Software Engineering 2: “PowerEnJoy”
Requirements Analysis and Specifications Document (V. 1.1)

Authors:
Sergio CAPRARA
Soheil GHANBARI
Erica TINTI

Milan, Italy
13/11/2016

Table of contents

1. Introduction

1.1 Revision History

Version	Date	Authors	Description
1.0	13/11/2016	S. Caprara, S. Ghanbari, E. Tinti	First release
1.1	07/02/2017	S. Caprara, E. Tinti	Corrections and updates

1.2 Purpose

The purpose of this document is to provide a description of the system to be developed. It specifies goals of the system, functional and non-functional requirements and assumptions, other than use cases, possible scenarios and UML diagrams. Also, the system has been modelled using Alloy Analyzer and the code is provided with results of the simulation.

The document is addressed to programmers involved in the developing phase and to the client that has to revise the contract basis for the system development.

1.3 Description of the given problem

The aim of PowerEnjoy is to offer a car-sharing service to users, so that they can reserve, use and release electric cars.

The user can register to the PowerEnjoy service by using a mobile application that communicates with a system. To get access to the service the user must provide driving license, ID card and payment details; after system validation, a password will be sent for accessing to the account.

Registered users have complete access to the functions of the application, being allowed to see available cars around them or close to a user specified address and to ask for a reservation. Users are also given the possibility to cancel the reservation.

After the reservation, a one hour time limit is imposed to reach the car, and only once the user gets close, the system unlocks it; the user can enable an option provided to save money. In that case the system designates the place where to park and the user can get a discount if he confirms and parks there.

The user is charged at the end of the rental and the due amount is calculated on the time of the ride; in some cases, depending on the number of passengers, the battery level of the car and the place in which the user parks it, he may get discounts or get some extra charge.

Discounts and extra charge are calculated as follows:

- -10% off if the car has at least two other passengers;
- -20% off if the car is left with more than 50% of the battery;
- -30% if the car is parked and the users plugs it into the power grid;
- +30% if the car is left at more than 3km from the nearest power grid station or with less than 20% of the battery.

The service provided is available in the Italian area and a European driving license is required to register. The system knows the coverage of the service, the valid parking areas (safe areas) and the list of the special parking areas (as a list of coordinates). Moreover, it can get information on the status of all cars (battery level, position, etc.) and can get users' locations via their GPS enabled on mobile phones.

Cars that need maintenance for any reason and that are unavailable are not shown on users' maps.

On the other side, another type of user, operator, has a different access to the service. Their objective is to do interventions and maintenance on cars that are unavailable for the following reasons:

- the car is left with less than 5% of the battery level;
- the car needs maintenance for issues reported through its control unit.

Operators interact with the system through a mobile application providing a login page for their identification, the list of cars in need of maintenance and a function to take them in charge in order to fix the issue. Once the work done, they inform the system and the car becomes available again.

1.4 Glossary

- **Car Sharing:** it is a car rental service that people can use to reserve cars, use them, usually for a short time, and then leave them in a parking.

- **User:** a person that is registered to the PowerEnjoy Service. He must be older than 18, and have a valid driving licence to drive cars. He can reserve PowerEnjoy cars using the PowerEnjoy mobile application.
- **Operator:** a person with technical skills, such as a mechanic, that fixes issues and takes PowerEnjoy cars to a special parking area and plug them at Power Plug when their battery is low. Since they are not PowerEnjoy service users, they can't reserve a car and perform any operation users do.
- **App:** It is the short term we use to define a mobile application.
- **Power Plug:** it is a column with one electricity socket where it is possible to plug the car. Each socket has a number written on it and a led showing its status, green if available, red otherwise.
- **Safe Area** (also Parking Area): it is an area with parking spaces. These spaces are not especially reserved to PowerEnjoy, but shared with all the other divers of the town. We will refer to "Safe area" also with the term "Parking area".
- **Special Parking Area** (also Power Station): a parking area reserved exclusively to PowerEnjoy cars where, for each parking space there is a Power Plug where the car can be plugged. We will refer to "Special parking areas" also with the term "Power Station".
- **Car:** in this document with the word "car" we refer exclusively to PowerEnjoy cars.
- **System:** it is the system that manages all the operations to make the car sharing work, getting information from external systems, and interacting with the user’s terminal, through the mobile application.
- **Pin:** it is a symbol on the map to represent the position of some element. Usually pins have different symbols depending on what they represent.
- **Money Saving Option:** it is an option that, if activated, shows the user some indication to save money during the car rental.
- **Energy Level:** it refers to the level of the battery of a car. It is shown on the car's dashboard.
- **Reservation:** it is the relation that creates between a user and a car. It allows the user to request a car and keeps the request active up to 1 hour. The reservation guarantees that no one else can reserve and use the reserved car till the expiration of the reservation.
- **Green E-box**: the system installed on cars, with which the system can interact to get information on the status of the car.

1.5 Reference Documents

- Specification document: Assignments AA 2016-2017
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

2. Overall Description

2.1 Product Perspective

The Data System interfaces with the system mounted on each car for getting real-time information through API exposed by this system.

The Data System also access to Power Plugs information to get their status and to set the led on the Power Plug. To allow this, each Power Plug should have an internal system that exposes some API and that is able to communicate with our system.

For payments, our system interfaces with an external Payment System sending all payment data such as credit cards information and the total amount to be paid. Given these information, the payments system performs the payment and should send a feedback to our system.

2.2 Actors

The actors that participate to our system are:

- **User**: it is a person that has registered to our system and has downloaded the app on his phone.
- **Operator**: it is a person, already known by the system, that can access to the list of the cars that have low battery or that need technical assistance.

2.3 Goals

User goals

[G1] The user should be able to register to the system.

[G2] Already registered users should be allowed to login.

[G3] The user should be able to find available cars around him or in a specified address.

[G4] The user should see the battery level of a car before making a reservation.

[G5] The user should be able to reserve a car.

[G6] The user should be granted the access to the reserved car once he reaches it.

[G7] The user should be allowed to cancel a reservation.

[G8] The user should be able to access profile and payment method and make changes.

[G9] The user should be informed of the amount he has been charged of.

[G10] After the rental, the user should be able to inform the system that he is leaving the car.

[G11] The user should be able to see all the parking areas.

[G12] The user should be able to see all the special parking areas.

[G13] User should be blocked if there is a problem in the payment.

[G14] The user should be banned if the payment is refused during a 30 days period.

Operator goals

[G15] Operators should be allowed to login to the system.

[G16] Operators should be allowed to see the list of cars that need maintenance and their details.

[G17] Operators should be allowed to notify the take in charge of a car and the end of the maintenance.

2.4 Assumptions

- The user interacts with the system using his mobile phone.
- The user has internet access enabled on his phone.
- Each car has GPS sensors (always turned on) providing the right coordinates, sensors to check the number of people sitting, a dashboard showing the status of the battery level and a screen displaying the current charge to users.
- Special parking areas are a subset of safe parking areas and correspond to the ones with power plugs reserved to PowerEnjoy cars exclusively. PowerEnjoy cars can only be charged and plugged in special parking areas.

