Money saving station: it’s the station provided by the money saving option.
* Power station: it’s a place in which a car can be recharged.

## 1.7. Constraints

### 1.7.1. Regulatory policies

The system must ask the user the permission to get his position and the permission to manage sensible data (position, mail) according to the privacy law. Furthermore, the systems must not use notiﬁcations to send SPAM respecting the privacy law.

### 1.7.2. Interfaces

* Mobile application
  + 3G/4G/Wi-Fi connection
  + GPS
  + 64 MB space for app package
  + 64 MB of RAM
* Web Browser
  + Modern browser with AJAX
  + Access to GPS of device
* Screen inside car
  + GPS navigator
  + Car battery level indicator
  + Power stations position
  + Real time cost indicator
  + Safe areas position

### 1.7.3. Others

The server supports parallel operations from diﬀerent clients.

# 2. Requirements

## 2.1. Functional requirements

Assuming that the domain properties stipulated in the paragraph 1.6. hold, and, in order to fulﬁll the goals listed in the paragraph 1.4., the following requirements can be derived. The requirements are grouped under each goal from which it is derived.

### 2.1.1. Guests

1. Allow guests to sign up:

* The system must be able to save the information of all the people who sign up.
* the system must be able to check the correctness of the registration info provided by the user.
* The system replies to every correct registration with a password that the user must use to access.
* The system replies to every incorrect registration by notifying the error.
* The system must prevent users to sign up more than once.

### 2.1.2. Users

1. Allow users to sign in:

* The system must be able to check if the credentials are correct.
* The system must allow the user to sign in if and only if the provided credentials are correct

1. Allow users to see the available cars (and their battery level) near them or near to a given address:

* The system must have access to the GPS position of all the available cars.
* The system must have access to the GPS position of the user.
* The system must be able to detect all the available cars within a certain distance from the user and show them on a map.
* The system must be able to detect all the available cars within a certain distance from a position selected by the user and show them on a map.
* The system must show the battery level of each available car displayed to the user.

1. Allow users to reserve an available car for up to one hour and to know if their reservation went successfully and eventually fine them if the hour expires

* The system must tag the car as not available as soon as the reservation is performed.
* The system must notify the user when a reservation goes successfully.
* The system activates the reservation timer for the reserved car as soon as the reservation is performed.
* The system must tag the car as available as soon as the reservation timer is expired.
* The system must notify the user when the reservation timer expires.
* The system must send a 1 Euro fine to users who haven’t taken the cars they reserved within one hour.
* The system must show the position of the reserved car only to the user who made the reservation as long as the reservation timer is running.

1. Allow users to unlock and have access to a car if and only if they are close to that car and the car is reserved by them.

* The system must detect if the user is less than 5 meters distant from the reserved car.
* The system must be able to unlock the car once the user is less than 5 meters distant from it.

1. Allow users to end a ride if and only if the car is in a safe area or the car has run totally out of battery or an accident happens:

* The system must have access to the GPS position of the car.
* The system must authorize the user to end the ride if and only if the car is switched off and its GPS position is in a safe area or an incident occurs or the car runs totally out of battery.
* The system must tag the car as available if and only if the ride is ended in a safe area and the user exited the car.
* The system must stop charging the user once he ends the ride.
* The system must alert the user if he attempts to end a ride but he is not in a safe area or the car is not switched off.
* The system must display the final total cost of the ride once the ride is ended.

1. Allow users to receive a 10% discount from the total fee if they carry more than two people:

* The system must detect the number of passengers that stay in the car for at least half of the duration of the ride.
* The system must apply a 10% discount on the total fee if that number is at least two and if the user hasn’t got neither a greater discount nor a fine.

1. Apply a fine of 30% of the total cost to users if the car has been parked more than 3 Km from the nearest power grid station or with less than 20% of battery.

* The system must detect the battery level of a car.
* The system must have access to the GPS position of the car.
* When the ride ends, before calculating the total fee, the system must detect whether the car is plugged into a power grid station or not.
* The system must apply a fine of 30% of total cost to a user if it detects that the user has ended the ride leaving the car with less than 20% of battery and not plugged into a power grid station.
* The system must apply a fine of 30% of total cost to a user if it detects that the user has ended the ride leaving the car in safe area that is more than 3 Km away from a power grid station.

1. Reward users with a 20% of discount if they leave the car with more than 50% of the battery

* The system must detect the battery level of a car.
* When the ride ends, before calculating the total fee, the system must detect whether the car is plugged into a power grid station or not.
* The system must apply a 20% of discount on the total fee if it detects that the user has ended the ride leaving the car with more than 50% of battery and he hasn’t got neither a greater discount nor a fine.

