POLITECNICO
DI MILANO

Computer Science and Engineering
A.A. 2016/2017
Software Engineering 2 Project:
Power&Joy

Requirements Analysis and Specification Document

November 11, 2016

Prof. Luca Mottola
Joshua Nicolay Ortiz Osorio Mat: 806568
Michelangelo Medori Matr: 878025

Index

1 Introduction

1.1 Purpose

This is the first of a series of documents aimed to project a digital management system for a car sharing service. Identifying stakeholders, modeling scenarios and formalizing requirements and constrains of the system are the main topics of this document.

1.2 Descriprtion of the given problem

We are going to design a digital management system for a car sharing service that only employs electric cars (i.e. cars powered by rechargeable batteries), which are environment-friendly and noise-free. Precisely, we want to offer the possibility to users to choose a car among an amount of cars dislocated into Milan's urban area, to travel across the city. Electric cars gather their fuel when they are plugged into proper power grids; these grids are placed all around the city: they can be found into specific charging stations, beyond near a decent number of parking lots around Milan. Cars can be parked everywhere inside Milan urban area (i.e. any kind of appropriate car park, accordingly to Italian laws, pay and display parking lots excluded). Operators are available to make sure that cars are never left with less than 20% battery charge by charging them into near charging stations or in place. The final aim of the system is to provide a service within anyone's reach, that stands for a solid alternative to public transport.

1.3 Stakeholders

The only stakeholders that we have is Power&Joy society that require the management system of car sharing and the city public administration .

1.4 Glossary

- **System:** is the system we will create which is going to manage the car sharing service. The system has a dedicated database where it can store and access all needed information.
- **User :** every person registered into the system that wants to use a car.
- **Not Register User (NRU) :** all people not registered in the system.
- **Operators:** employees working on charging stations whose job is recharge every car that has less than 20% battery level and to provide assistance to the users and issued cars.
- **Passenger :** a person who is traveling in a car (included the driver).
- **Registration (for Users):** consists in the act of inserting all the needed information into the system as

  - First name and Last name
  - E-mail
  - Password
  - Phone number
  - Birth date
  - ID card code
  - Drive license code
  - Credit card number
  - Fiscal code

- **Safe Area:** Milan's urban centre.
- **Power grid station:** a stopping place for electric cars equipped with electric socket and where operators work.
- **Available car:** is a car that is not currently used by other users and has the battery charge level over 20%.
- **Parking lot:** every available car park inside the Safe Area that respects the Italian traffic laws.
- **Special Parking lot:** every parking lot equipped with electric socket.
- **Safe Zone:** is a circular zone with 3 km of diameter which centre is a power grid station.
- **Reservation:** is the ability to reserve a car for at maximum 1 hour, then the reservation expires.
- **Contactless card:** card acquired by users at the moment of the registration used to unlock the reserved cars.
- **Special contactless card:** Magnetic card acquired by operators at the moment of the registration that can be used to unlock any car.
- **Unlock Car:** to undo the lock of car's doors by placing the contactless cards or special contactless cards near specific sensors placed over the car doors.
- **Password:** is the key to log in to the system.
- **Status (of a car):** any of the possible conditions of the cars (low charge/assistance needed/accident/no issues).
- **Reserved Operator area:** special section of the system where operators can register and log in, and from which they can access information about the position and the status of all the cars.

- **Ride:** it starts one minute after the user unlocks the car and ends when the user shuts down the car and pushes on the display "end the ride" button.
- **Pit Stop:** it happen when the user stops the car for a short period of time (maximum 60 minutes) and keeps the car reserved in order to continue the ride.
- **End the ride:** user stops the car, leaving it in a safe parking lot so that the car is made available for other reservations by other users.
- **Scalability:** is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged in order to accommodate that growth.
- **Quality of Service:** is the overall performance of a computer network, particularly the performance seen by the users of the network. His attributes are: Performance, Reliability, Scalability, Capacity, Accuracy, Availability, Robustness, Integrity and Confidentiality.
- **On-board computer:** a computer equipped with a touch screen and a GPS navigation device, placed in every car. It can also perform calls (in particular, it can perform accident calls, as defined below).
- **CID module:** a document that must be filled in case of accident by the people involved with the accident details (according to Italian traffic laws)
- **Assistance call:** call made by an user from his phone to ask for assistance about how to use the cars. These calls are picked up by the call center
- **Call center:** place where assistance calls are picked up by operators. The only job of the operators who work on the call center is to pick up assistance call (they do not provide physical assistance to users). Call center is NOT equipped with power grids.
- **Accident calls:** calls automatically started by the on-board computer when a user taps on the "REPORT ACCIDENT" button on the on-board computer.
- **Operator ID:** a code associated to each operator (written on their contactless card) that identifies them
- **UserID:** a code associated to each user (written on their contactless card) that identifies them
- **CarID** : a code associated to each electric car. It corresponds to the plate number of each car

- **ReservationID**: a code associated to each reservation made by users
- **RideID**: a code associated to each ride performed by users
- **ParkID**: a code associated to each special parking lot

1.5 Reference documents

These are the documents we used as guideline:

- Specification Document: Assignment AA 2016-2017 (RASD)
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- Examples Documents: RASD sample from Oct. 20 lecture.

1.6 Overview

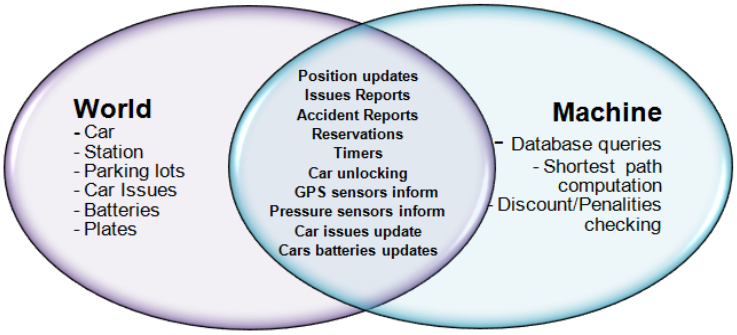
After a short description of the system's properties, we define goals (e.g. the functions the system has to be able to perform) and constrains, and make assumptions that we suppose hold in the analyzed world. Then we write down the goals and analyze and formalize them with the help of UML diagrams (use case, sequence diagrams, class diagram) and the formal language Alloy.

2 Overall Description

2.1 Product Perspective

We want this service to be accessible via mobile or web app. Every user (i.e. person who wants to reserve and ride a car) must register to the system via the mobile or web app providing general information (full name, age, e-mail etc.) and specific ID's (fiscal code, drive license code). Once registered and logged in, the system allows user to view on a map the current geographical position of every available car, thanks to GPS sensors on each car. Issued cars as well as cars already in use by other users do not display, but the system can access the information about the positions and the batteries of every single car, weather they are being used or not, any time it needs to (in case of assistance requests et al.).
Users can choose a car at the time, and make a reservation for it. They shall reach the car within 60 minutes from the moment the reservation is made, to be able to drive the car. When near, users can unlock the car by simply placing their contactless card (provided at the moment of the registration) near the sensor placed on the car door. The system unlocks the car soon after and the user is able to get in. In order to start the ride, the user simply needs to turn the key placed inside the car. Every car is equipped with an on-board computer, that can be used to communicate the existence of damages over the

car as well as GPS navigator device. Before starting the ride, the system encourages the user to make sure the car is not damaged or compromised, and to communicate it, in case issues are found, via the mobile app or the on-board computer. Once the car is started, the user starts to be charged per minute of ride.
Once they end the ride, users can park cars in every available car park inside the urban area (which includes any kind of parking lot, accordingly to Italian laws, pay and display car parks and private parking lots excluded). Some special parking lots are equipped with power grids to recharge the car, and discounts are available in case users take care to plug the car in. During the ride, the user can request to the system to guide him to the nearest special parking lot tapping a button on the on-board computer. The system then computes and displays on the GPS the shortest path to the nearest available special parking lot. Discounts are also available in case the user leaves the car with more than 50% battery charge or if he takes on the ride 2 more passengers , while the user is going to pay 30% more of the price of his ride in case he leaves the car with 20% or less battery charge or more than 3 km away from the nearest charging station. (i.e. outside every safe zone)
Payments are always carried out through the credit card that the user has to insert at the moment of the registration and a receipt is available to view accessing the history of rides from it's personal area via mobile or web app or via e-mail. A group of operators is available to give assistance to users and to make sure cars are never left under 20% battery charge. Operators work at charging stations and can access all the information about cars positions, issues and batteries.



2.2 Actors Identifying

There are two main kind of people the System needs to interact with:

- Users
- Oprators

Both of them need to be registered into the system.?Operators work at the charging stations and need to pick up and recharge every car below 20%

battery charge and provide assistance to users, if needed. The goals concerning the interaction between these two groups of people and the system are described below.

2.3 Goals (Product Functions)

Users:

- G1 The system shall allow users to register via the web or mobile app by inserting all the required information
- G2 The system shall allow users to log in
- G3 The system shall allow users to view the information about the history of rides and reservations they have made
- G4 The system shall allow users to see the positions of all the available cars on a map
- G5 The system shall allow users to make a reservation for an available car
- G6 The system shall allow users who reserved a car to cancel the reservation
- G7 The system shall allow users to unlock the car they reserved
- G8 The system shall allow users to communicate if the car they reserved is damaged or issues are found (e.g. dirt, damages, etc.)
- G9 The system shall allow users to start the ride if they don't find any issue on the car they want to ride
- G10 The system shall compute and display the shortest path from the user current position to a given location via the on-board computer
- G11 The System shall allow users to make a pit stop of maximum 60 minutes
- G12 The system shall compute and display the shortest path from the user's current position to the nearest available special parking lot, if rqeuested by the user itself
- G13 The system shall allow users to report an accident via the on-board computer
- G14 The system shall allow users to end the ride
- G15 The system shall compute the price of every ride taking into account discounts and penalties and send it to the system which takes care of the payment

Operators:

G16 The system shall allow operators to log into a dedicated area .

G17 The system shall allow operators to view the updated status and position of every car on a list

G18 The system shall allow operators to view the cars currently being assisted by other operators on a dedicated list accessible from the operator's reserved area

G19 The system shall allow operators to be able to select a car from the list in order to take care of it.