- Safe parking areas and special parking areas are already defined and identified by coordinates: the system is initialized at start-up time with such a set of areas.
- Only one user at a time can drive a specific car.
- Each car has a maximum number of five seats.
- Each car is equipped with a Green E-box, that makes it possible to get information on the car, such as battery level, number of passengers, technical issues, …
- A car is unavailable for reservations when it has less than 5% of battery level, has technical issues, or it is already in use or reserved by another user. Otherwise, it is available.
- A car that has less than 5% of battery or technical issues needs maintenance.
- Only registrations by users having a European driving licence are supported.
- The car can be driven anywhere in the Italian area.
- User payments are made using credit cards.
- Operators are already registered to the system.
- Discounts can be combined so that their total amount is subtracted to the final charge.
- In case the company receives a fine concerning a bad user behaviour caused by some violation, the user is due to pay it, so the system pays it in advance and then charges the amount to the user.
- The system can always access to the real-time information of the car, such as its position, the number of people inside it, the battery level, the charging status and control unit information. Depending on this information, the system can manage the status of the car.
- The system can check the availability of power plugs of all special parking areas at any time. The system can also set the status of a power plug if needed.
- Power plug status can be available or unavailable and is identified by the colour of a LED on it. This way other users won't use a reserved plug.
- A user can use a car if and only if he has reserved it in advance.
- A car can be reserved by only one user at a time.
- The system has access to European Union driving licence databases to verify and approve user registration.
- Only registered users can access to system functionalities.

2.5 Stakeholders

The stakeholder of this project is PowerEnjoy Company, a new company born from a famous Energy distribution company, that decided to build a car sharing service that would offer a service to the citizens that would be advantageous and eco-friendly at the same time.

3. Specific Requirements

3.1 Functional Requirements

User goals

[G1] The user should be able to register to the system:

- The user must enter all his personal data, ID card data and driving licence information.
- The system should validate the data entered by asking the external system for driving licence verifications.
- If correct, the system should provide to the user a password to access the service.

[G2] Already registered users should be allowed to login:

- The user should be allowed to enter login information.
- The system should validate entered data and allow the access if all is correct.

[G3] The user should be able to find available cars around him or in a specified address:

- The user can specify an address or use his GPS coordinates.
- The system can get user coordinates accessing GPS on user's phone or get coordinates of the specified address.
- The system should provide a map on which available cars are identified by a symbol showing the correct position.

[G4] The user should see the battery level of a car before making a reservation:

- The system should provide information about the car when the user picks one of the available, before reserving it.

[G5] The user should be able to reserve a car:

- The user should be able to select one of the cars to see its status and reserve it.

- The system should make the car unavailable to other users once the user confirms his choice.
- The system should retain the reservation for an hour and cancel it when the time limit is reached. In this case a 1 EUR fee should be charged.

[G6] The user should be granted the access to the reserved car once he reaches it:

- User should be able to notify the system that he has reached the car.
- The system should check user coordinates to compare them with car's ones and unlock it if the distance is less than 10m.

[G7] The user should be allowed to cancel a reservation:

- The system should offer an option to the user to cancel the reservation.
- Once the reservation is cancelled, the user should be able to access again to the map with available cars.

[G8] The user should be able to access profile and payment method and make changes:

- The system should provide a personal profile to the user with his information and give the possibility to apply changes.

[G9] The user should be informed of the amount he has been charged of:

- The system should calculate the total charge after subtracting all the accumulated discounts.
- The system should notify the user about the final amount of the reservation and how it was calculated.

[G10] After the rental, the user should be able to inform the system that he is leaving the car:

- The user should have a way to notify the system he has finished using the car.
- The system should verify the status of the car and check the parking.
- If all is correct the system should lock the car and make it available again.

[G11] The user should be able to see all the parking areas:

- The system should provide a map on which parking areas are identified.

[G12] The user should be able to see all the special parking areas:

- The system should provide a map on which special parking areas are identified by a symbol showing their correct position.

[G13] User should be blocked if there is a problem with the payment:

- The system should try to perform the payment at the end of the rental.
- The system should block the user account when the payment is refused.
- The system should retry the payment for the next 30 days.

[G14] The user should be banned if the payment is refused during a 30 days period:

- After 30 days, the system should ban the user from the service if the payment is still refused.

Operator goals

[G15] Operators should be allowed to login to the system:

- The operator should be allowed to enter login information.
- The system should validate entered data and allow the access if all is correct.

[G16] Operators should be allowed to see the list of cars that need maintenance and their details:

- The system should provide a list of cars that need maintenance and show all details.

[G17] Operators should be allowed to notify the take in charge of a car and the end of the maintenance:

- The system should provide a way to take in charge a car.

- The operator should be able to notify the system that the maintenance has finished.

3.2 Scenarios

Scenario 1

Alice gets out with friends but when she wants to come back home there are no buses, so she decides to take a PowerEnjoy car. She opens the application (she's already logged in) and she checks for the closest available car on the map. She finds some and she decides to reserve one, that's 10 min from her by feet. When she arrives at the car, she presses the button to pickit up and after a few seconds the car opens. She starts driving. When she arrives home, she parks in the first available parking space and presses the button to notify she's leaving the car. Since that parking space is in a safe are, the reservation is correctly concluded. She closes the door and she waits till the car locks.

Scenario 2

Rob should go to a restaurant not easily reachable by public transport. Then he decides to take a PowerEnjoy car. He opens the app and sees that the closest available car is 10 minutes from him by metro. He decides to take it anyway. Unfortunately, there are some traffic problems with the underground, then he decides to take a bus to get to the car. At that time, there is too much road congestion in his town, and the bus is late. The battery of his mobile phone is low and when he gets off the bus his phone turns off. Now Rob will not be able neither to pick up the car nor to cancel the reservation, then he will pay 1€ fee.

Scenario 3

Alex is at the restaurant with his girlfriend, and they're eating the dessert. He knows they will be out of the restaurant in half an hour, then he decides to have a look at the PowerEnjoy application on his phone. He notices that there is an available car just in front of the restaurant, then he immediately reserves it before someone else do it. When they pay the amount, Alex's girlfriend proposes him to have a walk around there. Then Alex opens again the app and decide to cancel the reservation.

Scenario 4

Omer is getting out of work when he notices that is raining cats and dogs. Then he decides to take a PowerEnjoy car to get back home, if there is one nearby. He finds one and reserve it. In a few minutes, he arrives close to the car, opens it, and drives home. Unfortunately, he forgot that the credit card he connected to PowerEnjoy is almost empty and when he lets the car close, the system notifies him that the payments has failed. The application shows him the total amount that he should pay, the last four numbers of his credit card and the deadline for the payment. The system will retry the payment for the next 30 days and he won't be able to reserve a car unless one of the next tries succeeds. The day after he recharges the credit card connected to PowerEnjoy. This time the payment is successful and the system notices him that he is again allowed to reserve a car.

