POLITECNICO
MILANO 1863

Requirements Analysis and Specifications Document

Guido Muscioni (mat. 876151)
Marco Orbelli (mat. 876649)
Paola Marchesini (mat. 876541)

Contents

1 Introduction 3
- 1.1 Purpose . . . 3
- 1.2 Document Conventions . . . 3
- 1.3 Intended Audience And Reading Suggestions . . . 5
- 1.4 Product Scope . . . 5
- 1.5 References . . . 5

2 Overall Description 5
- 2.1 Product Perspective . . . 5
  - 2.1.1 User interface . . . 6
  - 2.1.2 Website . . . 6
  - 2.1.3 Mobile application . . . 6
  - 2.1.4 Screen of the car . . . 6
- 2.2 Product functions . . . 6
- 2.3 User characterics . . . 8
- 2.4 Constraints . . . 11
  - 2.4.1 Regulatory policies . . . 11
  - 2.4.2 Hardware requirements . . . 11
  - 2.4.3 Interfacing with other systems . . . 12
  - 2.4.4 Parallel operation . . . 12
- 2.5 Assumptions and dependencies . . . 12

3 Specific requirements 14
- 3.1 Use Case diagrams . . . 14
  - 3.1.1 User . . . 14
  - 3.1.2 Employee . . . 23
- 3.2 Sequence diagrams . . . 25
  - 3.2.1 Activity diagrams . . . 27
  - 3.2.2 State chart diagrams . . . 29
- 3.3 Activity diagrams . . . 30
- 3.4 State chart diagrams . . . 30
- 3.5 User interface . . . 30
  - 3.5.1 Website . . . 30
  - 3.5.2 Mobile application . . . 30
  - 3.5.3 Screen of the car . . . 30

4 Alloy 30

5 Appendices 35

1 Introduction

1.1 Purpose

The aim of the project is to develop a digital system for an electric car-sharing company, that is called PowerEnjoy. There are not previous system so that document will describe the requirements that deal with all the system that the customer wants.

The users of PowerEnjoy system are:

- Client;
- Employee.

Both of these users are going to have unique credentials in order to use the system, so they must be registered.
The clients will use this system to reserve and use cars of PowerEnjoy company.
The employees will use that system in order to know what cars need help.

1.2 Document Conventions

- Headquarter: it is the head office of Power Enjoy Society. There are both call centre and employees who deal with taking care of cars.
- Emergency number: number labeled in the car that the user can call in case of any emergency.
- Call center: a large office in the Headquarter in which a Power Enjoy's employees provide information or service to its customers by phone when they call the emergency number.
- Car: it is an electric car registered in the Power Enjoy Company.
- User: he/she is a client of the service.
- All Credentials: these are all information about the user that the system needs. The information contains name, last name, social security number, home address, email and drive license number.
- Login Credentials : these are email and password that the user has to provide at the login phase.
- Check message: is the message that the system sends to the user if he/she fills out the form, during the registration phase, with information that are not correct.
- Reservation message: is the message that the system send to the user who has just reserved a car.

- State: is the state of the car which includes the level of battery, position and the number of passengers. If the vehicle is reserved or used the state will be associated with the respective user.

- Special Parking Areas : these are special areas with plugs for the car.

- Safe Areas: these are the area that determinate the boundary domain of our service.

- Unsafe Areas: all the areas that are not safe / this term includes everything that is not in the safe areas.

- Unsafe fine: the fine that the user has to pay if he/she left the car in an unsafe area.

- Time Expire: the period of time from reservation that the user has to get to the car without paying the expire time fine.

- Expire time fine: fine that the user has to pay if, after an hour from the reservation, he/she doesn't reach the car.

- Money saving option: if the user enables the money saving option, he/she can input his/her final destination and the system provides information about the station where to leave the car to get a discount. This station is determined to ensure a uniform distribution of cars in the city and depends both on the destination of the user and on the availability of power plugs at the selected station.

- Final destination: the destination that the user enters in the navigator once he/she has hired a car.

- Symbolic amount: money that the user has to pay once he/she registered, the value is of one cent .

- Money saving option: if the user enables the money saving option, he/she can input his/her final destination and the system provides information about the station where to leave the car to get a discount. This station is determined to ensure a uniform distribution of cars in the city and depends both on the destination of the user and on the availability of power plugs at the selected station.

- Types of discount:

  - If the system detects the user took at least two other passengers onto the car, the system applies a discount of 10% on the last ride.
  - If a car is left with no more than 50% of the battery empty, the system applies a discount of 20% on the last ride.
  - If a car is left at special parking areas where they can be recharged the system applies a discount of 30% on the last ride.

1.3 Intended Audience And Reading Suggestions

A chi verrà rivolto il documento

1.4 Product Scope

The aim of the softwar is to deveolp a system which will satisfy these goals:

Employee

- [G1] Allows the employee to log in in the system, by app or website.
- [G2] Allow the employee to put the car in unavailable mode.
- [G3] Allow the employee to know what car he/she has to take care.

User

- [G1] Allows user to create an account.
- [G2] Allows user to log in in the system, by application or website.
- [G3] Allow user to modify his/her credentials.
- [G4] Allows user to look for cars nearby.
- [G5] Allows user to reserve a car.
- [G6] Allow user to send a request for opening the car.
- [G7] Allow user to enable the money saving option.
- [G8] Allow user to finish his/her ride and visualize the information about it.
- [G9] Allow user to view the special parking area in the map.
- [G10] Allows user to pay for the ride.

1.5 References

Documenti di riferimento

2 Overall Description

2.1 Product Perspective

The software is a stand-alone system that does not extend or takes part in any nowdays software. The access to the sofware can be performed by the Website and the mobile application.

2.1.1 User interface

2.1.2 Website

2.1.3 Mobile application

2.1.4 Screen of the car

2.2 Product functions

Functional requirements

Employee:

- [G1] Allows the employee to log in in the system, by app or website:
  - The system must be able to check if the credentials provided are correct.
- [G2] Allow the employee to put the car in unavailable mode.
  - The employee must be able to set the state of the car in unavailable mode, in order to take car of the car.
  - The employee must be able to release the car, to make it available by all users.
- [G3] Allow the employee to know what car he/she has to take care.
  - In the headquarters, the employee will receive a list of car which he/she has to take care and the daily colleague.
  - That list is continuously being update according to the daily activity delivered by the system.