G20 The system allow an operator who took care of a car to communicate that the issues are resolved to all the other operators

2.4 Domain Properties

- The contact-less card is received by users not later than 3 days after the registration is performed.
- Parking a car where is not allowed by the traffic laws is subject to penalties for the user that parked it, according to Italian laws.
- Damages found on the cars are paid by the last user before the one who signals the found damages
- In case a user needs assistance he can call an operator with the number displayed on the window of every car
- All cars are equipped with a GPS navigation device.

2.5 Text Assumprtions

- There are no identical fiscal codes or license numbers related to different users or operators
- The credit card inserted at the moment of the registration by the user is a valid (existing and working) one
- Users only register into the system once
- The user who unlocks the car is the owner of the contact-less card he uses and is the same who is actually going to drive that car
- Cars are equipped with GPS sensors in order to be localized by the system
- Cars are equipped with pressure sensors on every seat, in order to let the system know exactly how many passengers are onto the car anytime

- Users start the ride after at maximum 1 minute after they unlock the cars
- Users will not leave the car in a zone where is not possible to detect the GPS sensor, even if it is inside the safe area
- Users do not leave cars into private parking lots or pay and display parking lots
- User will not perform any kind of maintenance or reparation operations to any car
- User will not leave the car in any kind of condition which could cause damage or any issue on the surrounding area.
- Users will not try to leave cars outside the safe area.
- Users will not try to reserve more than a car at once
- The credit card used by the users always has enough money to pay for the rides
- Special parking lots are equipped with sensors in order for the system to know how many cars are parked inside it and how many power grids are available at any time.
- Users will always leave the car with the windows closed, handbrake activated, lights and radio off and all the documents stored in their right place at the end of every ride
- Users will always instantly receive, after they end a ride, the information about the ride itself via e-mail. These information include time and money spent, distance traveled , etc
- All the payments are correctly performed at the end of every ride
- In case of issues on a car, users always notice them and notify it to the system before starting the ride
- If an operator taps the button "OPERATE" associated to an issued car, he actually performs all the needed actions in order to fix that car within 4 hours , and always remembers to tap the button "ISSUE SOLVED" once they finished
- Operators are registered into the system and all their data is stored onto the database. The information inserted by operators at the moment of the registration is always consistent
- Operators have a special section of the system where they can log and access all the information they need to perform their job. They can log using a password chosen during the registration

- Operators working on the call center are always available to pick up calls from users who need assistance
- The assistance number users can call is written on the window of every car, and is visible both from inside and from outside the car
- The number used for accident calls is different from the one used for assistance calls, and is not available for user to call unless they tap on the "REPORT ACCIDENT" on the onboard computer
- User only tap the "REPORT ACCIDENT" on the on-board computer if they were actually involved in an accident
- Once a user makes an accident call, the call is redirected to the nearest charging station, and always instantly picked up by an operator.
- Once a user makes an accident call, an operator always reaches the place of the accident in less than 15 minutes
- If the presence of a passenger is detected by the sensor at the beginning of the ride as well as at the end, the system supposes that the passenger has been on the car for the whole ride.
- User only pay for damages found on the cars if they actually performed those damages
- Accident calls are picked up by operators who do not work on the call center (only charging stations)
- Operators have a special contact-less card they can use to unlock the cars, acquired at the moment of the registration into the system
- Operators can drive any car unlocking them via the special contact-less card, and without paying for the ride
- In case of accident the user always takes care of stopping the car in some place where it does not create problems to the traffic flow, never panics, fills the CID modules (according to Italian laws) and remains in the place where the accident happened until the arrive of the operator
- an e-mail is always sent from the system to a user after he registered into the system, and any time he reserves a car , cancel a reservation or end a ride. The e-mail notifications are managed by an already existing notification system.
- Users always turn off the car engine before ending the ride or starting a pitstop
- After starting a pit-stop, user can communicate to the system that they are leaving the car clicking on a button " LEAVE CAR" on the on-board computer. If they do press that button, they always actually get off the car, within a minute, and the car locks itself after 2 minutes.

- Users will never try to leave the car taking the car key with them. The car key always stays plugged in.
- In case users leave any kind of items into the cars after ending the ride, they can inform operators about that via a phone call, and operators will pick their stuff; users will collect it in one of the charging station

2.6 Constrains

- The implementation language must be Java
- The credit card payment system must be able to be dynamically invoked by other systems relying on it

Network connections:
Power&Joy webApp needs internet connection for working. In particular if a user needs to make a reservation or cancel it , he has to switch on 3G or LAN wireless connection.

Concurrent operations
The system has to guarantee multiple processes from different connected users.

Hardware limitations
All the mobile phone must have GPS to localize and reach the cars. They required a space for App package too.

Interfaces to other applications
The system will interface with a new MySQL database. All the information about users and operators are saved here and the system through queries can use them.

3 Specific Requirements

3.1 Functional Requirements

1. **The system shall allow users to register via the web or mobile app by inserting all the required information**
   (a) The system requires to insert all personal information that consist on :
      - First name and Last name
      - E-mail
      - Password
      - Phone number
      - Birth date
      - ID card code

- Drive license code
- Credit card number
- Fiscal Code

(b) The system shall check that all information inserted are valid (e.g. all text boxes must be not empty, the name cannot contain numbers etc).

(c) The system includes the information of the new user onto the database

(d) The system sends a confirmation e-mail to the new user.

2. **The system shall allow users to log in**

   (a) In order to log in, the system shall require users to insert their credentials (consisting of an e-mail/username and a password) into an input form accessible from the web or mobile app.

   (b) The system shall verify that the credentials are valid , and only in that case give access to the area where cars positions are displayed.

   (c) n case the credentials are not valid, the system shall negate the access

3. **The system shall allow users to view the information about the history of rides and reservations they have made**

   (a) The user can access the information about its past reservations and rides clicking on a link from it's home page, after he has logged in

   (b) The system shall show on a list all past reservation and ride details of the user

4. **The system shall allow users to see the positions of all the available cars on a map**

   (a) If a user is currently on a ride or already reserved a car, he cannot access the map

   (b) The car are equipped with a GPS sensors that allow the system to localize them anytime.

   (c) The system puts all the cars positions on a map that can be visualized by users via web or mobile application on their home page after logging in

   (d) The user shall be able to insert a specific address or geographical position, in order for the system to show the available cars around the wanted location.

   (e) The system shall allow users to view all the positions of the cars which are strictly placed around users current position by simply tapping a button on the app. This is only possible if GPS sensor are working on the device the user is using to view the map.

5. **The system shall allow users to make a reservation for an available car**
    (a) When the user finds a suitable car he can reserve it by clicking on the "RESERVE" button on the application.
    (b) After doing the reservation, the system starts a timer of 60 minutes. If the user is not able to unlock the reserved car before the time expires, the system cancels the reservation making the car available again for other users.
6. **The system shall allow users who reserved a car to cancel the reservation**
    (a) Once a user reserved a car, the system shall allow him to cancel the reservation for it before the timer of 60 minutes expires by clicking on a button "CANCEL RESERVATION" on the mobile or web app.
    (b) If the user cancels a reservation, the system shall make the car available again for other users.
7. **The system shall allow users to unlock the car.**
    (a) Any reserved car can only be unlocked using the contactless card owned by the user that reserved that specific car, and before the timer of 60 minutes expires.
    (b) When the user places the contactless card near the sensor of the car, the system shall acquire information about the card, verify that it corresponds to the user that actually reserved that car and, in that case, the system shall unlock the car in order for the user to get in
    (c) In case the contactless card is not recognized as valid by the system, the system denies access to the car, until a valid card is recognized by the time the timer expires.
8. **The system shall allow users to communicate if the car they reserved is damaged or issues are found (e.g. dirt, damages, etc.).**
    (a) Before starting the engine the system shall remind the user through the on-board computer to check for damages on the car and to communicate it in case issues are found via the mobile app or the same on-board computer
    (b) In case of issues, the system shall cancel the reservation , make the car unavailable for users, marking it as "ASSISTANCE NEDEED" in order to inform all operators.
9. **The system shall allow users to start the ride if he doesn't find any issue on the car they want to ride**
    (a) In order to start the engine the user shall turn the key.

(b) The system shall start to charge the user by minute of ride as soon as he ignites the car engine.

(c) The system shall display on the onboard computer the updated amount of money spent for the ride.

10. **The system shall compute and display the shortest path from the user current position to a given location via the on-board computer**

    (a) After starting a ride, the user can insert the final location of his journy on the onboard computer

    (b) The system shall compute and display the shortest path from the user current location to the location insertd to the user

11. **The System shall allow users to make a pit stop of maximum 60 minute.**

    (a) The system shall allow users to make a pit stop of maximum 60 minutes communicating it tapping a button "START PIT STOP" on the onboard computer. When they do it, the system starts a timer of 60 minutes.

    (b) When the user wants to end the pit stop he needs to tap again a button "END PIT STOP" on the onboard computer, or simply restarting the car.

    (c) During the pit stop the system shall still go on charging users for 50% of the price during all the time the pit stop lasts or since the timer expires.

    (d) During the pit stop the system shall always make the car stay reserved for the current user, and unavailable to other users.

    (e) When the Timer expires the system cancels the reservation for that car, making it available for other users to use.

12.

13. **The system shall compute and display the shortest path from the user's current position to the nearest available special parking lot, if rqeuested by the user itself**

    (a) During a ride, the usr can tap on a button "find a park" on the onboard computer, in order to communicate to the system that they want to know where is the nearest special parking lot

    (b) The system shall compute the position of the nearest special parking lot from the user current position

    (c) The system shall compute the shortest path from the user's current position to the special car park found and display it on the on-board computer

14. **The system shall allow users to report an accident via the on-board computer**

    (a) Once the ride is started, the system shall display on the on-board computer a button "REPORT ACCIDENT"

    (b) In case the "REPORT ACCIDENT" button is pressed by a user, the system shall stop charging the user, forcing him to end the ride and inform operators marking the car with the "ACCIDENT" writing .

15. **The system shall allow users to end the ride**

    (a) When the user wants to end the ride, he has to communicate it to the system by tapping a button "END THE RIDE" on the onboard computer.