1. Reward users with a 30% of discount if they leave the car charging into a power grid station.

* When the ride ends, before calculating the total fee, the system must detect whether the car is plugged into a power grid station or not.
* The system must apply a 30% of discount on the total fee if it detects that the user has ended the ride leaving the car plugged into a power grid station and he hasn’t got neither a greater discount nor a fine.

1. Allow users to use the money saving option (see glossary).

* The system must ask to the user for money saving option activation once the user unlocks the car.
* Once the user activates the money saving option, the system must ask to the user his destination.
* The system must be able to calculate the availability of power plugs in all the power grid stations.
* The system must be able to calculate the distribution of cars in the city.
* The system must be able to determine the money saving station (see glossary).
* The system must show to the user the selected money saving station on a map.
* When the ride ends, before calculating the total fee, the system must detect whether the car is plugged into the selected power money station or not.
* The system must apply a 50% of discount on the total fee if it detects that the user has ended the ride leaving the car plugged into the selected power money station.

1. Allow users to know in real time all the information (cost, car’s battery level, safe areas and power grid station’s location) about their ride.

* The system must be able to calculate how much money the user is spending during a ride.
* The system must be able to display in real time the current fee of the ride.
* The system must be able to detect the battery level of a car during the ride.
* The system must be able to display in real time the current battery level of the car during the ride.
* The system must be able to display the location of all the predefined safe areas during the ride.

1. Allow users to report accidents or car damages.

* The system must allow users to report accidents during their ride.
* The system must allow users to report car damages provoked by previous users.
* The system must tag the car reported as damaged from the users as out of order.

### 2.1.3. Assistance coordinators

1. Allow assistance coordinator to sign in.

* The system must be able to check if the assistance coordinator credentials are correct.
* The system must allow the assistance coordinator to sign in if and only if the provided credentials are correct.

1. Allow the assistance coordinator to see the GPS position of all the available cars and their battery level in order to identify the cars in need of battery replacement.

* The system must have access to the GPS position of all the available cars.
* The system must detect the battery level of all the available cars.
* The system must show to the assistance coordinator all the available cars on a map, highlighting the ones with less than 20% of battery level.
* The system must be able to notify the assistance coordinator that a battery replacement went successfully.

1. Allow assistance coordinator to gather information from users about damages cars in order remove them

* The system must store accidents and car damages reports of the users.
* The system must display the reports to the assistance coordinator.
* The system must allow the assistance coordinator to archive a report.
* The system must allow the assistance coordinator to tag cars as available when they are repaired.

## 2.2. Non-functional requirements

# 3. Functional Modeling

## 3.1. Possible scenarios

Here are described some possible scenarios of usage PowerEnJoy system.

### 3.1.1. Scenario 1

Mario should go to work, but someone has parked in front of his garage, so he is unable to use his personal car. Fortunately, Mario is registered to PowerEnJoy, so he picks his smartphone and opens the PowerEnJoy app, then he inserts his credentials to log in the system. After that, he takes a look at the map to see if there is any available car near him. He notices that there is an available car parked 2 minutes walking away from him, so he immediately reserves it to prevent other users to take it away before him. Once he is close to the car, he looks the ID number of the car highlighted in the windshield, then he opens the app again and inserts the code. Since the system recognizes that Mario actually is the user who has made the reservation and he inserted the right code, the car unlocks the door so that Mario can get into it, ignite the engine with the keys provided inside the dashboard, and go to work on time.

### 3.1.2. Scenario 2

Mario is driving a PowerEnJoy car. Once he arrives to his home, he looks at the monitor to see if he is in a safe area, but he figures out that his car has the battery very low (10%). Since the football match on the TV is starting, he doesn’t want to look for a power grid station to recharge the car, so he leaves it as it is, out of battery. Once Mario gets out of the car, the system detects that Mario’s ride is over, but since the car has been left with less than 20% of battery charged, in addition to the cost of the ride the system will withdraw an additional amount of money as a fine from Mario’s credit card.

### 3.1.3. Scenario 3

Mario has an appointment to the cinema with his friends Ugo and Anna, but today there is a transport strike and the cinema is quite far from their houses. So, Mario, who is a PowerEnJoy user, decides to go taking his friends up to share the route. Car’s sensors detect that in the car there are more than two passengers in addition to the driver, so at the end of the ride the system will apply a discount to the total fee. Once Mario and friends arrive to the cinema, they decide to leave the car in the nearest power grid station, in order to get a bigger discount. Once the car is attached to the power charger and everyone is out, the system detects that the ride is over and calculates the total amount of money that will withdraw from Mario’s credit card, considering also the biggest of the discounts from the ones which he achieved. Since Mario got both the 10% discount for carrying at least 2 people and the 20% discount as he left the car in charging in a power grid station, the system will eventually apply a 20% discount from the total fee of the ride.