User:

- [G1] Allows user to create an account.
  - The user inserts his credentials into the specific section on the website or the application.
  - The system must verify the existence of the email account and then sends the password to the new user.
- [G2] Allows user to log in in the system, by application or website.
  - The system must be able to check if the credentials provided are correct.
- [G3] Allow user to modify his/her credentials.
  - The user must be able to modify the photo of his/her driving license.
  - The user must be able to modify his/her phone number.

  - The user must be able his/her mail address.
  - The user must be able to modify his/her payment information.
  - The user must be able to modify his/her password.

- [G4] Allows user to look for cars nearby.
  - The system must be able to know the position of the user who want to reserve a car.
  - The system must be able to send the positions of the cars parked nearby.
  - The user must be able to zoom in and zoom out the area which is visualized by the application to look for free cars.

- [G5] Allows user to reserve a car.
  - The user must be able to verify the state of the chosen car.
  - The user must be able to visualize the way to reach the car.
  - The user must be able to verify if the reservation time expires.
  - The system must send a message to the phone number of the user. This message explaines how to open the car in case of troubles with the GPS or with the internet connection.
  - The user must be able to delete the reservation if he does not need the car anymore.
  - The system must be able to fine (with the "Time expire fine") the user if he/she does not reach the car in time.

- [G6] Allow user to send a request for opening the car.
  - The user can open the car with a button in the application.
  - The user can open the car answeriing to the message received when he reserved the car.
  - The system must verifies if the user is near the car.
  - If the conditions are verified, the system send to the car the opening command.

- [G7] Allow user to enable the money saving option.
  - The user can activate this mode with a button on the screen.
  - The user can input the final destination.
  - The user can visualize the special parking area near his/her destination.
  - The user can select a special parking area.
  - The user can see the way for the selected location.

- [G8] Allow user to finish his/her ride and visualize the information about it.
  - If the car has been parked in a safe area, the user can select the option of "Terminate ride".
  - The user must be able to see the final information about the ride, but not the possible discounts.
- [G9] Allow user to view the special parking area in the map.
  - The user must be able to see the map of the special parking area during the ride.
  - The user must be able to select a specific parking area, and see the way to it.
- [G10] Allows user to pay for the ride.
  - The system has to be able to send, to the payment system, the payment credentials that the user provides during the registration phase.
  - The user has to be able to receive the confirmation of the payment, with the possible discounts.

Non-functional requirements vari requisiti non funzionali, mockup ecc...

- When the user finishes his/her ride, he/she has 1 minute to plug the car.
- The system closes the car 20 seconds after the user came out from the vehicle.

2.3 User characterics

These are some usefull scenarios to understan in a better way the needs of an user.

Scenario 1 scenario di quello che si deve registrare

Scenario 2 Yesterday Marco, that is a student at Politecnico di Milano, brought his scooter to a workshop to do the annual revision. Today he has a lecture at 9:30 am at his university, so he, who do not have a subscription for public service, decides to use a car of PowerEnjoy service, thus Marco enters the PowerEnjoy App, and looks for a car nearby. When he decides what car to rent, he reserve the car, and he starts to prepare himself because he knows that he has 1 hour before the reservation expires. Following the way that the application proposed, Marco arrives at the car and tap the button, which is visualized in the application, to open the car. He starts his ride to Politecnico and, once he arrives, he parks the car and stop it, to end the ride.

Scanario 3 Filippo usually go to work with the public transport, because he thinks that he does not have to buy a car in a big city like Milan, in fact Milan is full of other service that Filippo could use to move in the city. However, today the public transport society of Milan call a general strike and Filippo must go to work in another way. During his little breakfast, he decides to open his Laptop and go to the PowerEnjoy website. He clicks on the login button and he types his credentials. After that, he specifies his actual location and he look for a car nearby. When he finds it, he reserves the car and he turns off his Laptop. After ten minutes, Filippo is ready to go and he takes his smartphone to see the way to the reserved car in the PowerEnjoy application. Once arrived at the car he taps on the open button and wait the opening of the car. He starts his ride to work and, once he arrives, he parks the car and stop it, to end the ride.

Scanerio 4 Today Marco has to go to his grandfather, but his scooter is still in the workshop, so he has to take for another time the PowerEnjoy service. He opens the application, and look for a car nearby, when he finds it he decides to reserve the car. After that, he goes out from his home and he starts to walk in order to reach the car. Once arrived, Marco notice that his smartphone has a very slow internet connection, and sometimes he remains only with a signal that can allow him to make call or send SMS. He tries to tap the open button in the application, and in that moment, the application visualizes a message to make Marco know that he can open the car answering the SMS that the service sent to him when the reservation resolves successfully. Marco open the message application in his smartphone, he finds the message and he notice that he must answer with the identification code of the car, that is in the left part of the car. When he sends that message, the car opens and finally Marco goes to his grandfather.

Scenario 5 Filippo has to go to an important convention of the society where he works, so he needs a car to reach that place which is about 5 km out of the centre of Milan. He decided to reserve a PowerEnjoy car, so he opens the application and he does it. When he arrives to the car, he taps the open button on the application, but after few seconds the application answer that he cannot be located, and show four white square when he can type the code located to the left of the car, in order to show that he is near the car. Once he does it, the car opens his door and Filippo start his ride to the convention, that finishes when he parks and exits from the car.

Scenario 6 Paola, who is just exit from a pub, because there is no public bus at night, decides that is the right occasion to try the new electric car sharing that is called PowerEnjoy. She opens the specific application and, once she selects the car, based on her preferences, she reaches it, following the path visualized on the map of the app. When she gets in it, she sees on the screen of the car that there is an interesting option to save some money, named "Money saving option". She selects it and inserts her destination. The system calculates automatically the recharges station nearby her destination and the ride starts. Once she is arrived at the recharge station selected before, she plugs the car and she receives a notification on the app, visualizing the final cost of her ride, including the discount.