    (b) The system shall check that the GPS sensor information of the car is available, and only in that case allow user to end the ride.

    (c) The button "END THE RIDE" shall only be available to press if the car engine is shut down.

16. **The system shall compute the price of every ride taking into account discounts and penalties and send it to the system which take care of the payment.**

    (a) When a ride ends (if the users taps the "END THE RIDE " button or if the pit stop timer of 60 minutes expires) the system shall compute the total cost of the ride using the price currently displayed on the on-board computer and taking into account the following rules:

    - 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. In order to detect the presence of a passenger, the system shall check that the pressure sensors of the car seats are active both at the beginning and at the end of the ride.
    - At the end of the ride, the system shall check the battery charge: if a car is left with no more than 50% of the battery empty, the system applies a discount of 20% on the ride.
    - At the end of the ride, the system shall check if a car is left plugged into one of the power grids available through the charge sensor: in that case the system applies a discount of 30% on the last ride.
    - At the end of the ride the system shall check if a car is left at more than 3 KM from the nearest power grid station or with more than 80% of the battery empty, the system charges 30% more on the ride (in order to do that, the system shall compute the shortest path from the final position of the ride to all the

charging stations, find the minimum one and compare it with 3km).

17. **The system shall allow operators to log into a dedicated area .**
    (a) In order to log in, the system shall require operators to insert their credentials (consisting of an e-mail/username and a password) into an input form accessible from the web or mobile app.
    (b) The system shall verify that the credentials are valid , and in that case give access to the dedicated area.
    (c) In case the credentials are not valid, the system shall negate the access

18. **The system shall allow operators to view the updated status and position of every car**
    (a) Once they are logged in , the system shall allow operators to view (via web or the mobile app) the positions of all the cars on a list, associating to each car a status: "NO ISSUES" is for charged (or currently on charge) and working cars, "LOW CHARGE" is for cars under 20% battery charge, "ASSISTANCE NEEDED" is for cars that need some kind of maintenance or reparations, "ACCIDENT" in case a user is involved in a car accident while riding the car.
    (b) The system shall always keep all the positions and the status of the cars updated

19. The system shall allow operators to view the cars currently being assisted by other operators on a dedicated list accessible from the operator's reserved area
    (a) Once they are logged in, operators can access the information about the cars currently being repaired by other operators on a list clicking on a link inside their home page
    (b) The system shall redirect the operator to a page that contains a dynamic list of all the cars currently being repaired, retrieving the needed information from the database

20. **The system shall allow operators to be able to select a car from the list in order to take care of it.**
    (a) Operators shall be able to communicate to the system they are taking care of a car by selecting a row on the list and tapping a button "OPERATE".
    (b) Once an operator pressed the " OPERATE" button on a row of the list corresponding to a car, the system shall deny other operators to take care of the very same car, substituting the "OPERATE" button with a writing ? in progress..? (this is only valid for all the

operators except the one that is taking care of the car, to whom another kind of button is displayed by the system, as described below)

21. **The system allow an operator who took care of a car to communicate that the issues are resolved to all the other operators**

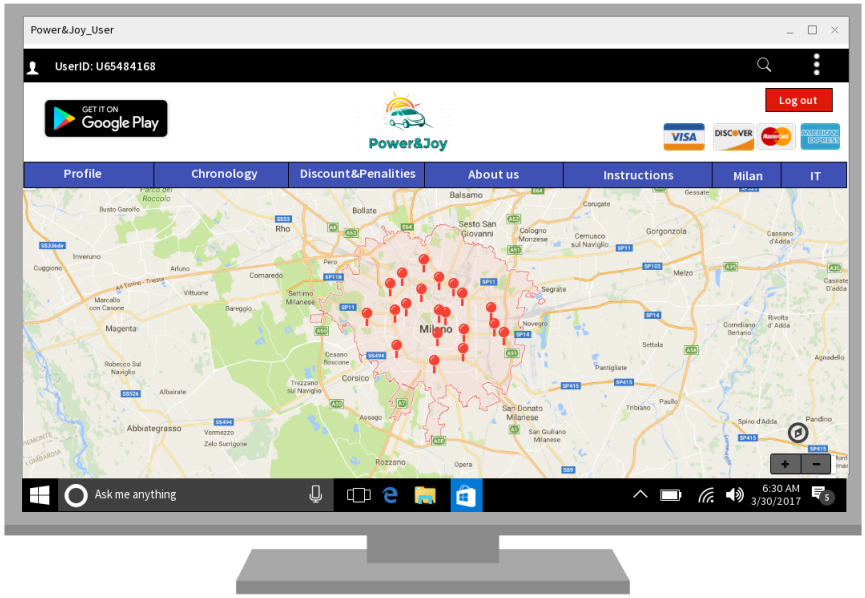
    (a) Once issues are resolved for a specific car , the specific operator who took care of it, shall be able to communicate it to the system by tapping a "ISSUES SOLVED?"button (only available to the operator who pressed the "OPERATE" button earlier). The system shall then mark the car as "NO ISSUES" again for all the operators.

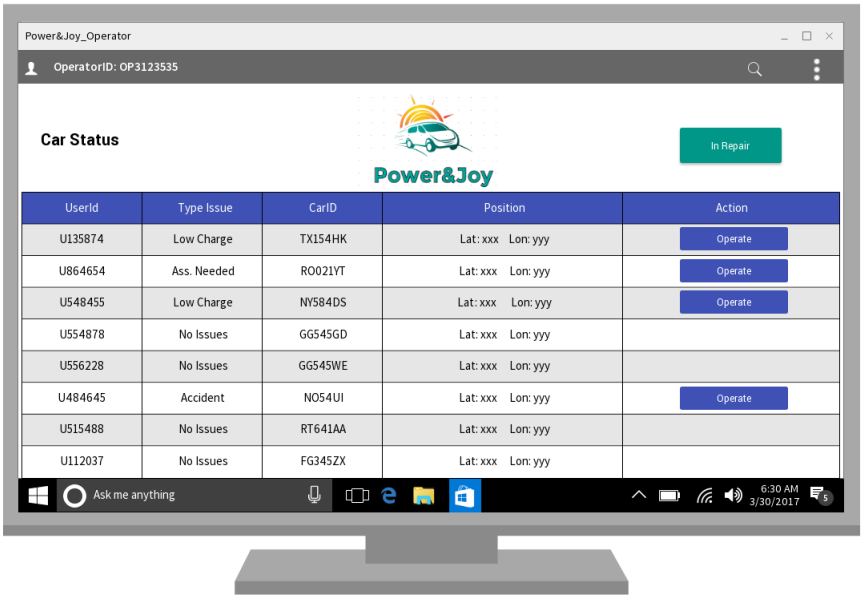
3.2 Non Functional Requirements

- The system must be available 24 hours a day
- The system must respect Quality of Service (QoS) attributes .
- The system shall start all the required timers within 1 second after the user?s input.
- The system shall unlock the car within 5 seconds after the contactless card is placed near the car sensor
- The system shall start charging the users within 2 seconds after they start the car engine at the beginning of a ride
- The system shall stop charging users within 2 seconds after they tap on the "END THE RIDE " button on the on-board computer, having the engine shut down, the car parked inside the safe area, and the car GPS signal available.
- The system shall change the status of every car within 2 seconds from one of the following events:
    1. A car switches from 20% to 19% battery charge
    2. An operator taps the "OPERATE" button associated to an issued car to take care of it
    3. An operator taps the "ISSUES SOLVED" button associated to a car they took care of
    4. A user taps on the button "ISSUES FOUND" on the on-board computer of a car
    5. A user taps on the "REPORT ACCIDENT" on the on-board computer

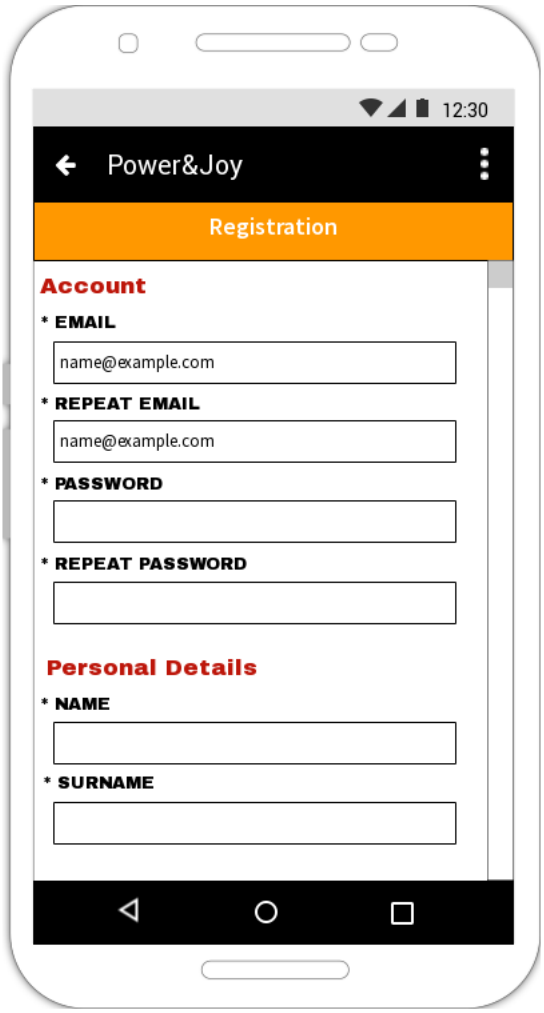
3.3 Mockups

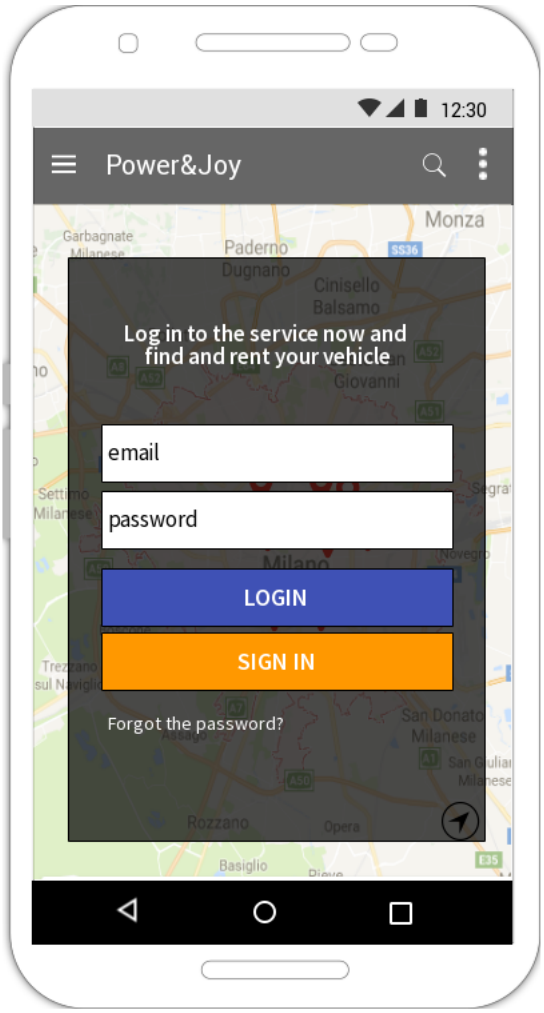
3.3.1 User's Home page / Operator's Home page