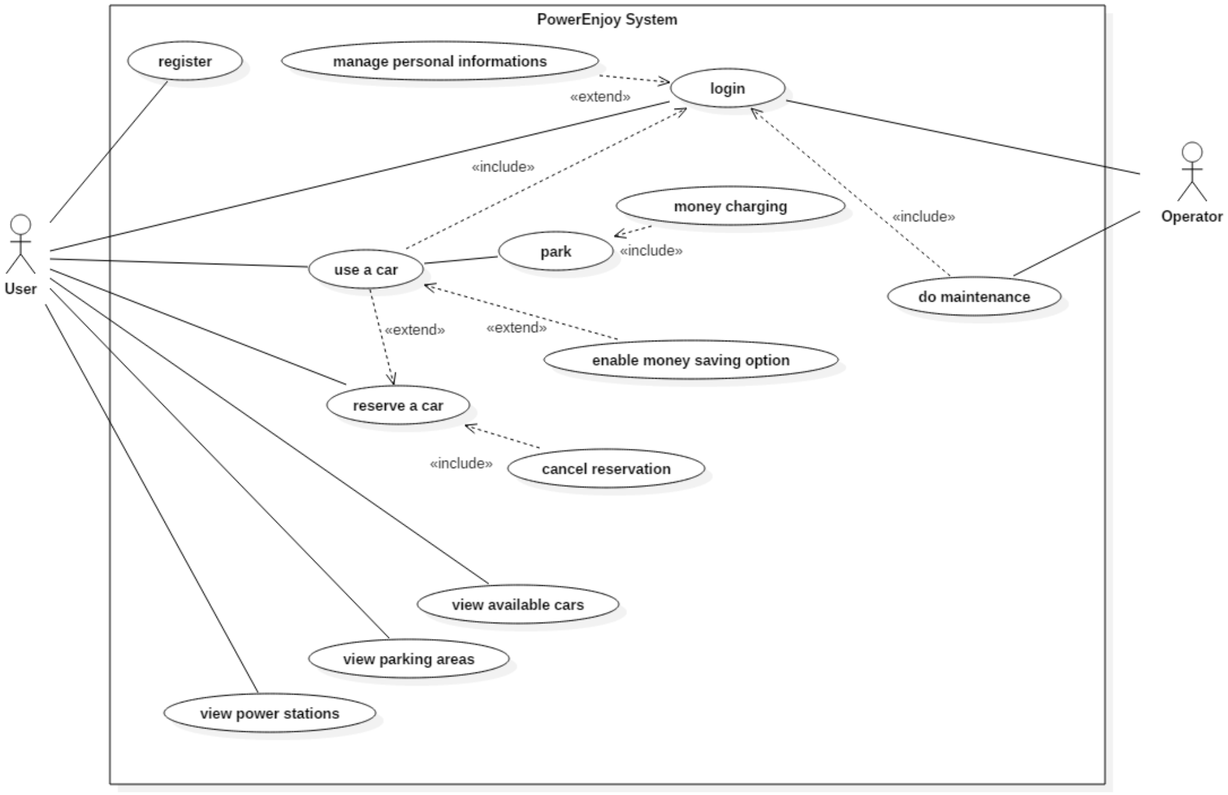
Scenario 5

Klarissa is a student and she likes to save money. By night, even if buses are cheaper, she prefers to move by car because she thinks it will be safer. Like every weekend, she's out with her friends, and now it's time to get back home. Her friends don't live far from her, and since they are only in 4, she offers them to bring them all home too, this way she knows she will have the 10% off. She finds a car close to them and she reserve it. The car has the 40% of energy, then she will not receive this kind of discount but anyway it's enough to bring everyone home. When she gets into the car, she activates the money saving option, and then she enters his address. The system shows the possible special parking areas close to her destination and she chose one that is just 300mt from home. The computer board inform her that it has reserved the charging columns 4 at the selected special parking area. She drives her friends their own home and then she goes to the selected special parking area and she plug the car in the right energy tower. When she closes the car, she gets notified on her phone that she achieved a total discount of 40%, then she only spent 2€! Not bad!

3.3 UML Diagrams

3.3.1 Use Cases

Use cases diagram



Use cases description

Name
User registers to the system

Actors
User

Flow of events

- The user accesses the application on his mobile phone.
- The user fills all the fields, entering his personal data and ID card and driving licence information.

- The user clicks the Confirm Registration button.
- The system validates entered data comparing them with those of the external system and confirms the registration.
- The system sends the password for the login on user email account and mobile phone.

Exit conditions
The user is redirected to the login page

Exceptions
Entered data for the driving licence is not valid, the system notifies the users and suggests to enter correct data. User is not allowed to register until he enters correct or valid information. The user is already registered

---

Name
User logs in

Actors
User

Entry Conditions
The user must be already registered to the system

Flow of events

- The user opens the loginActivity and enters login data.
- The system validates entered data.

Exit conditions
The user has access to the map showing available cars in his area.

Exceptions
The user account does not exist, the system notifies the user and suggests to register in case he has not.

Name
Manage personal info

Actors
User

Entry Conditions

The User has been logged in and decided to change the information.

Flow of events

- The user Click view the profile
- The user input information if user would like to update profile
- The user click Submit/Update Profile.
- The system notify user by showing a message or through email.

Exit conditions

The user has deleted/updated successfully.

Name

View available cars on the map

Actors

User

Entry Conditions

The user is logged in

Flow of events

- The system checks user's position using GPS and shows an area around him on the map.
- The application shows a pin corresponding to the position of an available car (close to him or close to an address entered by the user).
- By clicking on a pin, the details of that car are shown: battery level and position. A button for reserving that car is shown.

Exit conditions

The user clicks on the button to reserve a car or on the back button.

Name

Use of a car

Actors

User

Entry Conditions

The user has reserved a car and has reached it within one hour after the reservation.

Flow of events

- The user arrives close to the car and notifies the system by clicking the Unlock button.
- The system verifies user position using GPS.
- The system checks the position of the car.
- The system checks the distance and unlocks the car.
- The user opens and starts the car.
- At the end, the user parks the car and stops it.
- The user clicks the End button to inform the system.
- The system validates the position of the car and ensures it is in a safe parking area.
- The user exits the car.
- The system checks car's battery level, distance from special parking areas, number of passengers during the trip and charging status (if it is plugged).
- The system calculates the total of the ride and subtracts the discounts, if that's the case.
- The system entrusts the external payment system to pick up the final amount from the user.
- The system locks the car and marks it as available again.

Exit conditions
The system shows the user information about his last trip.

Exceptions
The user is not close to the car: the system denies the access to the car and provides a message to the user asking him to get closer to unlock.

The user is trying to park in a non-safe parking area: the system informs the user that he can't end the reservation and should move the car in an allowed place.

The payment fails due to missing money on the credit card or blocked card: the system receives a failure message and makes the user unable to reserve other cars. The system retries the payment during the next 30 days. If still refused, the system bans the account forever and delegates the problem to a debt collecting company.

Name
Allow the user to reserve a car

Actors

User

Entry Conditions
The user opens the detail of a car on the map

Flow of events

- Viewing the details of a car, the user notify that he wants to reserve that car
- The system checks if the car is still available
- The system marks that car as unavailable and start the countdown.
- On the user application sees the countdown, and he can notify that he wants to release the car, or that he arrived close to the car and he want to open it. The car position is still shown.

Exit conditions
Countdown starts

Exceptions
When the user notifies he want to reserve that car, if it is no longer available a message is shown on the user application.

Name
Cancel a reservation

Actors
User

Entry Conditions
The user views the details of the car he reserved and the countdown

Flow of events

- The user notifies that he wants to cancel the reservation of that car
- The system marks that car as available again and stops the countdown
- The system notifies the user

Exit conditions
The user views the map and can reserve another car

Name
View Parking Areas

Actors

User

Entry Conditions
User has logged in.

Flow of events

- The user clicks on the button to see parking areas.
- The system shows on the map all the parking areas by highlighting them.

Exit conditions
The user has decided to finish the viewing of parking area by clicking on other buttons.

Name
View Power Station

Actors
User

Entry Conditions
User has logged in.

Flow of events

- The user clicks the button to see power stations.
- The system shows on the map all the power stations.
- The user clicks on the pin of one of the areas.
- The system shows a list of the power plugs and their availability.

Exit conditions
The user has decided to finished the searching with clicking on other buttons.

Name
User chooses money saving option

Actors
User

Entry Conditions
The user has entered the car.

Flow of events

- The system asks the user for the destination.
- The user enters the address.

- The system checks entered address and looks for free power plugs in close stations (considering the distribution of cars in the city).
- The system shows the result to the user asking for confirmation.
- The user accepts to park in the station selected by the system.
- The system reserves the free power plug.
- The system provides the address and number of the slot reserved for the power supply.

Exit conditions
The user sees information and directions to the slot and the power station.