Scenario 7 Caterina needs to go to the party of her university and she decides to reserve a car, using the PowerEnjoy application. Selected the nearest car, she reaches it and, once she gets in it, she selects the "Money saving option". She already knows the way to go, so she does not insert her destination. Once she is arrived at destination, she plugs the car and she receives a notification on the app, visualizing the final cost of her ride, including discount.

Scenario 8 Anthony lives in Milan and, when some of his American friends organized a trip to visit him and the city, he decides to show them the old town centre and some characteristic places. Because Anthony knows that it is forbidden to go downtown with his car, he decides to use the PowerEnjoy application. Once selected the car, visualizing information about its state (battery, autonomy, place where it is parked etc.. . .), he reaches the car with his three friends. Started up the car, the sensors on the seats notice that there are more than two passengers, so the system is notified to give a discount of 10% at the end of the ride. Arrived to the destination, Anthony, tapping on the screen, terminates his ride and the screen notifies that he is going to have a discount due to the numbers of passengers.

Scenario 9 Today Guido has an important meeting with a society that is interested in the application of the new renewable energy. To share the principles of that new technology company he decides to try the service develop by the PowerEnjoy society, which allowed, with a simple subscription to its website, to use an electric car. After Guido registers in the system, he reserved a car near his home from the website. Reached the car, he starts his ride and, because the office of the society is near his home, he arrives at destination in about 10 minutes. The system detects that the car still has 80% of battery so, it applies 20% discount to the cost of the ride because the user has left more than 50% of the battery. Once comes back home, Guido sees on the website how much he spent, and he sees the discount of his ride as well.

Scenario 10 Paola decides to reserve a car to go to the train station to wait for one of her friends coming to Milan to look for a job. Because her friend Margherita has some baggage, she reserved an electric car, using the PowerEnjoy application. When she arrives at the station, she sees on the screen of the car that the parking area of the station is one of the "special parking area". So, she immediately decides to park in this area, knowing that she can obtain a discount if she plugs the car. Plugged it, she goes to the right platform to wait for Margherita and she receives a notify from the app about the final fare with the 30% discount.

Scenario 11 It is Sunday morning and Marco needs to go to San Siro Stadium to watch the match between Milan against Inter. He is late, so he decides to reserve an electric car to drive in the preferential lane, avoiding the traffic. He opens the PowerEnjoy application and he reserves a car. Marco knows that near the stadium there is a "special parking area", but when he arrives he notices that this area is closed due to security purposes. So, he is forced to park the car 3 km away from the special area and, because the initial level of battery was more than 70%, he pays the charges of 30% more in the cost of the ride.

Scenario 12 It is Sunday and Jack with his family decides to go to the hippodrome. They decide to use an electric car so Jack opens the PowerEnjoy applications and reserve it. He drives until the stadium, but, because of the traffic, the battery level goes down 20%. Marco has taken the car with a level battery better than 70% and he does not recharge the car. Therefore, when he finishes the ride, the system applies a fee, about 30% of the cost of the ride, to compensate for the cost required to charge the car on site.

Scenario 13 Paola goes to work and once she has arrived she logs in the web site using the company's computer. The system creates a list of electric cars for Paola and her collegue, to make their work more efficient as possible. In actuak fact the list is composed by a set of nearby vehicles. Paola can see, for the rest of the day, both via web site and web application the list of vehicles that she and her collegue have to take care. At this point Paola and her partner take one company's machine, and start their work, reaching the first car in the list.

2.4 Constraints

2.4.1 Regulatory policies

The users must accept the "License agreement terms and conditions" to use the software that will be develop and to permit to PowerEnjoy company the use of general statistic over these data. The systems must require the authorization in order to treat with the sensible data that users will give in the registration phase. In order to receive the company's newsletter the users must dive an additional permission.

2.4.2 Hardware requirements

The essential use of that system require:

- A computer with an available internet connection;
- A mobile phone with an active number, that is capable to send and receive SMS.

To use the mobile application:

- A smartphone with:

  - Android: KitKat (4.4) and later;
  - iOS: version 6 or later;
  - Windows Mobile: version 10 or later;

- An active number, with a plans that contains internet connection:
  - 3G;
  - 4G: not required, but recommended.

- An amount of space in order to install the application.

To use the system on a website:

- An up to date broswer.

2.4.3 Interfacing with other systems

- Interface with PayPal system, following the rules describe in PayPal developer documentations. Fuelled by that requirements analysis, the PayPal service that fit better in that system is "PayPal Pro", that gives all set of payment that the customer needs.

- Interface with an SMS gateway, with SMS APIs owned by the SMS provider. That fact in order to send message tothe users of the system.

2.4.4 Parallel operation

The server must support parallel request both from users and employees.

2.5 Assumptions and dependencies

We supposed that these properties hold in the analysed word:

- All the cars are already accessible by the system.

- All the cars have already a way for recharge their battery, in special parking area and on site;

- The safe areas are already defined;

- The special parking areas are already defined;

- All cars can park in all the parking available without pay any fine;

- All cars contain a set of sensors that allow us to take all the information about the state of the car that we need, as the state of the battery, the number of passengers, and all other stuff that a normal auto contains;

- All the users use the system in the respect of the law;

- All the employees use the system in the respect of the law defined by the company;
- There is an external company that deals with accident;
- There are other employees, of our company, that deals with car parked in unsafe area;
- If the user cannot open the car with the standar commands, at the headquarter, the operators have a special command to open cars.
- All cars have a functional GPS connection that works without internet;
- The fine must be applied only if the initial battery level was more than 70%;
- Before the ride starts, the system keeps in memory the initial state of the car, to verify, at the end, the conditions for a discount;
- The manufacturer of the car has already provided it with a touch screen;
- The manufacturer of the car has already provided it with a data connection;
- There is a call center, deals with the emergency situation, localized in the headquarter;
- The employees works in couple, because of safety law.