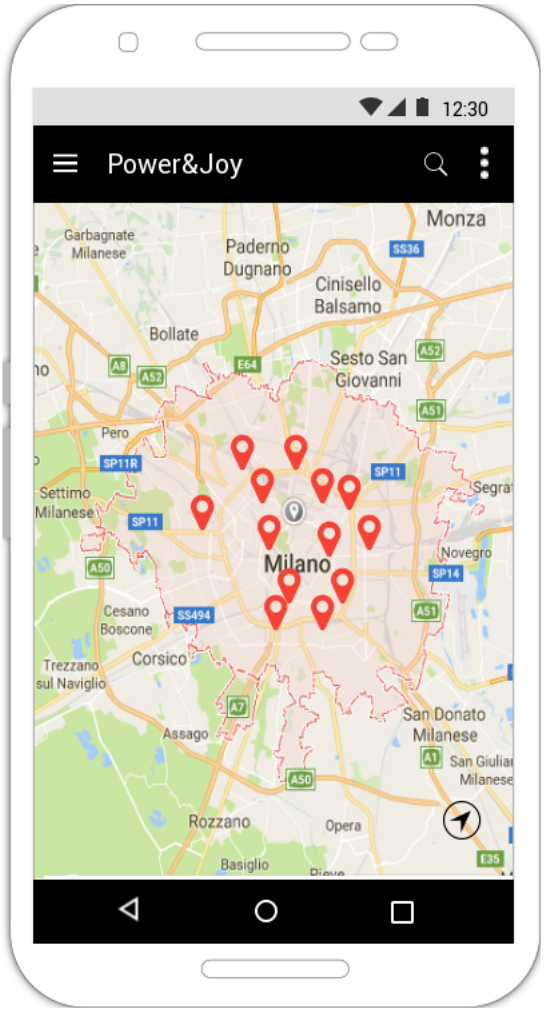

Power&Joy_Operator
OperatorID: OP3123535
Car Status
Power&Joy
In Repair
UserId
Type Issue
CarID
Position
Action
U135874
Low Charge
TX154HK
Lat: xxx Lon: yyy
Operate
U864654
Ass. Needed
RO021YT
Lat: xxx Lon: yyy
Operate
U548455
Low Charge
NY584DS
Lat: xxx Lon: yyy
Operate
U554878
No Issues
GG545GD
Lat: xxx Lon: yyy
U556228
No Issues
GG545WE
Lat: xxx Lon: yyy
U484645
Accident
NO54UI
Lat: xxx Lon: yyy
Operate
U515488
No Issues
RT641AA
Lat: xxx Lon: yyy
U112037
No Issues
FG345ZX
Lat: xxx Lon: yyy
Ask me anything
6:30 AM
3/30/2017

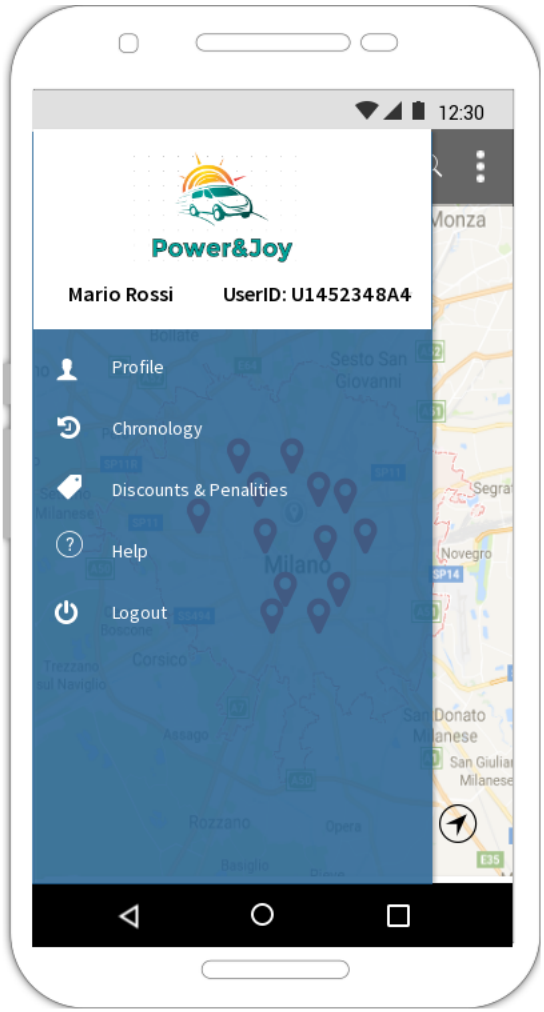
3.3.2 Mobile: User registration / User Log in



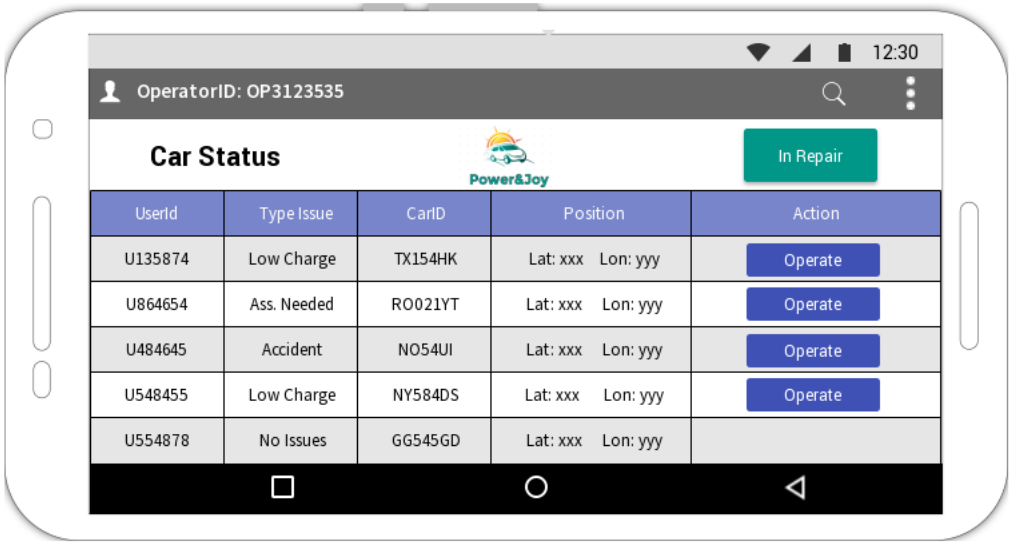

12:30
Power&Joy
Monza
Garbagnate Milanese
Paderno Dugnano
Cinisello Balsamo
Log in to the service now and find and rent your vehicle
email
password
LOGIN
SIGN IN
Forgot the password?
Milano
Rozzano
Opera
Basiglio