Exceptions
The address entered by the user does not exist or is out of the area: the system informs the user of the error and suggests to retry. The user refuses to park in the suggested station: the system closes the money saving option. No free slots available: the system informs the user that there are no charging slots available, so it suggests the user to retry later.

Name
Operator logs in

Actors
Operator

Entry Conditions
The operator opens the application.

Flow of events

- The operator enters login data.
- The system validates entered data.

Exit conditions
The operator has access to the list of the cars that need maintenance.

Exceptions
The entered data is wrong: the system notifies the operator.

Name
Operator does the maintenance on a car.

Actors
Operator

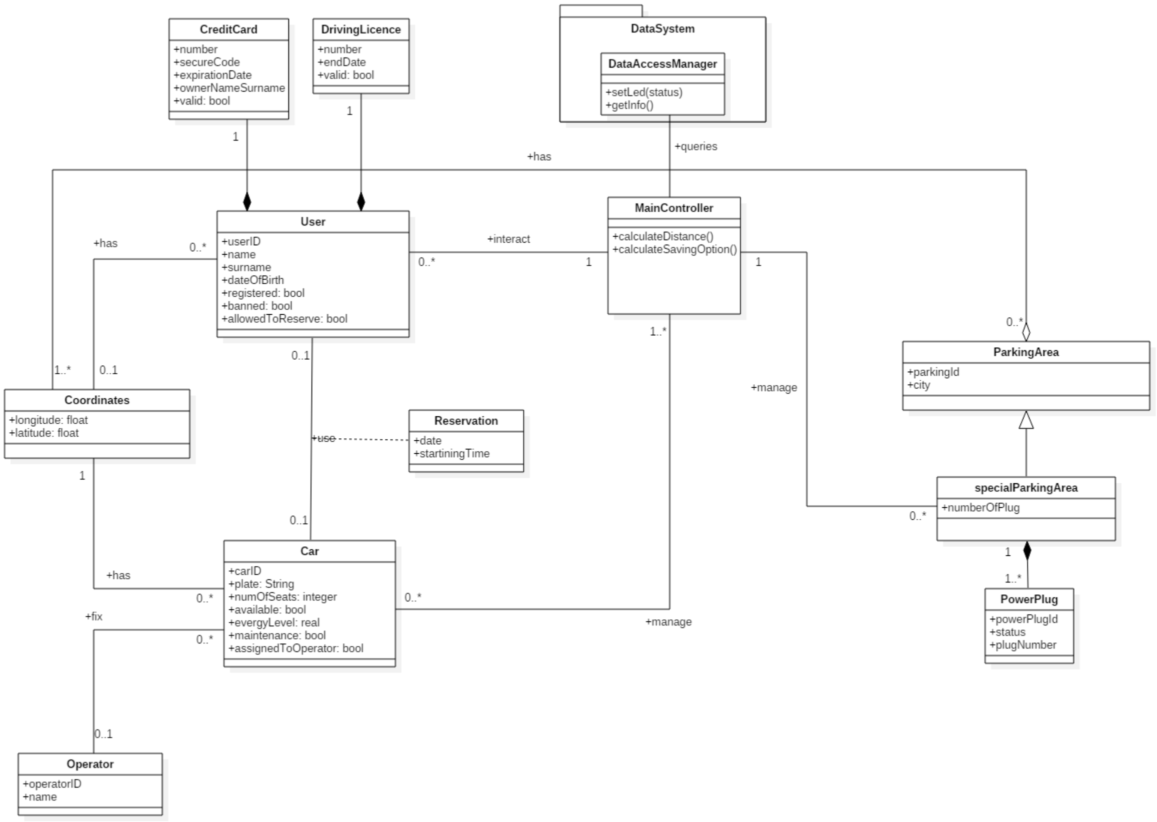
Entry Conditions
The operator opens the application.

Flow of events

- The system provides a list of the cars in need of maintenance.
- The operator chooses a car to fix and notifies the system.
- The system sets the status of the car as assigned.
- The operator reaches the car and fixes it.
- The operator notifies the system that the maintenance has finished.

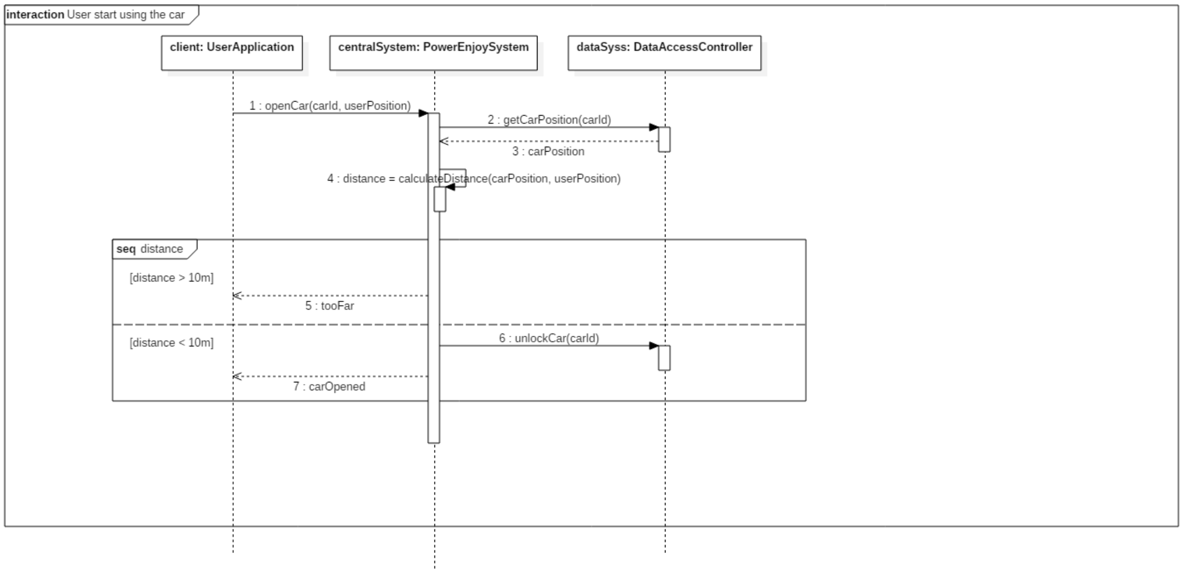
Exit conditions
The car is marked as available by the system.