3 Specific requirements

3.1 Use Case diagrams

3.1.1 User

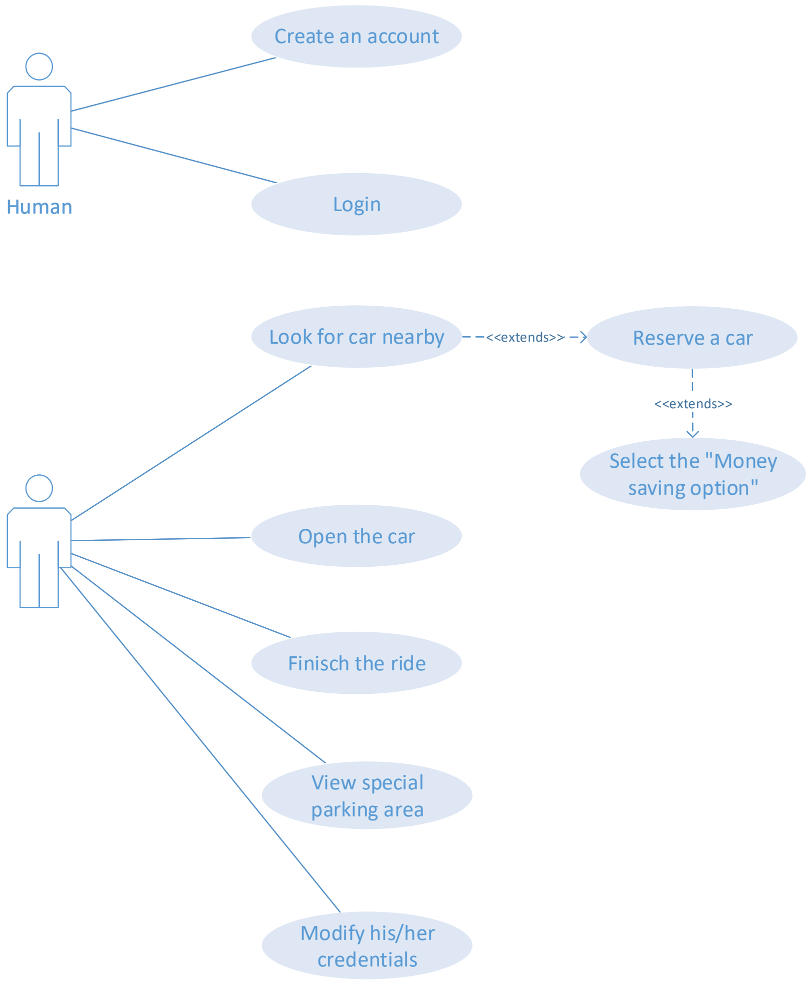


Figure 1: Use Case - User (and human).

Create an account

Name	Create an account.
Actors	Human.
Entry conditions	The human has an internet connection.
Flow of events	1. The human opens the website or the app and he/she selects the option “Sign up”. 2. The human inserts all his/her credentials. 3. The system verifies the correctness of the credentials inserted.
Exit conditions	The system sends the password to the mail address of the new user.
Exceptions	• The system detects that the email address is wrong or does not exist so it does not send the password. The system redirects the human on the registration page, asking him to write correctly his credentials. • The PowerEnjoy staff detect that the driving license inserted is not valid, so the system sends an email to the new user, asking to take a better picture of his/her license. • The system does not successfully withdraw the symbolic amount using the payment information inserted. The system redirects the human on the registration page, asking to insert the payment information correctly.

Table 1: User, Use Case description - Create an account.

Login

Name	Login of the user.
Actors	User.
Entry conditions	The user has already registered in the system.
Flow of events	• The user must open the PowerEnjoy website or the application. • The user must click or tap the button “Sign in”. • The user inserts the email, which he/she has already used for register himself/herself, and his password. • The user must click/tap on the log button.
Exit conditions	The system redirects the user to his/her personal page.
Exceptions	The user does not insert the correct login credentials. The system does not redirect him/her to his/her personal page but it requests to check if he/she has inserted the correct login credentials.

Table 2: User, Use Case description - Login of the user.

Look for car nearby

Name	Look for car nearby.
Actors	User.
Entry conditions	The user must be logged in the application or in the website.
Flow of events	1. The user must select the specific option (tapping on the phone or clicking with the mouse). 2. The user can insert his position manually if he/she does not have a GPS connection. 3. The user clicks or taps the option to start the research.
Exit conditions	The user visualizes correctly the cars nearby him/her position.
Exceptions	• The system does not detect the position through the GPS, so it retries for 30 seconds. If the GPS connection is still unavailable, the system asks to the user to insert manually his/her position. • The system detects that the positions inserted manually by the user does not exist, so it asks to the user to write an existent location. • The system is not able to establish the position of the cars, so it shows an error message asking to try again.

Table 3: User, Use Case description - Look for car nearby.

Reserve a car

Name	Reserve a car.
Actors	User.
Entry conditions	The user has looked for a car nearby.
Flow of events	1. The user can visualize the state of the car; 2. The user selects one of the car near his/her position, tapping on the screen or clicking on it; 3. The user could decide to visualize the way to reach the car selected; 4. The user click or tap on the screen the reserve option to officialise it.
Exit conditions	• The system shows to the user the reservation time; • The system enables the option to delete the reservation; • The user receives the reservation message on his phone number.
Exceptions	Another person has reserved the car that the user selected. The system shows a message to communicate the fact to the user, asking him if he/she wants to select another car.

Table 4: User, Use Case description - Reserve a car.

Select the "Money saving Option"

Name	Select the “Money saving option”.
Actors	User.
Entry conditions	• The user must have reserve a car; • The user must get into the car.
Flow of events	1. The user visualizes the screen in the car; 2. The user selects the “Money saving option” tapping on the screen; 3. The user can input the final destination and he/she could visualize the way to reach it; 4. The user can select a special parking area near his/her destination and he/she could visualize the way to reach it.
Exit conditions	The user confirms his/her choices and the system notifies on the screen that the “Money saving option” is enabled.
Exceptions	The screen of the car is temporally unavailable. The user is forced to call the emergency number that he/she can find in the car, asking to the call center to enable the “Money saving option” for his/her ride, telling to the operator his choices about the option chosen.

Table 5: User, Use Case description - Select the “Money saving option”.