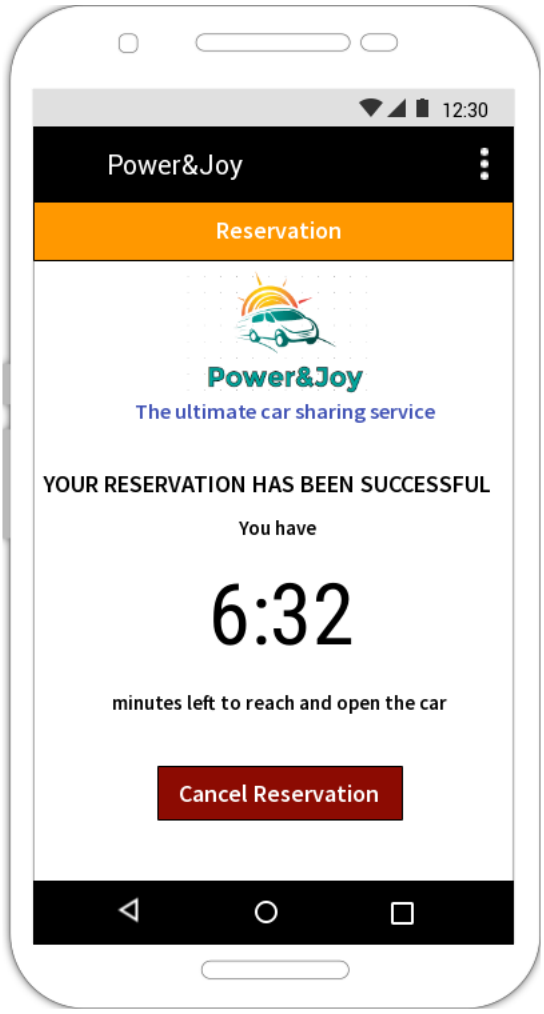
3.3.3 Mobile: User Home page



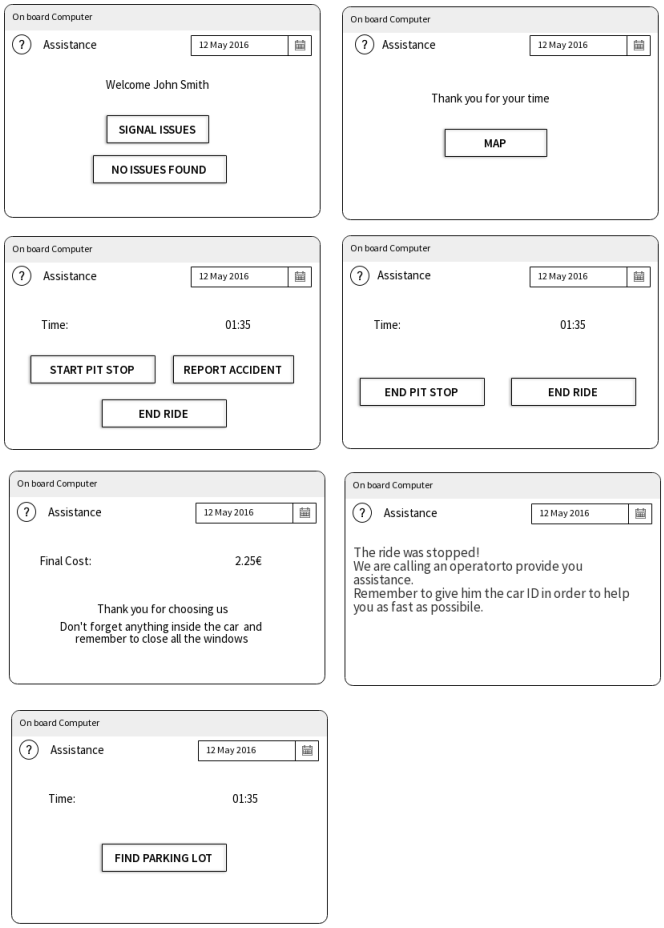

12:30
Power&Joy
Mario Rossi
UserID: U1452348A4
Profile
Chronology
Discounts & Penalities
Help
Logout
Monza
Sesto San Giovanni
Milano
Novegro
Corsico
Rozzano
San Donato Milanese

3.3.4 Mobile: Operators Home Page/ User Reservation page



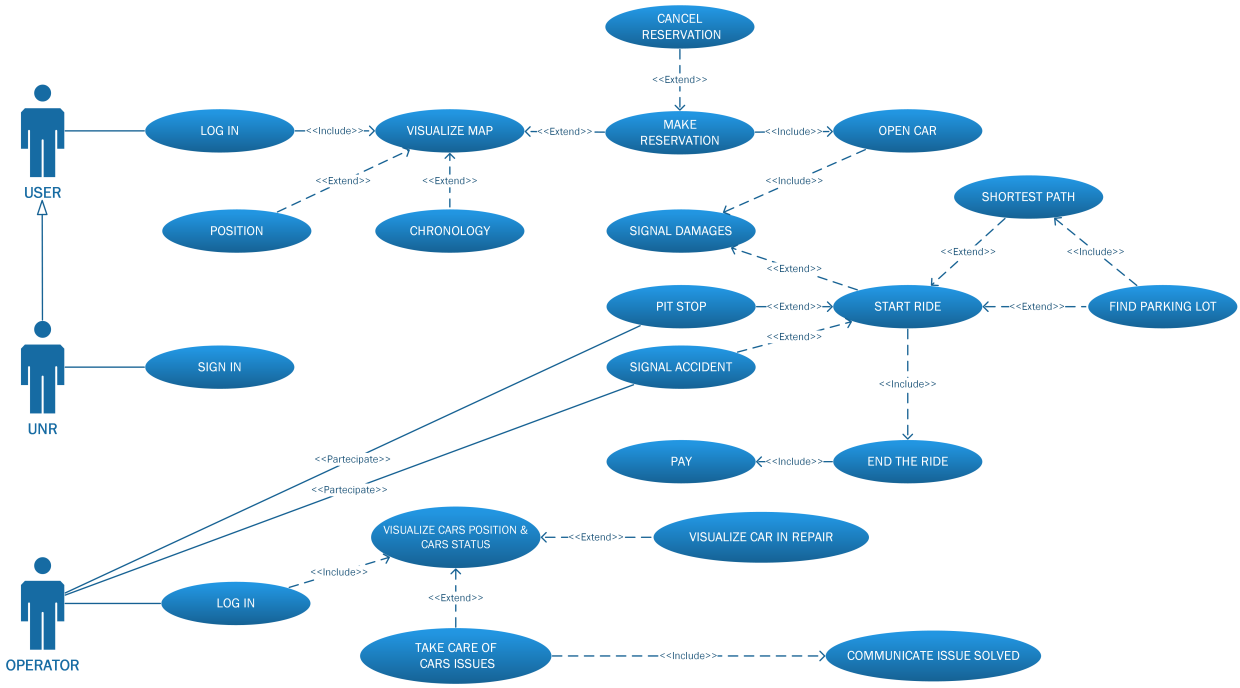

12:30
Power&Joy
Reservation
Power&Joy
The ultimate car sharing service
YOUR RESERVATION HAS BEEN SUCCESSFUL
You have
6:32
minutes left to reach and open the car
Cancel Reservation

3.3.5 On-Board Computer



4 UML Models

4.1 Use case Diagram



Use case Description
In this paragraph some use cases will be described. These use cases can be derived from the scenarios and the use case diagram.

Name	**Log in**
Description	A user logs into the system logs in
Actors	User
Entry Conditions	The user must be registered into the system
Flow of events	• A user opens the user's log in page in the web or mobile app • The user inputs his credentials, consisting of a username and a password • The user clicks on the "LOG IN" button • The system checks that the credentials are valid (they are associated to an existing user) and redirects the user to the User's Home page
Exit conditions	The user is redirected to the User?s home page.
Exceptions	If the username or password inserted by the user are not valid, the system does not redirect the user to the user?s home page. Instead, it notifies him that the credentials he inserted are not valid and tells him to try again

Name	**User visualizes a map of the cars**
Description	A user visualizes a geographical map where are enlighten the current positions of all available cars
Actors	User
Entry Conditions	The user must have correctly performed the log in
Flow of events	• The user can visualize from the user's home page, directly accessible from the login page, a map of Milan's urban area, where are enlighten the current geographical positions of all the available cars
Exit conditions	The user is now aware of the positions of all the available cars (i.e. cars which can be reserved)
Exceptions	if the user has an on-going reservation or ride, the map is not shown. The reservation or ride page is shown instead

Name	**User visualizes a list of his past reservations and rides**
Description	A user visualizes the history of his reservations and rdes
Actors	User
Entry Conditions	The user must have correctly performed the log in
Flow of events	• The user can access, clicking on a link from the user's home page, a page where a list of all his previous reservations and rides is shown
Exit conditions	The user can access all information about his past reservations and rides
Exceptions	none

Name	**Location of cars whiten 1 km from a given position**
Description	A user inserts specific location and visualize all near available cars
Actors	User
Entry Conditions	The user must be logged in
Flow of events	• The user inputs a geographical location (consisting of a street/postal code/building etc.) into an input area at the bottom of the user's home page • The user clicks on the "SUBMIT" button • The system dynamically changes the geographical area shown on the map, in order to highlight the cars within 1 km from the location inserted by the user •
Exit conditions	The user can now see highlighted on the map the cars near the location he inserted
Exceptions	if the location is not recognized, the map does not change, and an error message is shown.

Name	**Car reservation**
Description	The user reserves a car
Actors	User
Entry Conditions	The user can now see highlighted on the map the cars near the location he inserted, or the whole list of cars, in case he didn't insert any location
Flow of events	• The user select a car on the map, among the available cars shown. • The system shows a "RESERVE CAR" button near the selected car • The user clicks on the "RESERVE CAR" button • The system reserves a car for the user, making it unavailable to others, and redirects him to the reservation page, where he can visualize a message confirming that the reservation has been successful and a timer that shows the remaining minutes before the reservation expires, the position of the car etc
Exit conditions	The user is redirected to the reservation page, where he can visualize the details of his reservation.
Exceptions	In case problems occur on the reservation process, an error message is shown, and the car remains available for other reservations

Name	**Reservation cancellation**
Description	A user cancels a reservation he made
Actors	User
Entry Conditions	The user must be logged into the system, and must have successfully reserved a car
Flow of events	• From the reservation page, the user is able to visualize a "CANCEL RESERVATION" button • The user clicks on the"CANCEL RESERVATION" button • The system cancels the reservation, making the car available to other users • The system shows a message to inform that the reservation has been correctly cancelled • The system redirects the user to the user's home page
Exit conditions	The user is redirected to the User's home page
Exceptions	None

Name	**Open car**
Description	A user opens the car he reserved
Actors	User
Entry Conditions	The user must have logged into the system and have successfully reserved a car. The user has reached the car he reserved and has the contactless card needed to unlock the car with him
Flow of events	• The user places his contactless card near the car sensor before the timer expires • The system retrieves the information about the card and verifies that it corresponds to the user that reserved that car • The system unlocks the car
Exit conditions	The user has successfully unlocked the car
Exceptions	• If the timer has expired by the time the user places the contactless card near the car sensor, the system denies access to the car, notifying the user with a message on the mobile app and with a red light appearing on the car • If the contactless the user uses to try unlock the car is not belonging to him, the system denies access to the car and notifies the user with a red light appearing on the car • If the car the user tries to unlock is not the one he reserved, the system denies access to the car and notifies the user with a red light appearing on the car

Name	**Car issue signaling**
Description	A user communicate to the system weather there are issues on the car he reserved or not
Actors	User
Entry Conditions	The user has successfully reserved and unlocked a car
Flow of events	• The system reminds the user to check for issues on the car with a message on the onboard computer before letting the user start the ride • The user checks if there are issues and do not find any • The user taps on a button " ISSUES FOUND" on the on-board computer • The system cancels the reservation, making the car not available to the current user as well as all the others. The system tells the user to leave the car with a message on the on-board computer. The user is unable to start the ride and leaves the car. The system informs operators about the issue by marking the car as "ASSISTANCE NEEDED".
Exit conditions	The user has correctly signald an issue in the car he wanted to ride to the system
Exceptions	If the user finds no issues on the car, he communicate it to the system tapping a "NO ISSUES FOUND" button on the on-board computer, and normally start a ride