3.3.2 Class Diagram

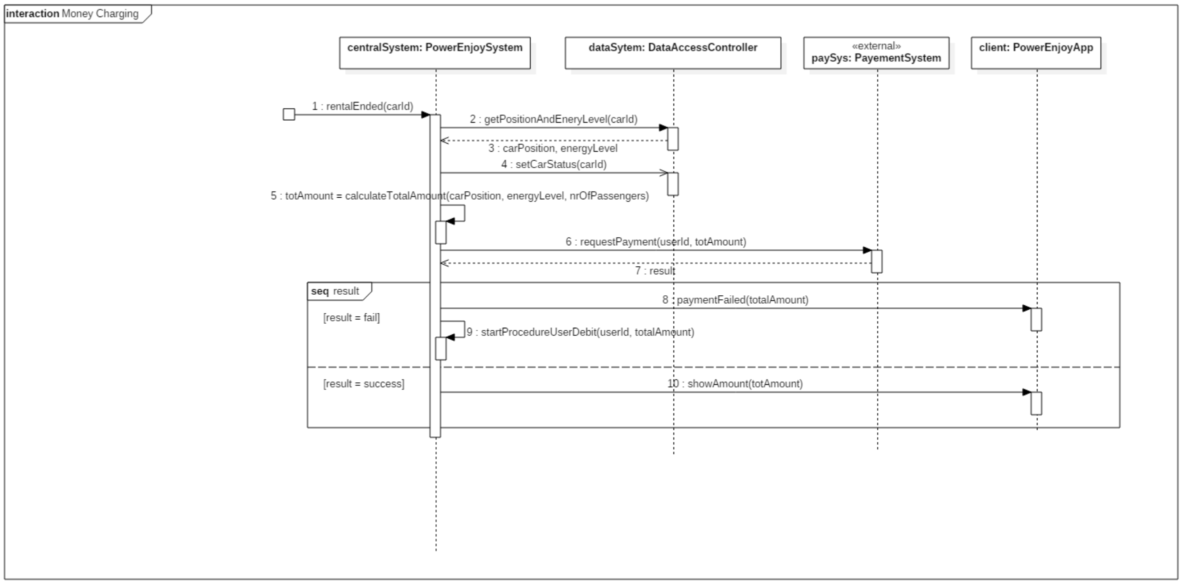


3.3.3 Sequence Diagram

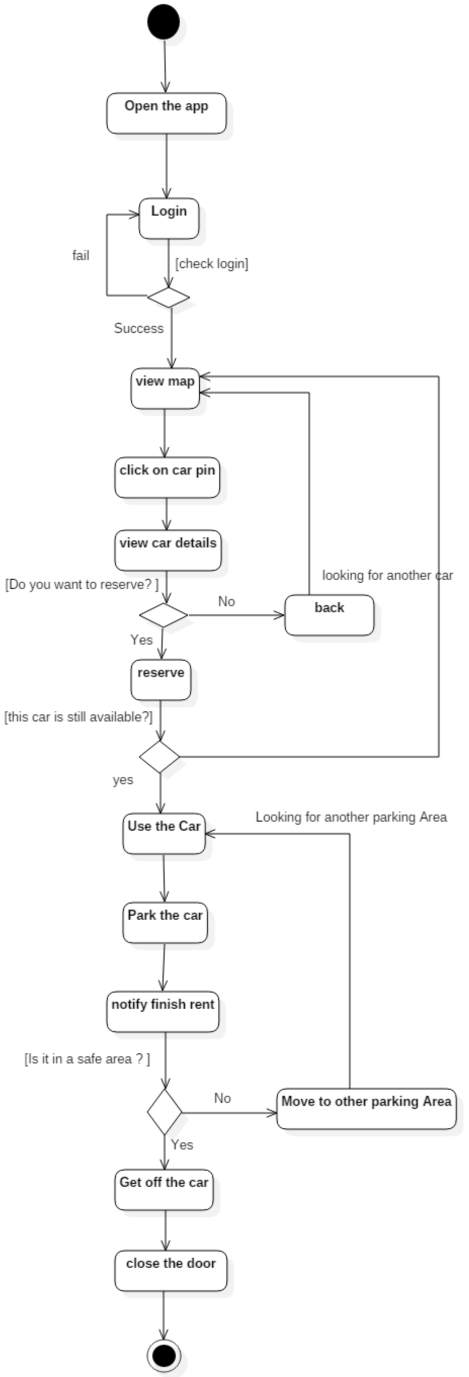
Start of the use of the car


interaction User start using the car
client: UserApplication
centralSystem: PowerEnjoySystem
dataSyss: DataAccessController
1 : openCar(carId, userPosition)
2 : getCarPosition(carId)
3 : carPosition
4 : distance = calculateDistance(carPosition, userPosition)
seq distance
[distance > 10m]
5 : tooFar
[distance < 10m]
6 : unlockCar(carId)
7 : carOpened

Money charging

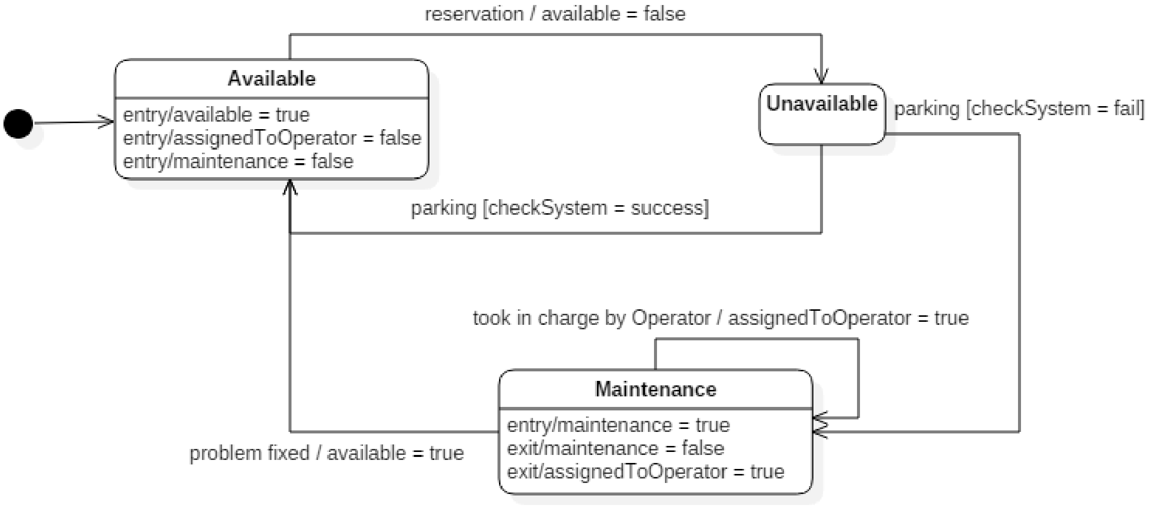

interaction Money Charging
centralSystem: PowerEnjoySystem
dataSytem: DataAccessController
«external»
paySys: PayementSystem
client: PowerEnjoyApp
1 : rentalEnded(carId)
2 : getPositionAndEneryLevel(carId)
3 : carPosition, energyLevel
4 : setCarStatus(carId)
5 : totAmount = calculateTotalAmount(carPosition, energyLevel, nrOfPassengers)
6 : requestPayment(userId, totAmount)
7 : result
seq result
[result = fail]
8 : paymentFailed(totalAmount)
9 : startProcedureUserDebit(userId, totalAmount)
[result = success]
10 : showAmount(totAmount)