### 3.1.4. Scenario 4

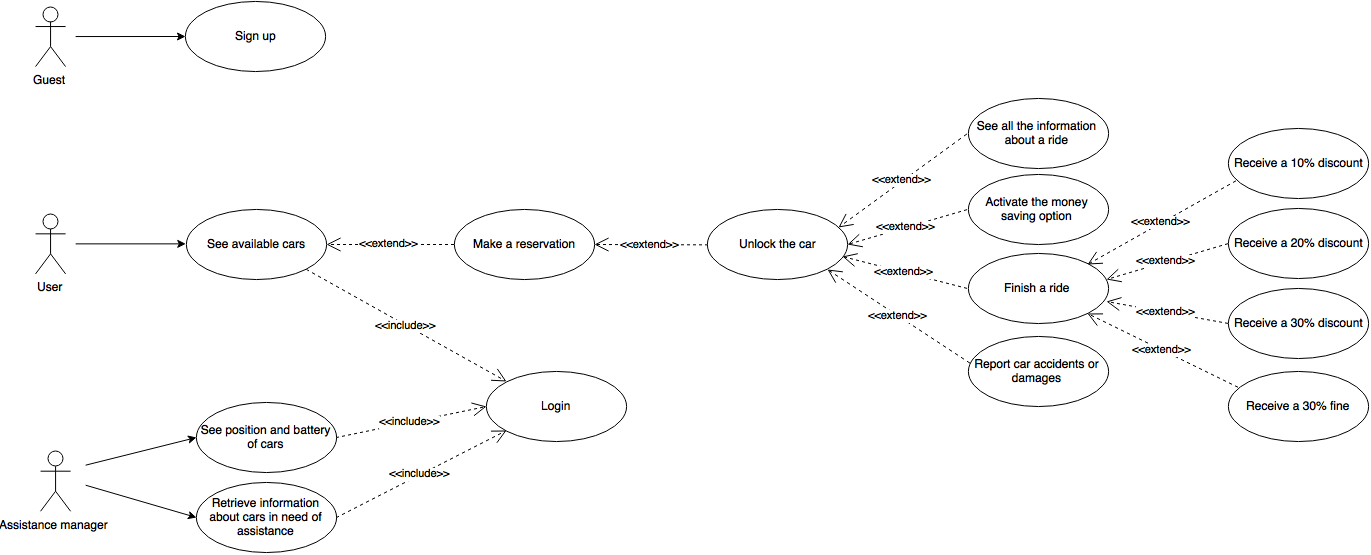
Mario is an operator at the Electric Engine Inc. and one of his main tasks is to monitor the cars in order to prevent the company to have cars out of battery left around the city. The procedure is simple: all that Mario has to do is login to the system via web or mobile app, and select the option which allows him to see all the available cars on a map. In order to facilitate Mario’s search, the system highlights the cars whose battery level is under 20%. In fact, cars with less than 20% of battery are supposed to be almost unusable, so Mario contacts an operator and tells him to reach the car to do a battery replacement. The operator is therefore meant to go immediately at the location of the car and replace its battery with a fully charged one.

### 3.1.5. Scenario 5

Anna would like to drive a PowerEnJoy car, so she makes a reservation and reaches the selected car. Unfortunately, when she arrives to the car, she realizes that the car door beside the driver seat is smashed and even if Anna has run successfully the procedure of unlocking, the door doesn’t open. Therefore, Anna opens her PowerEnJoy mobile app and selects the option for reporting car damages or accidents. Anna fills in the form with a detailed description of the issue and points out that the damage isn’t her fault as the car was already damaged at the arrival. At this point, the car reported as damaged is tagged by the system as out of order and nobody will be able anymore to reserve it. Meanwhile in the Electric Cars Inc. office, an assistance manager receives a notification from the app concerning a report by a user. So, the assistance manager reads the report left by Anna, and orders an operator to retrieve it in order to repair the problems. When the car is repaired, the assistance coordinator tags it at available, then archives the report previously provided by Anna.

## 3.2. Use case diagram

We derive the use case diagram according to the goals listed in paragraph 1.4. and the scenarios mentioned above:



## 3.3. Use case description

We describe in a detailed way the use cases that we derived from the scenarios. We try to define them all. It is important to understand that all the references to “pages”, “buttons” or “input forms” are only hypothesis to make the situation as clear as possible ad to help the reader to draw a visual picture in his mind of what we plan to do, but the real structures will be well defined in the Design Document.