Name	**Start the ride**
Description	A user starts a ride with the car he reserved and unlocked
Actors	User
Entry Conditions	The user has successfully reserved and unlocked a car, found no issues on that car and communicated it to the system.
Flow of events	• The user turns the car key and starts the engine • The user starts to be charged by the system • The user starts to be able to visualize the updated amount of money spent for the ride on the on-board computer
Exit conditions	The user successfully started the ride
Exceptions	None

Name	**Path to destination**
Description	The user inserts a location and the system shows the shortest path to reach it
Actors	User
Entry Conditions	The user has successfully started a ride
Flow of events	• The user taps a button "INSERT LOCATION" on the on-board compiuter • The user inserts a location • The system computes the shortest path from the car's current location to the destination location inserted by the user and displays it on the on-board computer
Exit conditions	The user successfully visualized the path from his current position to the desired location
Exceptions	If the system does not recognise the inserted location, an error is shown and the user can try again

Name	**Pit stop**
Description	The user makes a pit stop
Actors	User
Entry Conditions	The user has correctly started a ride on a car he reserved and unlocked
Flow of events	• The user parks the car and stops the engine turning the key • The user taps on the "START PIT STOP" button on the on-board computer • The user starts to be charged by the system for half the price of a normal ride • The user can now leave the car parked for maximum 60 minutes, keeping it reserved for future use • The user gets back to the car, unlocks the car placing his contactless card near the car sensor • The user taps on the "END PIT STOP" button placed on the on-board computer • The system restarts to charge the user for the full price as soon as the user taps on the " END PIT STOP" button , and the user is able to restart the engine and continue to ride the car
Exit conditions	The user has successfully made a pit stop
Exceptions	In case the timer of 60 minutes expires by the time the user unlocks the car again in order to end the pit stop, the reservation for the car is cancelled, the system stops to charge the current user and makes the car available for other users.

Name	**Path to nearest special parking lot**
Description	The system shows the shortest path to reach the nearest available special parking lot
Actors	User
Entry Conditions	The user has successfully started a ride
Flow of events	• The user taps a button "FIND PARK" on the on-board compiuter • The system computes the location of the nearest available special parking lot • The system computes the shortest path from the car's current location to the destination location inserted by the user and displays it on the on-board computer.
Exit conditions	The user successfully visualized the path from his current position to the desired special parking lot
Exceptions	none

Name	**End the ride**
Description	The user ends a ride
Actors	User
Entry Conditions	The user must have successfully started a ride
Flow of events	• The user parks the car and stops the engine turning the key • The user taps on the "END THE RIDE" button on the on-board computer • The system stops charging the user • The system reminds the user to close the windows and to leave the car in the same conditions in which he found it at the beginning of the ride with a message on the on-board computer
Exit conditions	The user has successfully ended a ride
Exceptions	None

Name	**Accident**
Description	A user reports an accident he is involved in while riding the car
Actors	User
Entry Conditions	The user must have successfully started a ride
Flow of events	• The user taps on the "REPORT ACCIDENT" button on the on-board computer, which is available to press once the ride is started • The system marks the car as ?accident? and at the same time makes a call start from the on-board computer to allow the user to talk to an operator in order to provide the details of the accident and receive instructions. • The operator that has been called taps on the "OPERATE" button on the row of the list associated to that car and provides assistance to the user. • Once the issue is resolved , the operator taps on the "ISSUE SOLVED" on the same row of the list. • The system marks back the car as "NO ISSUE"
Exit conditions	The user has successfully reported an accident to the operator, and the operator provided assistance.
Exceptions	None

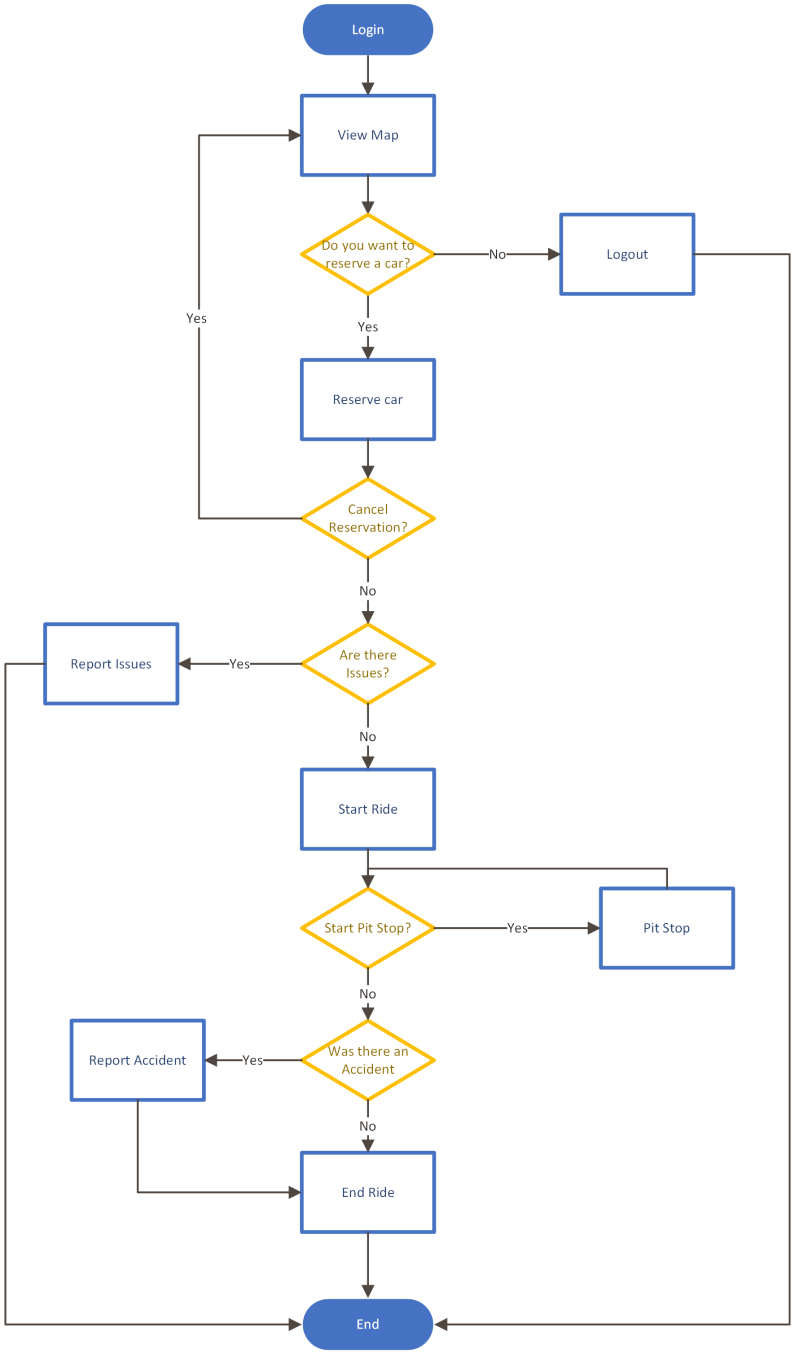
Name	**Operator log in**
Description	An operator logs into the system
Actors	Operator
Entry Conditions	None
Flow of events	• An operator opens the operator?s log in page on the web or mobile app • The operator inputs his credentials, consisting of a username and a password • The operator clicks on the "LOG IN" button • The system checks that the credentials are valid (they are associated to an existing operator) and redirects the operator to the Operator's Home page
Exit conditions	The operator has successfully logged into the section of the system dedicated to operators
Exceptions	If the credentials inserted by the operator are not valid, the system shows an error message, inviting to try again and doesn't redirect the operator anywhere

Name	**Operator views cars information**
Description	An operator visualizes the status and position of all the cars on a list
Actors	Operator
Entry Conditions	The operator must be logged into the system
Flow of events	• The operator visualizes from the Operator?s home page, directly accessible from the login page, a table: each row of the table is associated to a car, and it features the position (expressed in terms of latitude and longitude, the status (consisting of a writing among "NO ISSUES", "ASSISTANCE NEEDED", "LOW CHARGE" and "ACCIDENT"), the name of the user who is using the car and the ID of the car. The list also shows if other operators already decided to take care of a car by showing a writing "in progress" or a button "OPERATE " on the rows associated to issued cars.
Exit conditions	The operator has successfully visualized the information about all the cars.
Exceptions	None

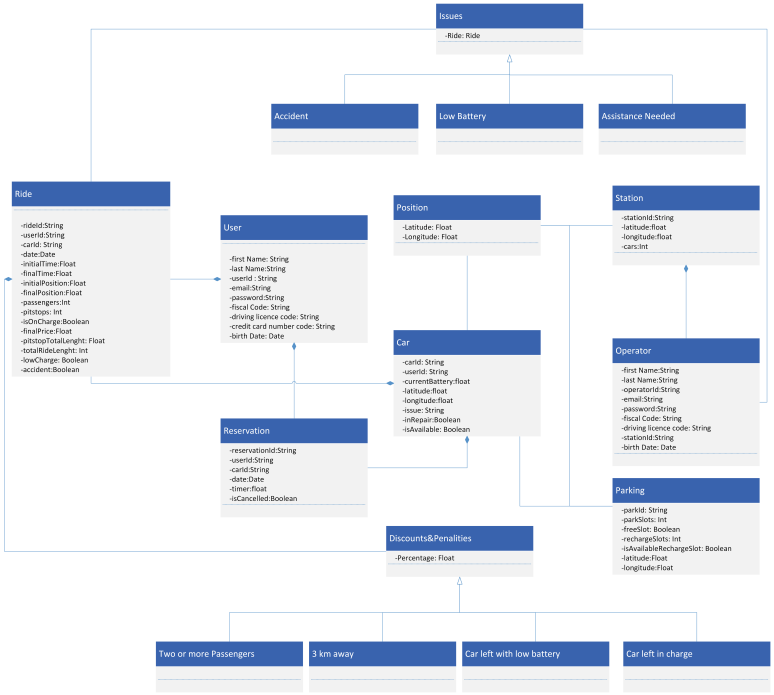
Name	**Oprator takes care of a car**
Description	An operator takes care of an issued car that appears on the list
Actors	Operator
Entry Conditions	The operator must be logged into the system and have visualized the list of all cars
Flow of events	• An operator taps on the "OPERATE " button related to a car which status is "ASSISTANCE NEEDED", "LOW CHARGE" or "ACCIDENT" • After he took care of the issue, the operator taps on the button "ISSUE SOLVED" on the same row of the list, to communicate that the car is now back to its regular functions. • The system marks back the car as "NO ISSUE"
Exit conditions	The operator has successfully solved an issue of a car.
Exceptions	None