3.3.4 Activity Diagram

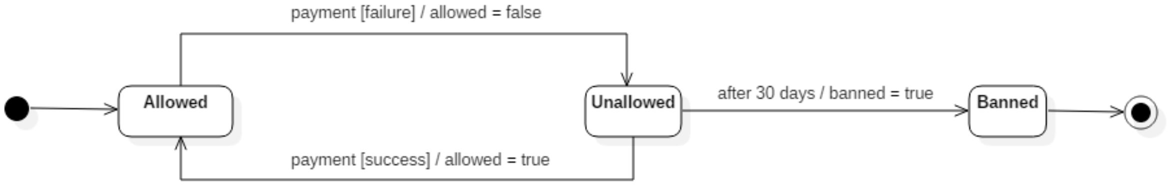


3.3.5 State Diagram

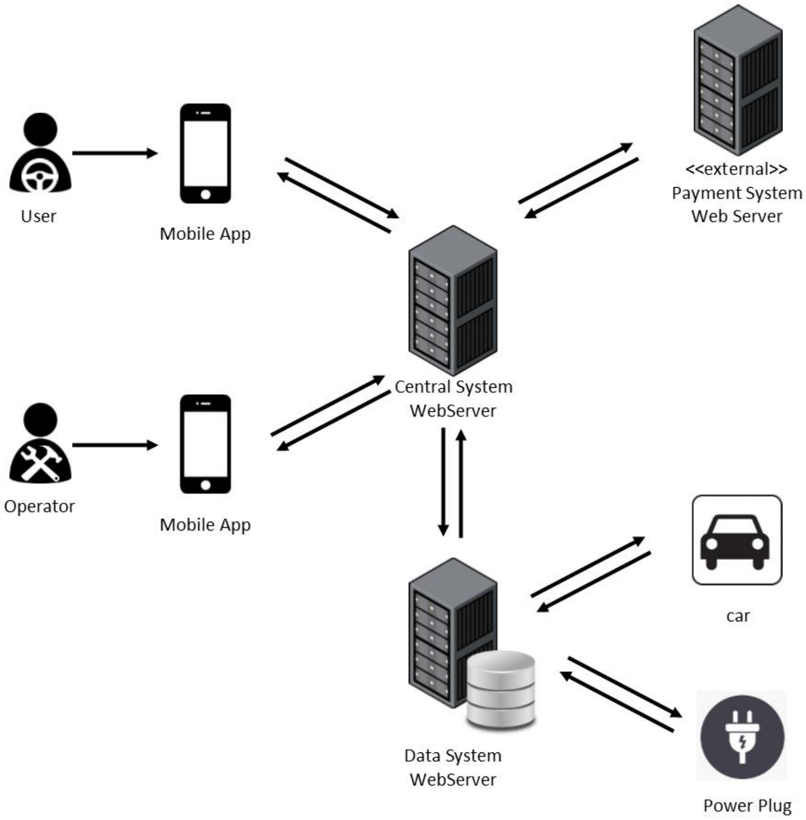
Car States


reservation / available = false
Available
entry/available = true
entry/assignedToOperator = false
entry/maintenance = false
Unavailable
parking [checkSystem = fail]
parking [checkSystem = success]
took in charge by Operator / assignedToOperator = true
Maintenance
entry/maintenance = true
exit/maintenance = false
exit/assignedToOperator = true
problem fixed / available = true

User States


payment [failure] / allowed = false
Allowed
Unallowed
after 30 days / banned = true
Banned
payment [success] / allowed = true

3.4 Proposed System



Users and Operators interact with PowerEnjoy System through a mobile application on their smartphones. The PowerEnjoy Central system runs on a WebServer and interacts with the users and operators' applications, with a DataSystem for accessing to cars and power plug information, and with an external Payment System for requesting a payment by the user. The PowerEnjoy Central System also queries a database for checking users' credentials.

3.5 Non-Functional Requirements

3.5.1 Interfaces

User Interfaces

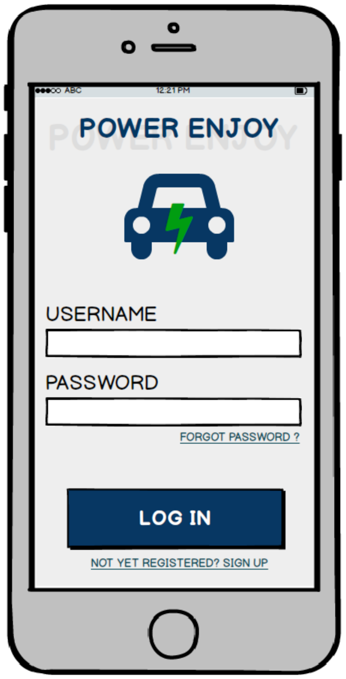


Fig. 1 User login

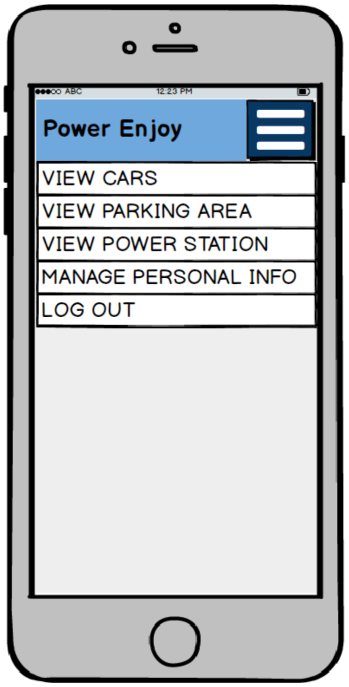


Fig. 2 User menu

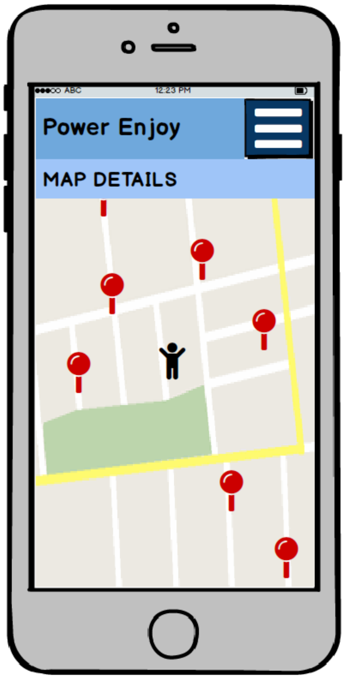
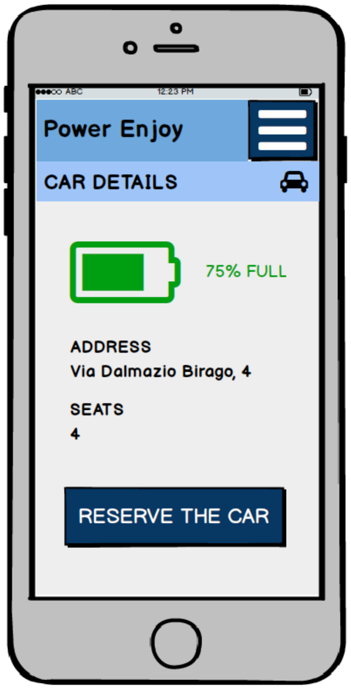