### 3.3.1. Sign up

|  |  |
| --- | --- |
| Name | Sign up |
| Actors | Guests |
| Entry conditions | The guest isn’t registered to the service |
| Flow of events | * The guest accesses to the service via web app or mobile app * The guest clicks on the “sign up” button * The guest fills in the form where he has to write:   + Name   + Surname   + Phone number   + Email   + Address   + SSN   + Credit card number   + Driving licence’s number * The guest clicks the “done” button * System checks the correctness of the information provided * System reply to the user with his new password |
| Exit conditions | Registration successfully done, the guest becomes a registered user |
| Exceptions | An exception can be caused if the email address of the guest already exists or if some fields of the form aren’t filled properly, in this case the user is asked by the system to try again. |

### 3.3.2. Login

|  |  |
| --- | --- |
| Name | Login |
| Actors | Users, assistance manager |
| Entry conditions | The user is already registered to the service, and of course assistance managers have got special credentials to access the system |
| Flow of events | * User/assistance manager accesses the system via web app or via mobile app * User/assistance manager fills in the text fields in the home page with email and password * User/assistance manager clicks on the “login” button. |
| Exit conditions | Login successfully done |
| Exceptions | An exception can be caused if the email or the password aren’t correct, then an error message is displayed and the credentials are requested again |

### 3.3.3. See available cars

|  |  |
| --- | --- |
| Name | See available cars |
| Actors | Users |
| Entry conditions | The user is already logged in to the service |
| Flow of events | * Users open the map to see all the available cars near them or they select a destination to see all the available cars near that address |
| Exit conditions | The user can see all the available cars displayed in a map and their battery level |
| Exceptions | No exceptions |

### 3.3.4. Report car accidents or damages

|  |  |
| --- | --- |
| Name | Report car accidents or damages |
| Actors | User |
| Entry conditions | User is logged in the system |
| Flow of events | * User selects the form to report a damage to a car * User fill in the form the trouble he has encountered, specifying if the damage of the car is caused by his/her fault or by a previous user * Users confirm the request by pressing a button |
| Exit conditions | A message is displayed to the user to notify that the report went well, the car is no longer available because tagged as out of order by the system |
| Exceptions | No exceptions |

### 3.3.5. Make a reservation

|  |  |
| --- | --- |
| Name | Make a reservation |
| Actors | Users |
| Entry conditions | User is viewing the available cars in a map |
| Flow of events | * User selects the car which they want to reserve * User presses on the “reserve” button * A message is displayed to the user to notify that the reservation went successfully |
| Exit conditions | User can see the information and the position of the car they have reserved |
| Exceptions | No exceptions |

### 3.3.6. Unlock the car

|  |  |
| --- | --- |
| Name | Unlock the car |
| Actors | Users |
| Entry conditions | User is less than 5 m distant from the car they have reserved |
| Flow of events | * User selects in the application the option of unlocking the car * The system unlocks the car |
| Exit conditions | The screen in the car welcomes the user and the system gets ready to start a ride (the ride will start once the user ignites the engine) |
| Exceptions | User is more than 5 m away from the car, in this case the system shows an error message and doesn’t unlock the car |

### 3.3.7. See all the information about a ride

|  |  |
| --- | --- |
| Name | See all the information about a ride |
| Actors | Users |
| Entry conditions | The user has unlocked the car |
| Flow of events | * The user starts the car |
| Exit conditions | The screen inside the car starts displaying the details of the ride to the user (cost, battery, safe areas, power station) |
| Exceptions | nothing |

### 3.3.8. Activate the money saving option

|  |  |
| --- | --- |
| Name | Activate the money saving option |
| Actors | Users |
| Entry conditions | The user has unlocked the car |
| Flow of events | * The user selects in the car’s screen the money saving option by pressing the related button * The user inserts the address of his destination * The user presses a button in order to confirm his choice |
| Exit conditions | The screen is highlighting the path leading to the “money saving station” according to the destination inserted by the user |
| Exceptions | nothing |

### 3.3.9. Finish a ride

|  |  |
| --- | --- |
| Name | Finish a ride |
| Actors | Users |
| Entry conditions | The user is performing a ride |
| Flow of events | * The user turns off the engine of the car * The user can see a recap of their ride on the screen * If a discount is achieved/user gets fined, it will be displayed on the screen |
| Exit conditions | The car is tagged as available again from the system after 5 min |
| Exceptions | The user has not parked in a safe area, then the car will make an acoustic signal and display a message on the screen and it will not be possible to finish the ride. |