Open the car

Name	Open the car.
Actors	User.
Entry conditions	• The user must have reserved a car. • The user must reach the car, and he/she can see it.
Flow of events	1. The system detects, thanks to the GPS, the position of the user; 2. If the system establishes that the user is near the car, it enables the button to open the car in the application; 3. If the user has reserved the car using the website, he/she could open the reservation message sent to him/she by the system when he has done the reservation; 4. The user execute the command to open the car, but the car does not open. The user call the emergency number written on the car. The operator at the call center verifies the correctness of the reservation and then he/she opens the car with a special command.
Exit conditions	The system receives correctly the command to open the car and it opens it.
Exceptions	• The user taps the button on the application, but there is no connection. In this case, the user is forced to open the message that the system sent to him/she when he reserved the car. He/she answers to this message, writing the code of the car. The system receives the message and it opens the car; • The user loses the signal of the GPS. In this case, the user is forced to open the message that the system sent to him/she when he/she reserved the car. He/she answers to this message, writing the code of the car. The system receives the message and it opens the car; • The user execute the command to open the car, but the car does not open. The user call the emergency number written on the car. The operator at the call center verifies the correctness of the reservation and then he/she opens the car with a special command.

Table 6: User, Use Case description - Open the car.

Finish the ride

Name	Finish the ride.
Actors	User.
Entry conditions	The user must has opened a car.
Flow of events	1. The user parks the car in a safe area. 2. The system enables the option “Finish the ride” on the screen. 3. The user selects this option tapping on the screen of the car. 4. The system visualizes on the screen the final information of the ride, in particular the final fare (without including eventually discount) and the level of the battery.
Exit conditions	The user receives the bill of the ride through email.
Exceptions	• The user parks the car in an unsafe area. In this case, the system puts the car unavailable and adds it to a list of car of an employee, in order to move it in a safe area. When the employee finishes his/her work the system gives to the user an “unsafe fine”. • The screen of the car is temporally unavailable. The user is forced to call the emergency number that he can find in the car, asking to the call centre operator to finish his/her ride.

Table 7: User, Use Case description - Finish the ride.

View special parking area

Name	View special parking area.
Actors	User.
Entry conditions	The user must be logged in the system.
Flow of events	1. The user must select the specific option on the screen (tapping on the phone or clicking with the mouse); 2. The user can insert his position manually if he/she does not have a GPS connection; 3. The user click or tap the option to start the research.
Exit conditions	The user correctly visualizes the special parking area near his/her position. The system visualizes only the special parking are which have some free plugs.
Exceptions	• The system does not detect the position through the GPS, so it retries for 30 seconds. If the GPS still undetectable, the system asks to the user to insert manually his position; • The system detects that the position inserted manually by the user does not exist, so it asks to the user to write an existent location; • The system is not able to establish the positions of the special parking areas, so it shows an error message asking to try again.

Table 8: User, Use Case description - View special parking area.

3.1.2 Employee

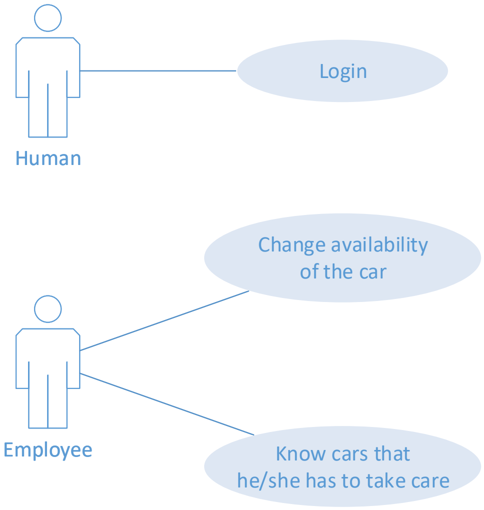

Login
Human
Change availability of the car
Employee
Know cars that he/she has to take care

Figure 2: Use Case - Employee

Login

Name	Login.
Actors	Employee.
Entry conditions	The employee must work for the PowerEnergy society.
Flow of events	1. The employee must open the PowerEnjoy application or the website; 2. The employee must insert his ID and his password; 3. The employee must click the log-in button.
Exit conditions	The website or the app redirects him to his personal page.
Exceptions	The employee does not insert the correct ID or the correct password. The system does not redirect him/her to his/her personal space/page but it requests to check if he/she has insert the correct ID or the correct password.

Table 9: Employee, Use Case description - Login.

Change availability of the car

Name	Change availability of the car.
Actors	Employee.
Entry conditions	• The employee must be logged in the system; • The car is in the employee's list of cars; • The employee reaches the car.
Flow of events	1. The employee puts it in unavailable mode; 2. The employee takes care of the car.
Exit conditions	The car is ready and the employee changes its state in available.
Exceptions	The employee cannot put the car available because of serious damages of the car. He calls the headquarters in orders to communicate that he needs to bring the car at the workshop.

Table 10: Employee, Use Case description - Change availability of the car.

Know cars that they have to take care

Name	Know cars that they have to take care.
Actors	Employee.
Entry conditions	• The employee is at work; • The employee is logged in the system.
Flow of events	1. In the headquarter, the system assigns to the employee a list of car that he must take care during the day; 2. This list is created in the personal page of the employee.
Exit conditions	The employee receives correctly the list of car that he must take care.
Exceptions	The employee does not visualize the list in his/her personal page, so he/she is forced to ask it to the general secretary of the headquarter.

Table 11: Employee, Use Case description - Know cars that they have to take care.