Name	**Operator views information about cars on repair**
Description	An operator visualizes the list of cars that are being taken care of by other operators
Actors	Operator
Entry Conditions	The operator must be logged into the system
Flow of events	• The operator clicks on a link "cars on repair" from his home page , directly accessible from the login page, and is redirected to a page that shows table: each row of the table is associated to a car, and it features the position (expressed in terms of latitude and longitude, the status (consisting of a writing among ', "ASSISTANCE NEEDED", "LOW CHARGE" and "ACCIDENT"), the name of the user who is using the car, the ID of the car and the ID of the operator that is taking care of that car.
Exit conditions	The operator has successfully visualized the list of all issued cars that are being repaired by some other operator.
Exceptions	None

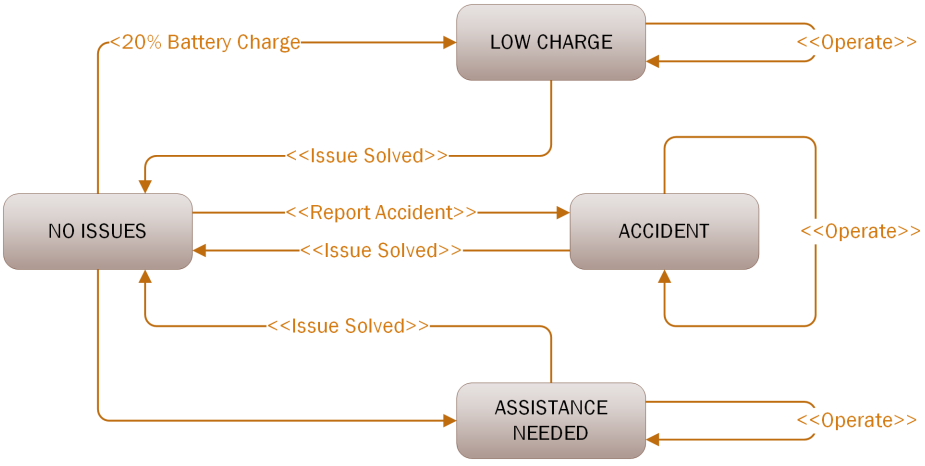
4.2 Activity Diagrams



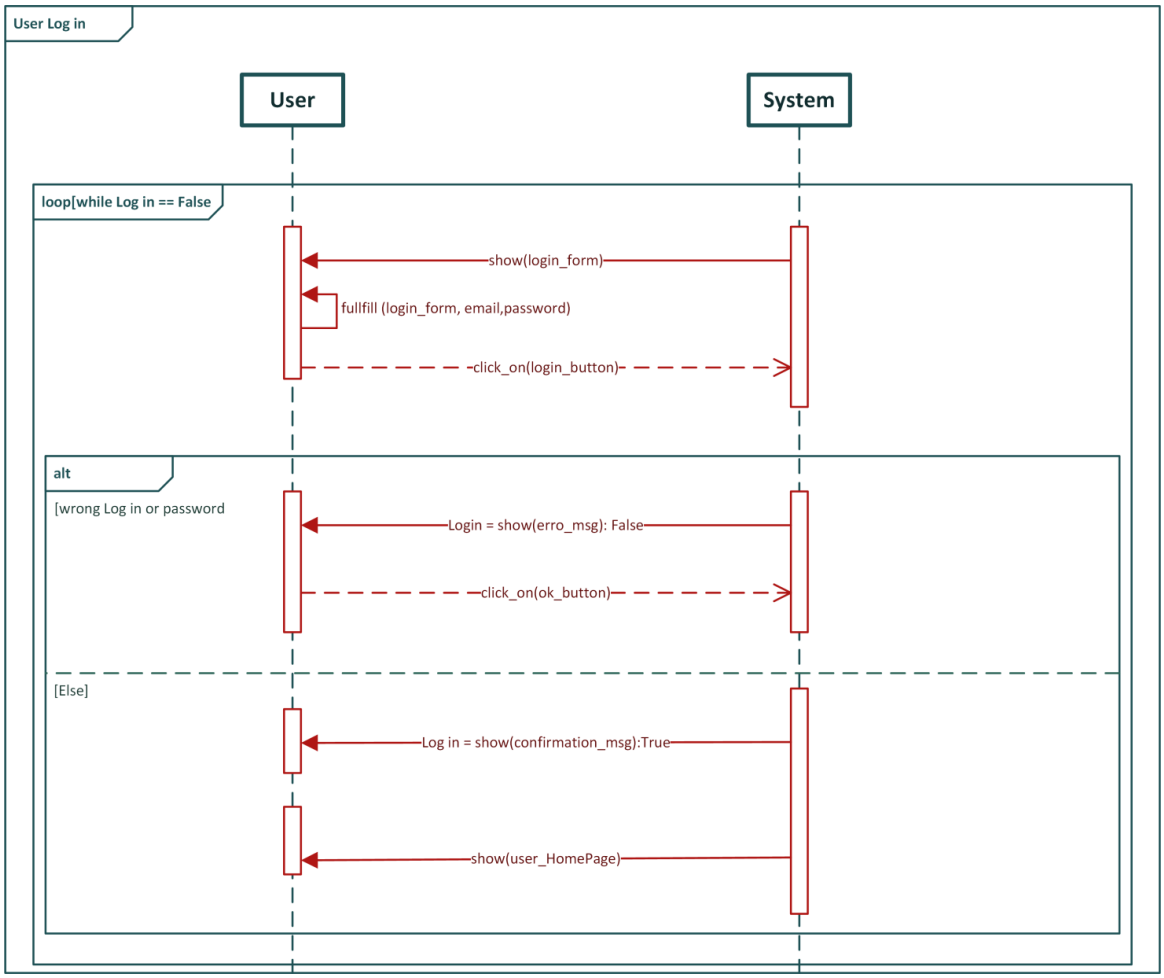
4.3 Class Diagrams

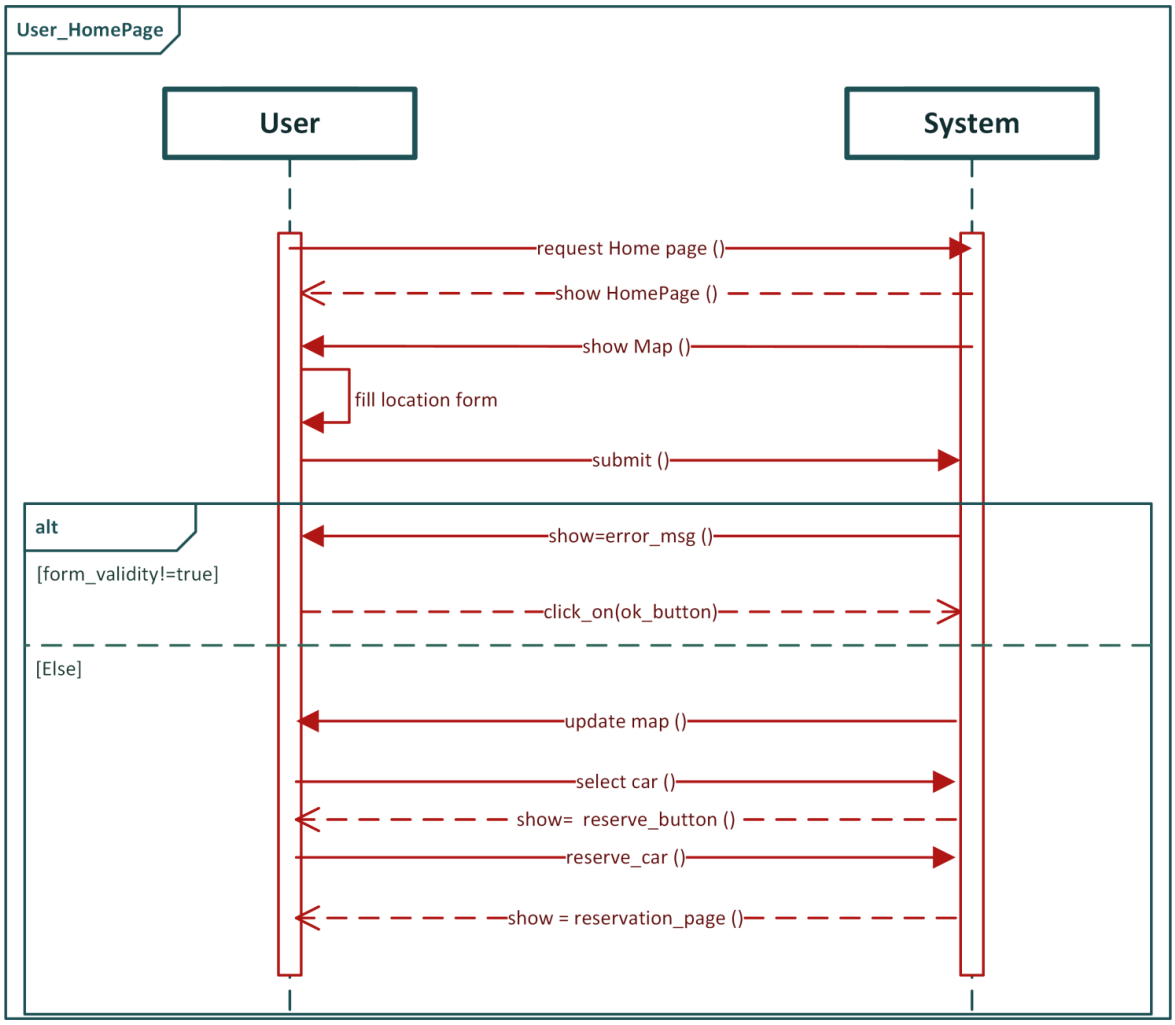


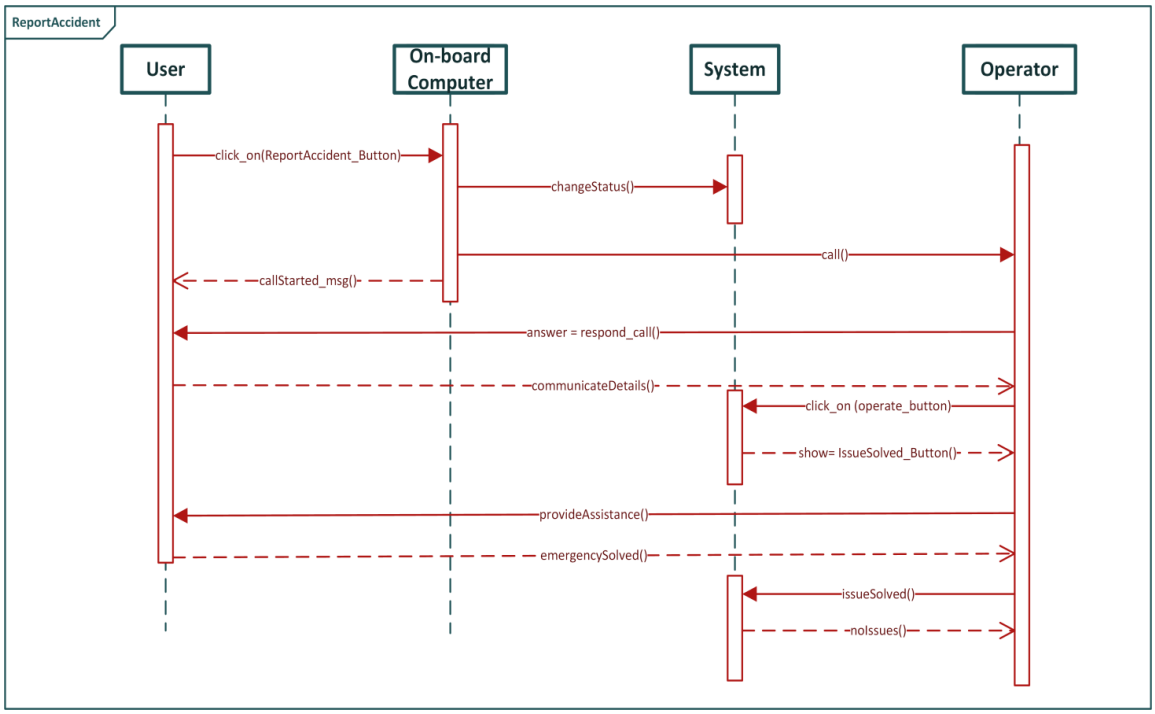
4.4 State Diagrams



4.5 Sequence Diagrams




User_HomePage
User
System
request Home page ()
show HomePage ()
show Map ()
fill location form
submit ()
alt
[form_validity!=true]
show=error_msg ()
click_on(ok_button)
[Else]
update map ()
select car ()
show= reserve_button ()
reserve_car ()
show = reservation_page ()


ReportAccident
User
On-board Computer
System
Operator
click_on(ReportAccident_Button)
changeStatus()
call()
callStarted_msg()
answer = respond_call()
communicateDetails()
click_on (operate_button)
show= IssueSolved_Button()
provideAssistance()
emergencySolved()
issueSolved()
noIssues()

5 Alloy

5.1 Model

```
open util/boolean

sig Email{}

sig User {
identifyby:Email,
}

fact UniqueEmail{
User.identifyby=Email
}

fact OneEmailperUser {
all u1,u2 : User | u1!=u2 => u1.identifyby != u2.identifyby
}

sig Car {
plate: Int,
position : CarPosition
}{plate>0}

fact OnlyOnePlatePerCar{
all c1,c2 :Car | c1!=c2 => c1.plate != c2.plate
}

fact UniqueCarPosition {
Car.position = CarPosition
}
fact OnlyOnePositionPerCar{
all c1,c2 :Car | c1!=c2 => c1.position != c2.position
}


sig Reservation {
reservedby: one User,
car: Car,
issue : lone ReservationIssue
}

fact UniqueRes {all r1,r2 :Reservation |
r1!=r2 => r1.reservedby!=r2.reservedby}
fact UniqueCar{all r1,r2 :Reservation | r1!=r2 => r1.car!=r2.car }
```

```
fact AllReservationIssuesAreAssociatedToAReservation
{ Reservation.issue = ReservationIssue}
fact NoReservationsAssociatedToTheSameIssue{ all r1, r2 : Reservation
| r1!=r2 => r1.issue!=r2.issue}

sig Ride {
driveby:  one User,
car:Car,
passengers :Int ,
issue : set RideIssue ,
discountsPenalities : set DiscountPenalities
}{
passengers<5
passengers>1
#issue=1 or #issue =0
}

fact LowChargeIssueImpliesLowChargePenality  {
no r: Ride |  all i : LowCharge | all dp: CarWithLowCharge |
 (dp in r.discountsPenalities && !(i in r.issue)) ||
(i in r.issue && !(dp in r.discountsPenalities))
 }
fact  LowChargeIssueImpliesNoPluggedInCar {
no r:Ride | some dp: CarWithLowCharge | some dp2: CarPluggedIn |
 (dp in r.discountsPenalities) && (dp2 in r.discountsPenalities)
 }
fact LowChargeIssueImpliesNoChargedCar {
 no r: Ride | some dp: CarWithLowCharge | some dp2 : ChargedCar |
 (dp in r.discountsPenalities) && (dp2 in r.discountsPenalities)
 }
fact morePassengerDiscountImpliesAtLeastThreePassengers {
all r:Ride | all dp: MorePassengers  |
(dp in r.discountsPenalities) && r.passengers>2