### 3.3.10. See position and battery of cars

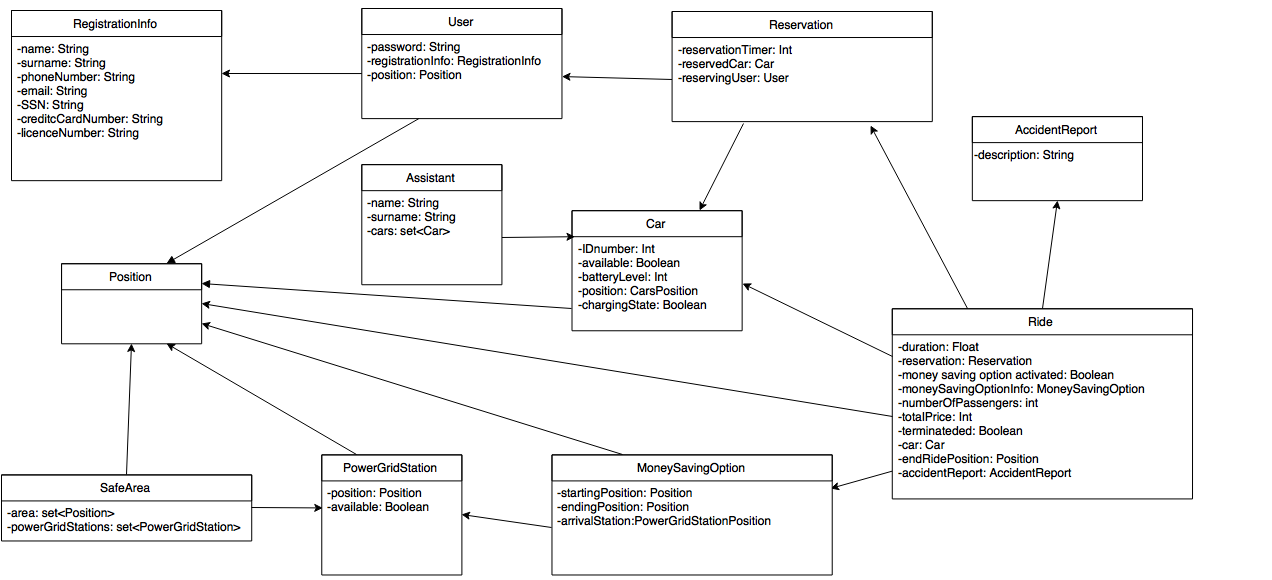
|  |  |
| --- | --- |
| Name | See position and battery of cars |
| Actors | Assistance coordinator |
| Entry conditions | Assistance coordinator is already logged into the system |
| Flow of events | * The assistance coordinator presses a button from the web app or the mobile app |
| Exit conditions | The assistance manager can see all the available cars displayed on a map and their battery level, cars with less than 20% of battery are highlighted |
| Exceptions | nothing |

### 3.3.11. Retrieve information about damaged cars

|  |  |
| --- | --- |
| Name | Retrieve information about damaged cars |
| Actors | Assistance coordinator |
| Entry conditions | Assistance coordinator is already logged into the system |
| Flow of events | * The assistance manager presses a button from the web app or the mobile app * The assistance manager selects one of the reports provided by a user by clicking on it * The assistance manager archives the report clicking the related button * The system tags the car of the report as available again |
| Exit conditions | The report is no longer stored into the system and the car is available again |
| Exceptions | nothing |