3.2 Sequence diagrams

Login of the user

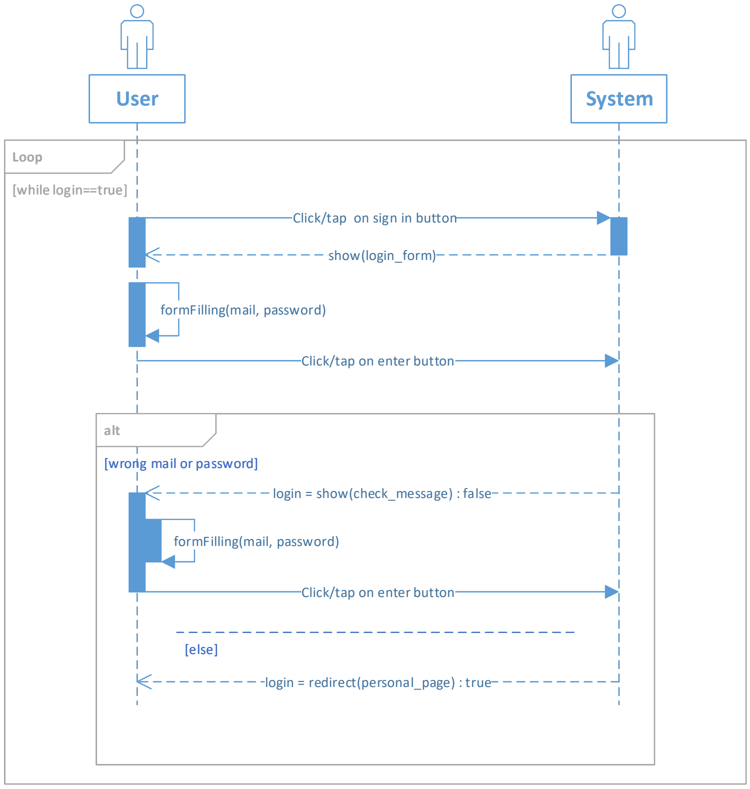

User
System
Loop
[while login==true]
Click/tap on sign in button
show(login_form)
formFilling(mail, password)
Click/tap on enter button
alt
[wrong mail or password]
login = show(check_message) : false
formFilling(mail, password)
Click/tap on enter button
[else]
login = redirect(personal_page) : true

Figure 3: Sequence Diagrams - Login of the user.

Reservation and opening of the car

This sequence diagram represent the scenario with these hypothesis:

- The user reserves a car using the PowerEnjoy application.
- The user wants to visualize the way to reach the car.
- The user open the car tapping on the application.

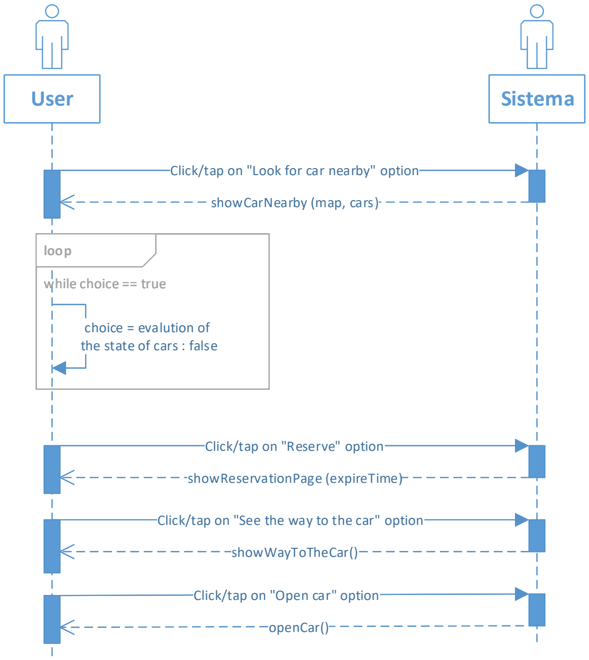
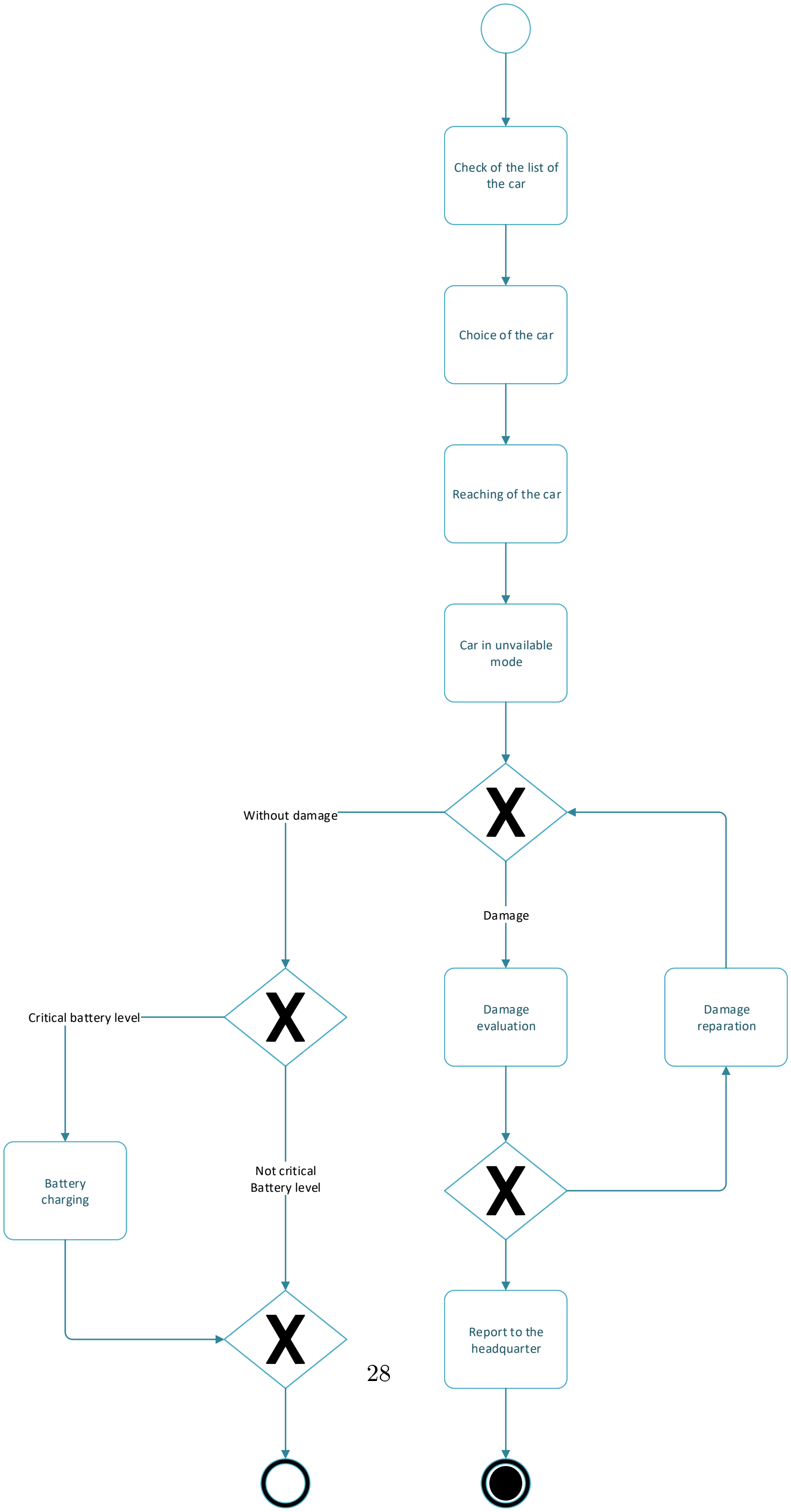