Fig. 3 Map view

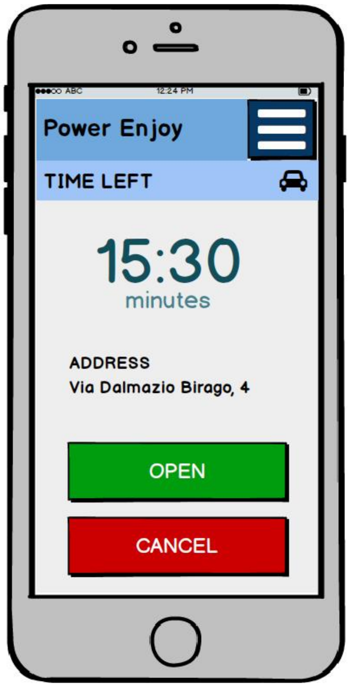


75% FULL

ADDRESS
Via Dalmazio Birago, 4

SEATS
4

Fig. 4 Car Details



15:30
minutes

ADDRESS
Via Dalmazio Birago, 4

Fig. 5 Reservation countdown

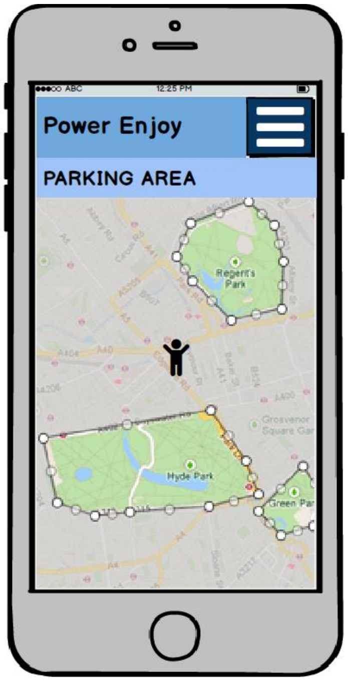


Fig. 6 Parking Areas view

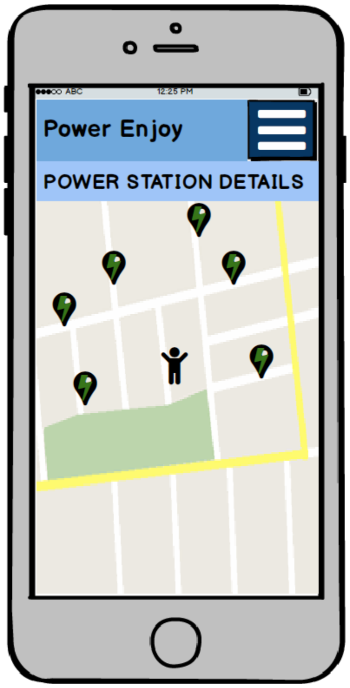
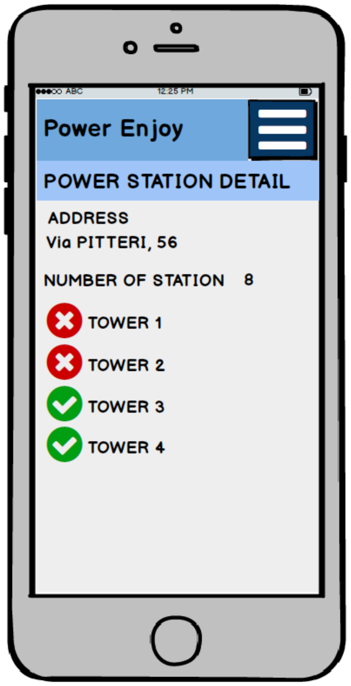