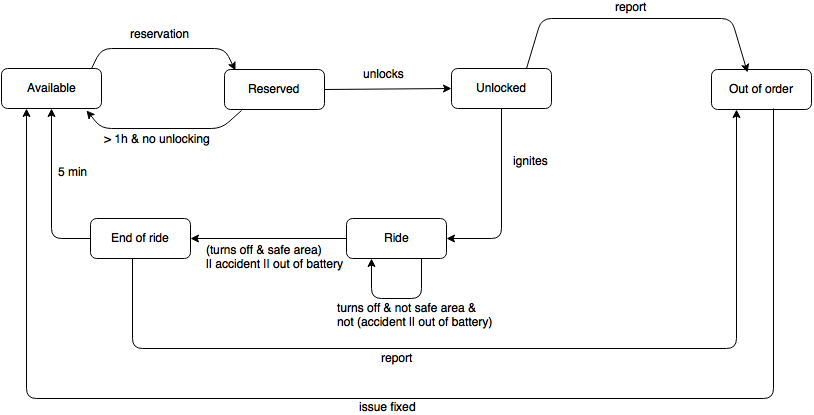
# 4. Object Modeling

## 4.1. Class diagram



## 4.2. Statechart diagram

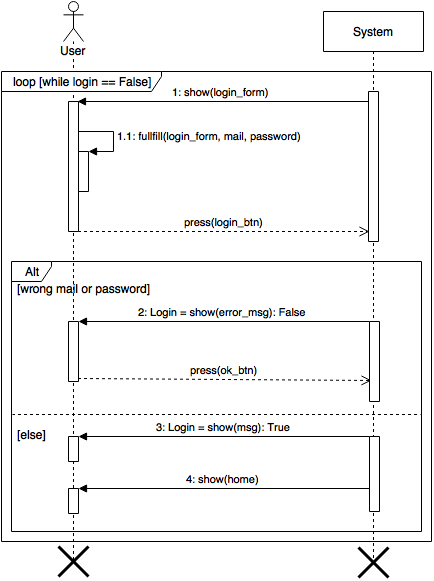
Here is a representation of all the states in which PowerEnJoy cars are supposed to be and the conditions of moving from one state to another.