User
Sistema
Click/tap on "Look for car nearby" option
showCarNearby (map, cars)
loop
while choice == true
choice = evalution of
the state of cars : false
Click/tap on "Reserve" option
showReservationPage (expireTime)
Click/tap on "See the way to the car" option
showWayToTheCar()
Click/tap on "Open car" option
openCar()

Figure 4: Sequence Diagrams - Reservation and opening of the car (sequence diagrams).

3.2.1 Activity diagrams

Change availability of the car


Check of the list of the car
Choice of the car
Reaching of the car
Car in unvailable mode
X
Without damage
Damage
Critical battery level
X
Damage evaluation
Damage reparation
Battery charging
Not critical Battery level
X
X
Report to the headquarter

Selection of “Money saving option”

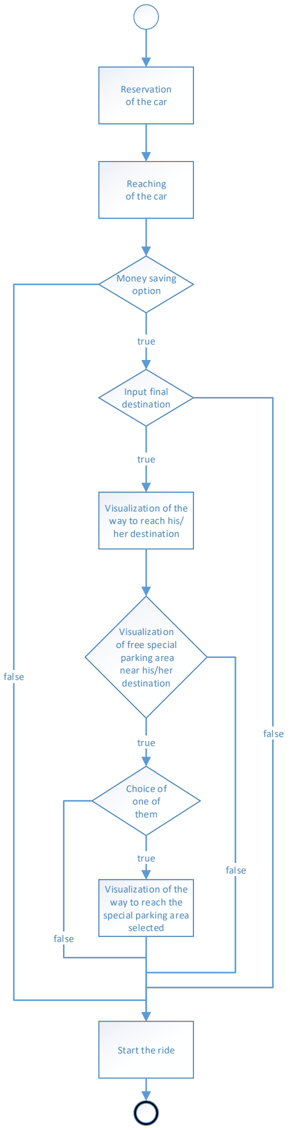

Reservation of the car
Reaching of the car
Money saving option
true
Input final destination
true
Visualization of the way to reach his/ her destination
Visualization of free special parking area near his/her destination
false
false
true
Choice of one of them
true
false
Visualization of the way to reach the special parking area selected
false
Start the ride

Figure 6: Activity Diagrams - Selection of “Money saving option”.

3.2.2 State chart diagrams

Phase of the car

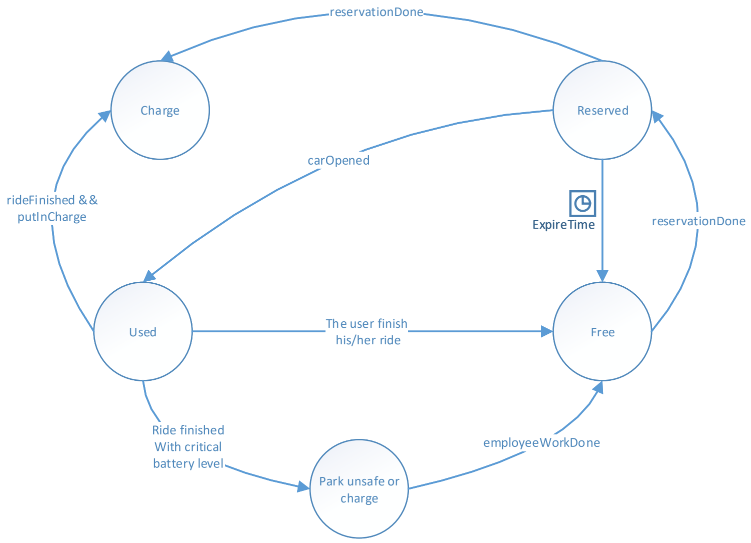


Figure 7: State Chart - Phase of the car.