Fig. 7 Power grid stations view


Power Enjoy

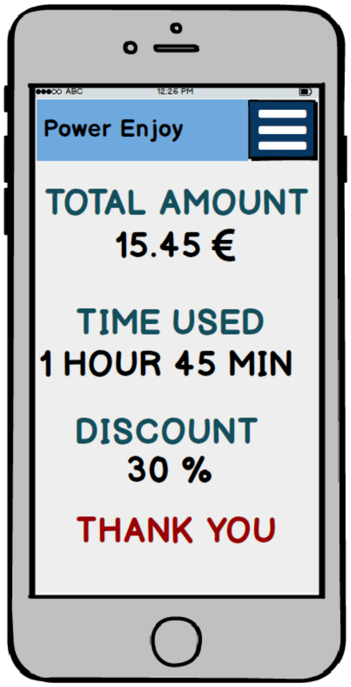
POWER STATION DETAIL

ADDRESS
Via PITTERI, 56

NUMBER OF STATION 8

✗ TOWER 1
✗ TOWER 2
✓ TOWER 3
✓ TOWER 4

Fig. 8 Detail of power grid station


Power Enjoy

TOTAL AMOUNT
15.45 €

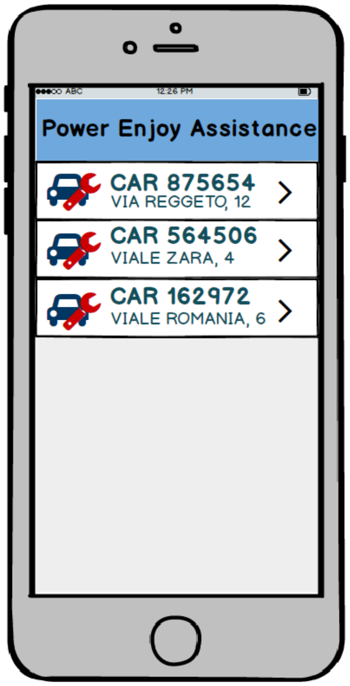
TIME USED
1 HOUR 45 MIN

DISCOUNT
30 %

THANK YOU

Fig. 9 Final amount charged info

Operator Interfaces

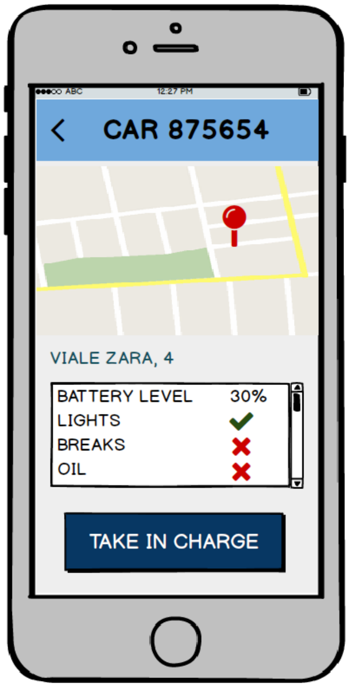


CAR 875654
VIA REGGETO, 12

CAR 564506
VIALE ZARA, 4

CAR 162972
VIALE ROMANIA, 6

Fig. 10 List of cars with issues



VIALE ZARA, 4

BATTERY LEVEL	30%
LIGHTS	✓
BREAKS	✗
OIL	✗

Fig. 11 Detail of chosen car

4. Appendices

4.1 Alloy Modelling

4.1.1 Alloy Code

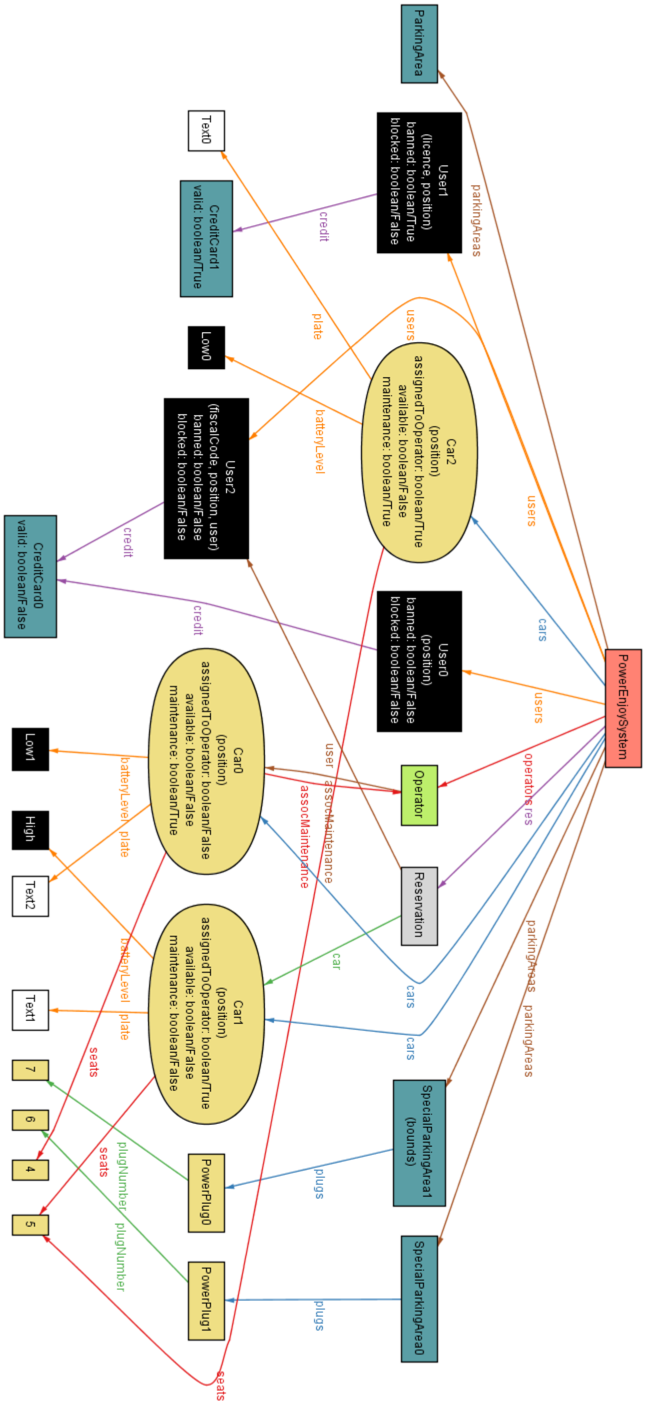
```
open util/boolean

sig Float {}

sig Text {}

abstract sig BatteryLevel {}

sig Low extends BatteryLevel {}

sig High extends BatteryLevel {}

sig Car {
	plate: one Text,
	seats: Int,
	available: one Bool,
	batteryLevel: one BatteryLevel,
		position: one Coordinates,
		maintenance: one Bool,
		assocMaintenance: lone Operator,
		assignedToOperator: one Bool
} {
	seats > 0
	seats <= 5
}

sig Coordinates {
	longitude: one Float,
	latitude: one Float
}

sig Licence {
	valid: one Bool
}


sig CreditCard {
	valid: one Bool
}

sig FiscalCode {
	user: one User
}

sig User {
	licence: one Licence,
	credit: one CreditCard,
	fiscalCode: one FiscalCode,
	blocked: one Bool,
	banned: one Bool,
	position: Coordinates
}

sig Reservation {
	user: one User,
	car: one Car
} {
	user.blocked = False
	user.banned = False
	user.licence.valid = True
	car.available = False
	car.batteryLevel = High
	car.maintenance = False
}

sig Operator {
	assocMaintenance: lone Car
}

one sig PowerEnjoySystem {
	users: some User,
	operators: some Operator,
	cars: some Car,
	res: some Reservation,
	parkingAreas: some ParkingArea
}

sig ParkingArea{
	bounds: set Coordinates,
}

sig SpecialParkingArea extends ParkingArea {
	plugs: some PowerPlug
}

sig PowerPlug {
	status: one Bool,
	plugNumber: Int
```

```
}

// FACTS

fact powerEnjoyOwnsAll {
    PowerEnjoySystem.users = User
    PowerEnjoySystem.operators = Operator
    PowerEnjoySystem.cars = Car
    PowerEnjoySystem.res = Reservation
        PowerEnjoySystem.parkingAreas = ParkingArea
}

fact noReservationWhenMaintenance {
    all c:Car | c.maintenance = True => c.available=False
    all r: Reservation, c:Car |
        c.maintenance = True => r.car != c
    all o: Operator, r: Reservation, c:Car |
        o.assocMaintenance = c => r.car != c
}

fact noCreditCardwithoutUser {
    no c: CreditCard | c not in User.credit
}

fact noBatterywithoutCar {
    no b: BatteryLevel | b not in Car.batteryLevel
}

fact allLicencesAreOwned {
    User.licence = Licence
}

fact noFiscalCodeWithoutUser {
    all f: FiscalCode, u: User | f = u.fiscalCode <=> u = f.user
}

fact assocMaintenance {
    all c:Car, o:Operator |
    ((o in c.assocMaintenance) <=> (c in o.assocMaintenance))
}

fact userBlockedIfCreditNotValid {
    all u: User | u.blocked = True => u.credit.valid = False
}

fact plateIsUniqueForCar {
    all c1, c2: Car | c1 != c2 => c1.plate != c2.plate
```

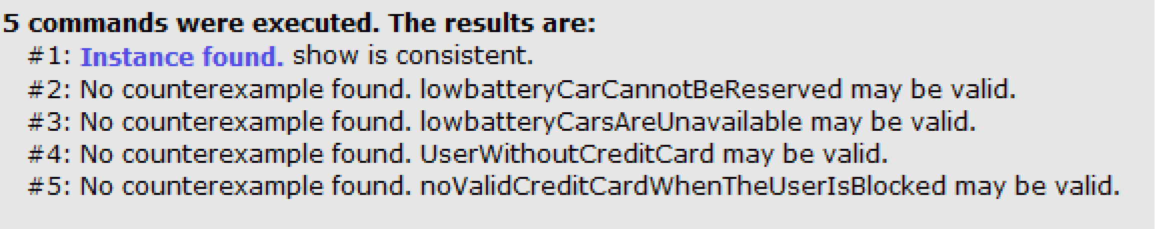
```
}

fact usersHaveUniqueFiscalCodeAndLicence {
	all u1, u2: User | u1 != u2 => u1.fiscalCode != u2.fiscalCode
		&& u1.licence != u2.licence
}

fact sameReservationDifferentUserAndCar {
	all r1, r2: Reservation | r1 != r2 => r1.user != r2.user
		&& r1.car != r2.car
}

fact lowBatteryCarUnderMaintenanceUnavailable {
	all c: Car | c.batteryLevel in Low => c.maintenance = True
		and c.available = False
}

fact noPlugWithoutSpecial {
	no p: PowerPlug | p not in SpecialParkingArea.plugs
}

// ASSERTIONS

assert lowbatteryCarCannotBeReserved {
	no r: Reservation | r.car.batteryLevel in Low
}

assert lowbatteryCarsAreUnavailable{
	all c: Car | c.batteryLevel in Low => c.available=False
		and c.maintenance=True
}

assert UserWithoutCreditCard {
	no c: CreditCard | c not in User.credit
}

assert noValidCreditCardWhenTheUserIsBlocked {
	all u: User | u.blocked=True =>  u.credit.valid=False
}


pred show() {
	//#PowerEnjoySystem = 1
	#ParkingArea = 3
	#SpecialParkingArea = 2
	#Car = 3
}
```

```
run show for 3
check lowbatteryCarCannotBeReserved for 4
check lowbatteryCarsAreUnavailable for 3
check UserWithoutCreditCard for 4
check noValidCreditCardWhenTheUserIsBlocked for 4
```

4.1.2 Alloy Model



4.1.3 Alloy Result



```
5 commands were executed. The results are:
  #1: Instance found. show is consistent.
  #2: No counterexample found. lowbatteryCarCannotBeReserved may be valid.
  #3: No counterexample found. lowbatteryCarsAreUnavailable may be valid.
  #4: No counterexample found. UserWithoutCreditCard may be valid.
  #5: No counterexample found. noValidCreditCardWhenTheUserIsBlocked may be valid.
```

4.2 Used tools

- StarUml: draw of UML diagrams
- git: documentation versioning
- github: online repository
- ReText: writing text and first MarkUp
- Microsoft Word: assembling document and check ortography
- Alloy Analyzer: system modelling
- Balsamiq: drawing mockups
- Asana: progect management - track task

4.3 Hours of work

Day	Sergio Caprara	Soheil Ghanbari	Erica Tinti
25/10/2016	2	2	--
26/10/2016	--	2.5	--
27/10/2016	2	1.5	1.5
28/10/2016	3	--	1
29/10/2016	3	2.5	--
30/10/2016	--	4	--
31/10/2016	--	--	--
01/11/2016	1	--	--
02/11/2016	2.5	3	2.5
03/11/2016	3	--	1.5
04/11/2016	2	--	2
05/11/2016	4	--	4
06/11/2016	2	3	2
07/11/2016	5	--	5.5
08/11/2016	2.5	--	4
09/11/2016	4	4	4
10/11/2016	1.5	2.5	1.5
11/11/2016	2.5	2	2.5
12/11/2016	8	4	8
13/11/2016	13	11	13
Total	**61**	**42**	**53**