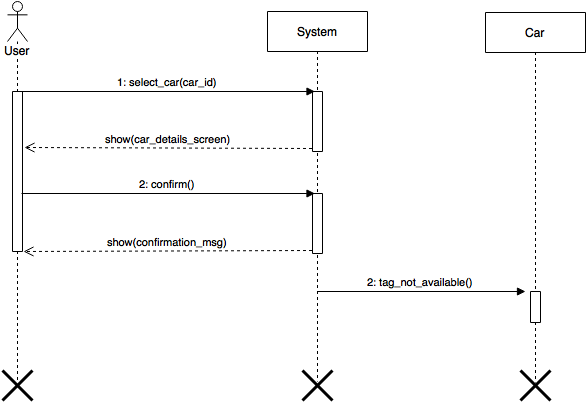
# 5. Dynamic Modeling

## 5.1. Sequence diagrams

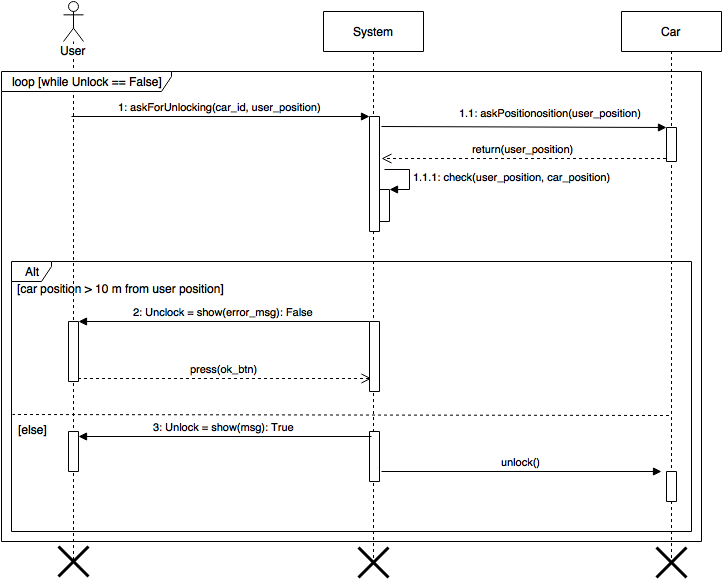
### 5.1.1. Login



### 5.1.2. Reserve a car

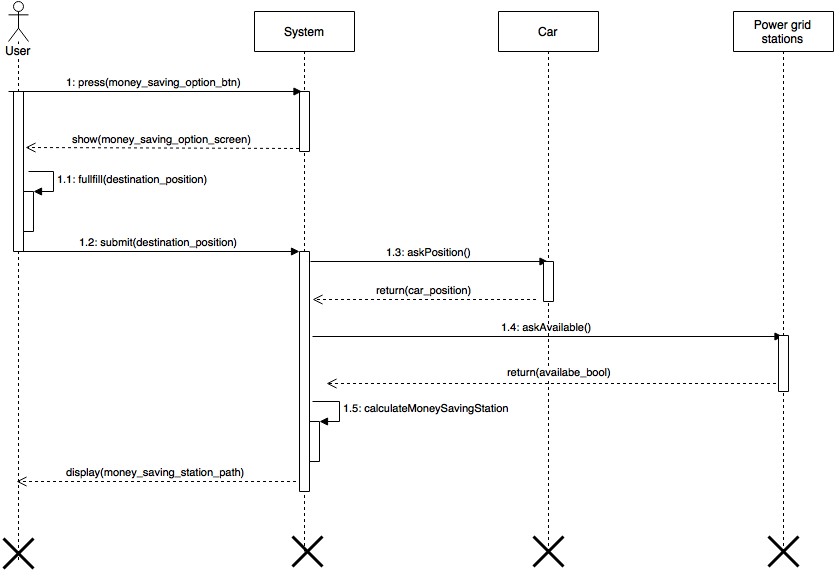


### 5.1.3. Unlock the car

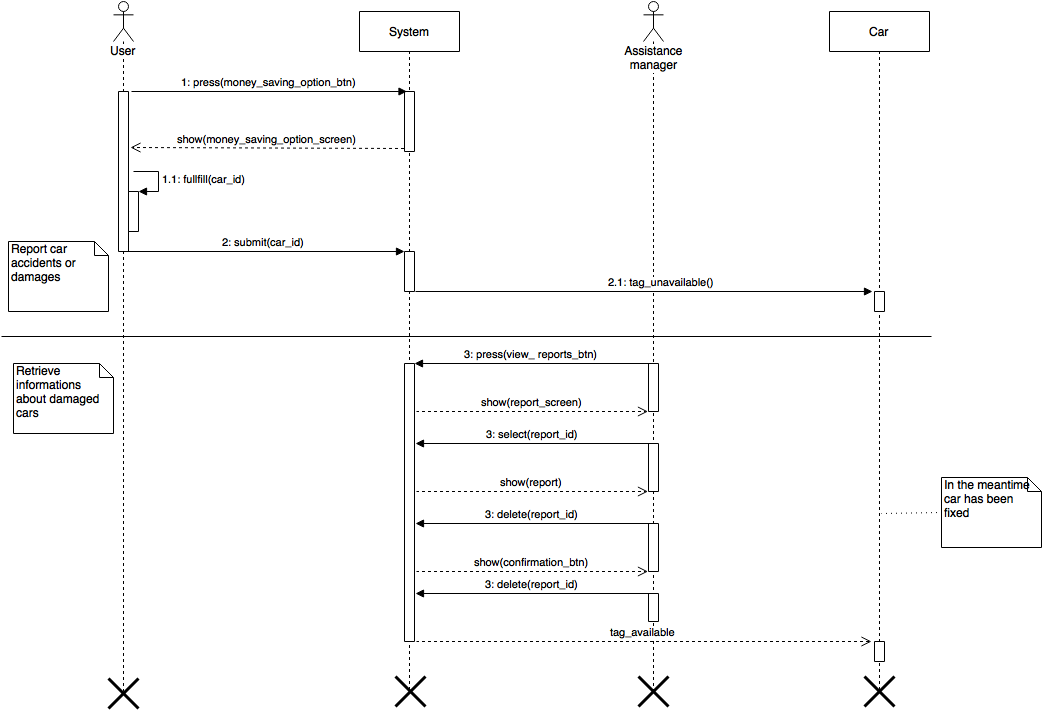


### 

### 5.1.4. Activate the money saving option

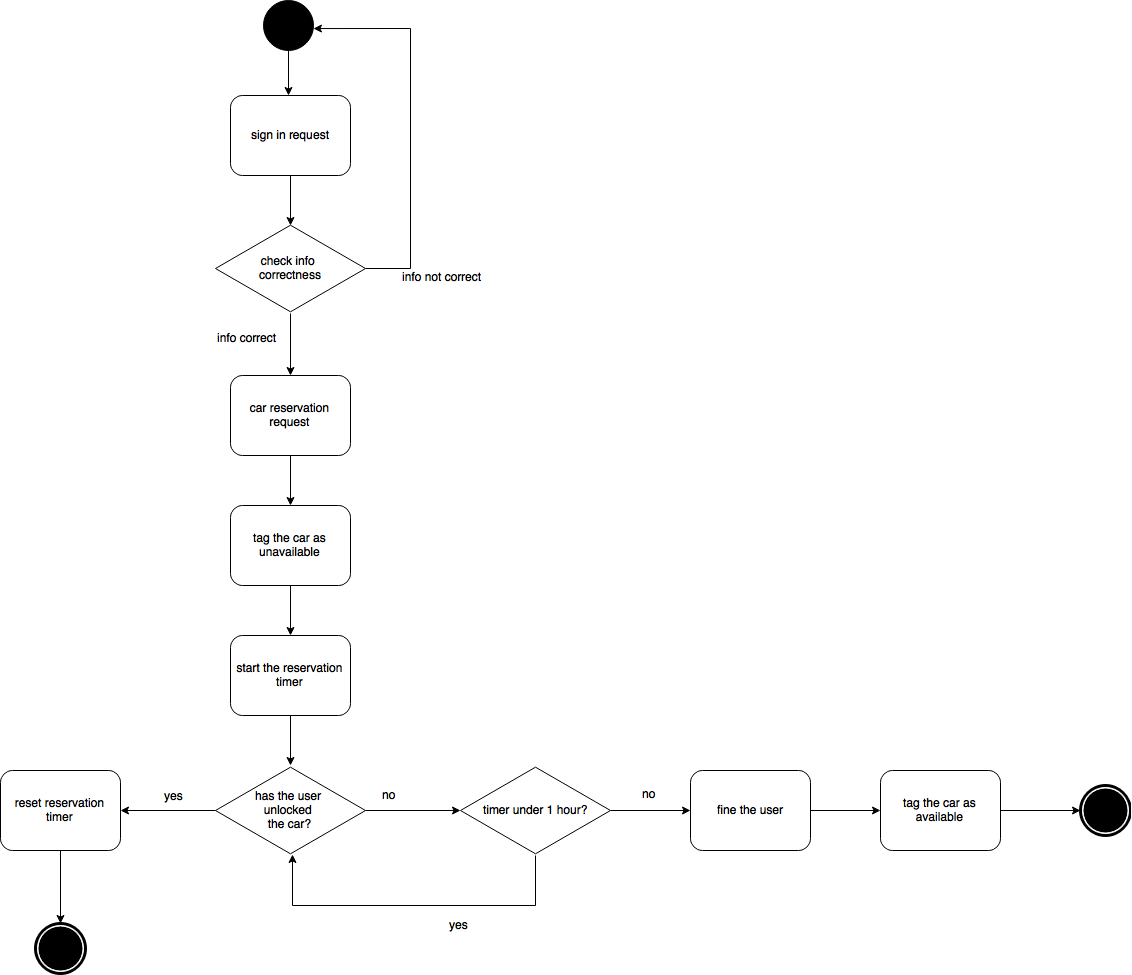


### 5.1.5. Report/collect information about damaged cars

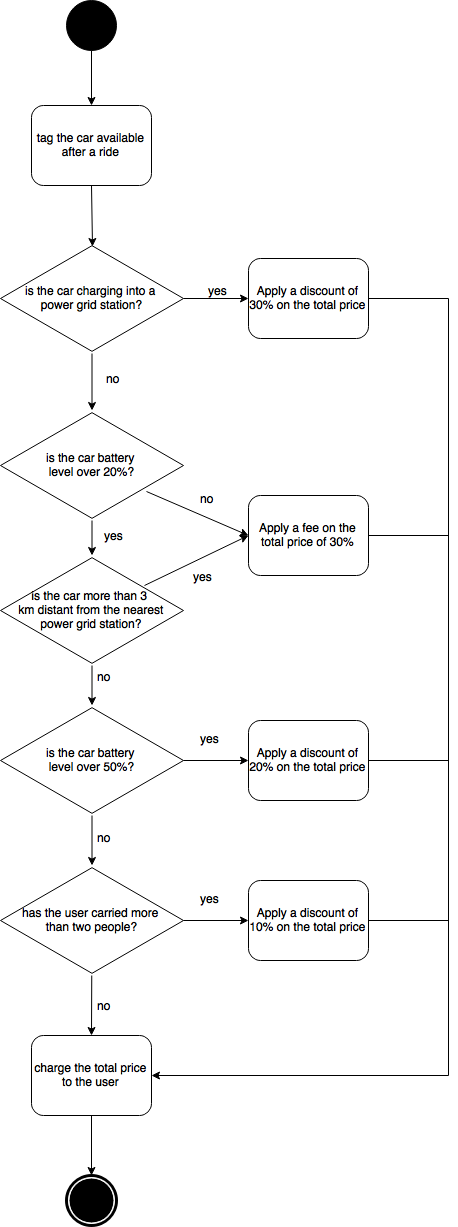


## 5.2. Activity diagrams

### 5.2.1. Process of unlocking a car



### 5.2.2. Process of finishing a ride



# 

# 6. Alloy modelling

## 6.1. Model

## 6.2. Alloy tool results

## 6.3. World generated

# 7. Other info

## 7.1. Future development

## 7.2. Reference documents

* Assignments AA 2016-2017.pdf
* IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Speciﬁcations
* Sample documents:
  + RASD sample from Oct. 20 lecture.pdf
  + RASD\_meteocal-example1.pdf
  + RASD\_meteocal-example2.pdf
  + RASD\_meteocal-example3.pdf
* Course slides:
  + Requirement engineering Part I.pdf
  + Requirement engineering Part II.pdf
  + Requirement engineering Part III.pdf

## 7.3. Used tools

* Microsoft Word 2016, for the drafting of RASD
* Microsoft OneDrive, to allow concurrent editing
* GitHub, to store the project in a repo
* Draw.io, for the drawing of the diagrams
* Alloy Analyzer 4.2, to prove the consistency of our model
* Balsamiq, for the mockups

## 7.4. Hours of work

For redacting and writing the Requirements analysis and specification document we spent approximately 30 hours per person.