3.3 Activity diagrams

3.4 State chart diagrams

3.5 User interface

3.5.1 Website

3.5.2 Mobile application

3.5.3 Screen of the car

4 Alloy

————————————————Position things—————————

```
    sig Position{
    latitude: Int,
    longitude: Int
    }{
    latitude >=0
    latitude <=3
    longitude >=0
    longitude <=5
    }
    //the positions are uniquely determinate by the place
```

```
fact differentPosition{
all p1, p2: Position| (p1!=p2)implies(p1.latitude != p2.latitude or p1.longitude != p2.longitude)
}
——————————————————Safe Areathings——————————
sig SafeArea{
safePositions: some Position,
}
fact differentPositionInDifferentSafeArea{
all s1,s2: SafeArea| all p: s1.safePositions| (s1 !=s2) implies ((p in s1.safePositions) and (p not in s2.safePositions))
}
//in our world there are always unsafe position
fact positionConstraint{
all s: SafeArea| some p: Position| p not in s.safePositions
}
————————————————Special Area things——————————
sig SpecialArea{
specialPositions: some Position,
plugs: Int,
}{
plugs >=0
plugs <=5
}
fact plugsConstarint{
all s:SpecialArea| #(s.specialPositions) = s.plugs
}
fact specialAreaConstraint{
all p: SpecialArea.specialPositions | (p in SafeArea.safePositions) &&
(#(SpecialArea.specialPositions) < #(SafeArea.safePositions))
}
fact differentPositionInDifferentSecialArea{
all s1,s2: SpecialArea| all p: s1.specialPositions| (s1 !=s2) implies ((p in s1.specialPositions) and (p not in s2.specialPositions))
}
————————————————————-Car things——————————
//there are no less than two seats
sig Car{
code: Int,
seats: Int,
state: one State,
}{
code>=0
seats>=2
seats<=5
}
```

```
fact indentificationByCode{
all c1, c2: Car | (c1 != c2) implies (c1.code != c2.code)
}
pred Car.isUsed[]{
#(this.state.usedBy)=1
}
pred Car.isInSafe[]{
one s: SafeArea| this.state.statePosition in s.safePositions
}
pred Car.isInSpecial[]{
one s: SpecialArea| this.state.statePosition in s.specialPositions
}
pred Car.isAvailable[]{
!(this.isUsed) and (this.state.batteryLevel >1) and (this.isInSafe)
}
pred Car.isNotAvailable[]{
this.isUsed or this.state.batteryLevel <1 or !(this.isInSafe)
}
pred Car.isInList[l1:List]{
this in l1.cars
}
————————————————————-State things——————————
sig State{
batteryLevel: Int,
phase: one Phase,
passengers: Int,
statePosition: one Position,
usedBy: lone User
}{
//per future implementazioni meglio passenges > < seats dell'auto
passengers >=0
batteryLevel >=0
batteryLevel <=3
}
fact numberOfState{
#Car = #State
}
fact passengersConstraint{
all c:Car| ((c.state.phase=used) implies (c.state.passengers<=c.seats)) and((c.state.phase!=used) implies (c.state.passengers=0))
}
//this means that there are not cars in the same position
fact differentPositionInCar{
all c1,c2 : Car | (c1!=c2) implies (c1.state.statePosition != c2.state.statePosition)
}
```

```
————————————————-Phase things————————
//these are the possible internal state of the car
enum Phase {reserved, used, charge, free, parkUnsafeOrCharge}
fact reservedPhase{
all c: Car| (c.state.phase = reserved) iff (c.isInSafe and c.state.batteryLevel>1
and c.isUsed and (c.state.phase!=used and c.state.phase!=charge and c.state.phase!=free
and c.state.phase!=parkUnsafeOrCharge))// batteryLevel >=1 usata
}
fact usedPhase{
all c: Car| (c.state.phase = used) iff (c.isUsed and c.state.batteryLevel != 0
and (c.state.phase!=reserved and c.state.phase!=charge and c.state.phase!=free
and c.state.phase!=parkUnsafeOrCharge)) //batteria >0 e deve essere usata
}
fact chargePhase{
all c: Car| (c.state.phase = charge) iff (!(c.isAvailable) and c.isInSpecial and
(c.state.phase!=reserved and c.state.phase!=used and c.state.phase!=free and
c.state.phase!=parkUnsafeOrCharge))// position in special e non è usata
}
fact freePhase{
all c: Car| (c.state.phase = free) iff (c.isAvailable and (c.state.phase!=reserved
and c.state.phase!=used and c.state.phase!=charge and c.state.phase!=parkUnsafeOrCharge))// batteria maggiore di 1 non è usata non è in carica
}
fact parkUnsafePhase{
all c: Car| (c.state.phase = parkUnsafeOrCharge) iff (!(c.isAvailable)and
/*(c.isInSafe or !(c.isInSafe)) and */ (c.state.phase!=reserved and c.state.phase!=used
and c.state.phase!=charge and c.state.phase!=free))// posizione è in unsafe e
non è usata
}
————————————————-User things————————
sig User {
licenseID: Int, //should be an alphanumeric value in the real world
}{
licenseID > 0
}
//this means that there are not equal users
fact differentIDUser{
all u1, u2: User | (u1 != u2) implies (u1.licenseID != u2.licenseID)
}
fact differentUserOnCar{
all c1,c2: Car| (c1 != c2) implies (c1.state.usedBy != c2.state.usedBy)
}
————————————————-Employee things————————
sig Employee {
employeeID: Int,
} {
```

```
employeeID > 0
}
fact uniqueEmployee{
all e1,e2: Employee| (e1 != e2) implies (e1.employeeID != e2. employeeID)
}
——————————————————-PairOfWorkers things—————————
sig PairOfWorkers {
employees: some Employee
} {
#employees = 2
}
fact uniquePairOfWorkersForEmployee {
all e :Employee| lone p : PairOfWorkers| ((e in p.employees))
}
fact mapPairOfWorkersOnLists{
#PairOfWorkers = #List
}
fact differentPairPerList{
all l1,l2: List| (l1!=l2) implies (l1.pair!=l2.pair)
}
——————————————————-List things—————————
sig List{
code: Int,
cars: some Car,
pair: one PairOfWorkers,
}{
code>0
#cars>0
}
fact uniqueList{
all l1,l2: List| (l1!=l2) implies (l1.code!=l2.code)
}
fact phaseOfCarInTheList{
all c: List.cars| c.state.phase=parkUnsafeOrCharge
}
fact phaseOfCarInTheList{
// all c:Car| one l:List| (c.state.phase=parkUnsafeOrCharge) implies (c in l.cars)
}
fact differentList{
all l1,l2: List| all c: l1.cars| (l1!=l2) implies !(c.isInList[l2])
}
fact listConstraints{
all c:Car| (c.state.phase=parkUnsafeOrCharge) implies (c in List.cars)
}
pred show(){
```

```
//some c:Car| c.state.phase=reserved
//some c:Car| c.state.phase=used
//some c:Car| c.state.phase=charge
//some c:Car| c.state.phase=free
//some c:Car| c.state.phase=parkUnsafeOrCharge
// l: List| #(l.cars) =2
}
run show for 1 but exactly 15 Position, exactly 1 SafeArea, exactly 2 SpecialArea, exactly 7 Car, exactly 7 State, exactly 6 User, exactly 5 Employee, exactly 2 PairOfWorkers, exactly 2 List
```

5 Appendices