}
fact d1 {
all r: Ride | all dp1: CarPluggedIn | all dp2:  CarWithLowCharge |
dp1 in r.discountsPenalities <=>!(dp2 in r.discountsPenalities)
 }

fact AllDiscountsHaveARide {
Ride.discountsPenalities=DiscountPenalities
}

fact d2 {
all r:Ride ,dp1 , dp2 : ThreeKmAway |
```

```
dp1!=dp2 && dp1 in r.discountsPenalities <=>!(dp2 in r.discountsPenalities)
}
fact d2 {
all r:Ride, dp1, dp2 : CarWithLowCharge |
dp1!=dp2 &&  dp1 in r.discountsPenalities <=>!(dp2 in r.discountsPenalities)
}
fact d2 {
all r:Ride, dp1, dp2 : CarPluggedIn |
 dp1!=dp2 && dp1 in r.discountsPenalities <=>!(dp2 in r.discountsPenalities)
}
fact d2 {
all r:Ride,  dp1, dp2 : ChargedCar |
dp1!=dp2 && dp1 in r.discountsPenalities <=>!(dp2 in r.discountsPenalities)
}
fact d2 {
all r:Ride,  dp1, dp2 : MorePassengers |
(dp1!=dp2 && dp1 in r.discountsPenalities) <=>!(dp2 in r.discountsPenalities)
}
fact AllRideIssuesAreAssociatedToARide {
Ride.issue = RideIssue
}
fact NoRidesAssociatedToTheSameIssue{
 all r1, r2 : Ride | r1!=r2 => r1.issue!=r2.issue
}
fact UniqueRide {
all r1,r2 :Ride | r1!=r2 => r1.driveby!=r2.driveby
}
fact UniqueRideCar {
all r1,r2 :Ride | r1!=r2 => r1.car!=r2.car
}
fact CarRideOrRes1 {
Ride.car = Ride.car - Reservation.car
}
fact CarRideOrRes2 {
Reservation.car= Reservation.car- Ride.car
}
fact ResOrRide {
Ride.driveby = Ride.driveby - Reservation.reservedby
}


fact ResOrRide1 {
Reservation.reservedby= Reservation.reservedby- Ride.driveby
 }

sig CardID{}
```

```
sig Operator{
id: CardID,
station :  one ChargingStation,
issueToTakeCareOf : lone Issues
}

fact everyIssueHasAnOperator {
Operator.issueToTakeCareOf = Issues
}
fact DifferentOperatorsAssociatedToDifferentIssues {
 all o1, o2: Operator | o1!=o2 => o1.issueToTakeCareOf !=o2.issueToTakeCareOf
}
fact OneIDperOperator {
all o1,o2 : Operator | o1!=o2 => o1.id != o2.id
}
fact UniqueCardID{
Operator.id=CardID
}

sig ChargingStation {
position: StationPosition,
}

fact UniqueWorker {
Operator.station=ChargingStation
}
fact OnePositionPerStation {
all cg1, cg2 : ChargingStation | cg1!= cg2 => cg1.position != cg2. position
}
fact UniquePosition{
ChargingStation.position = StationPosition
}




abstract sig DiscountPenalities{}

abstract sig Discounts extends DiscountPenalities{}
abstract sig Penalities extends DiscountPenalities{}

sig MorePassengers extends Discounts{}
sig CarPluggedIn extends Discounts{}
sig  ChargedCar extends Discounts{}
```

```
sig  ThreeKmAway extends Penalities{}
sig  CarWithLowCharge extends Penalities{}

abstract sig Issues {}

abstract sig RideIssue extends Issues {}
abstract sig ReservationIssue extends Issues{}

fact ItsAlwaysIssuesWeAreTalkingAbout {
 RideIssue + ReservationIssue = Issues
}


sig Accident extends RideIssue {}
sig AssistanceNeeded extends ReservationIssue {}
sig LowCharge extends RideIssue {}

fact AllRideIssues{
Accident + LowCharge = RideIssue
}

abstract sig Position{}
sig StationPosition extends Position {}
sig CarPosition extends Position {}

pred show {
#Ride = 1
#Reservation = 1
#User = 3
#ChargingStation = 1
#Operator = 2
}


assert everyOperatorDoesHisJob {
all r:Ride |all  i : RideIssue | some op: Operator |
( i in r.issue) =>i in op.issueToTakeCareOf
 }
assert everyOperatorDoesHisJob2 {
all res: Reservation | all i: ReservationIssue |
some op: Operator| (i in res.issue ) =>i in op.issueToTakeCareOf
}
assert everyRideLowChargeIssueHasALowChargePenality {
all r:Ride | some i: LowCharge | some dp: CarWithLowCharge |
i in r.issue <=> dp in r.discountsPenalities
}
```

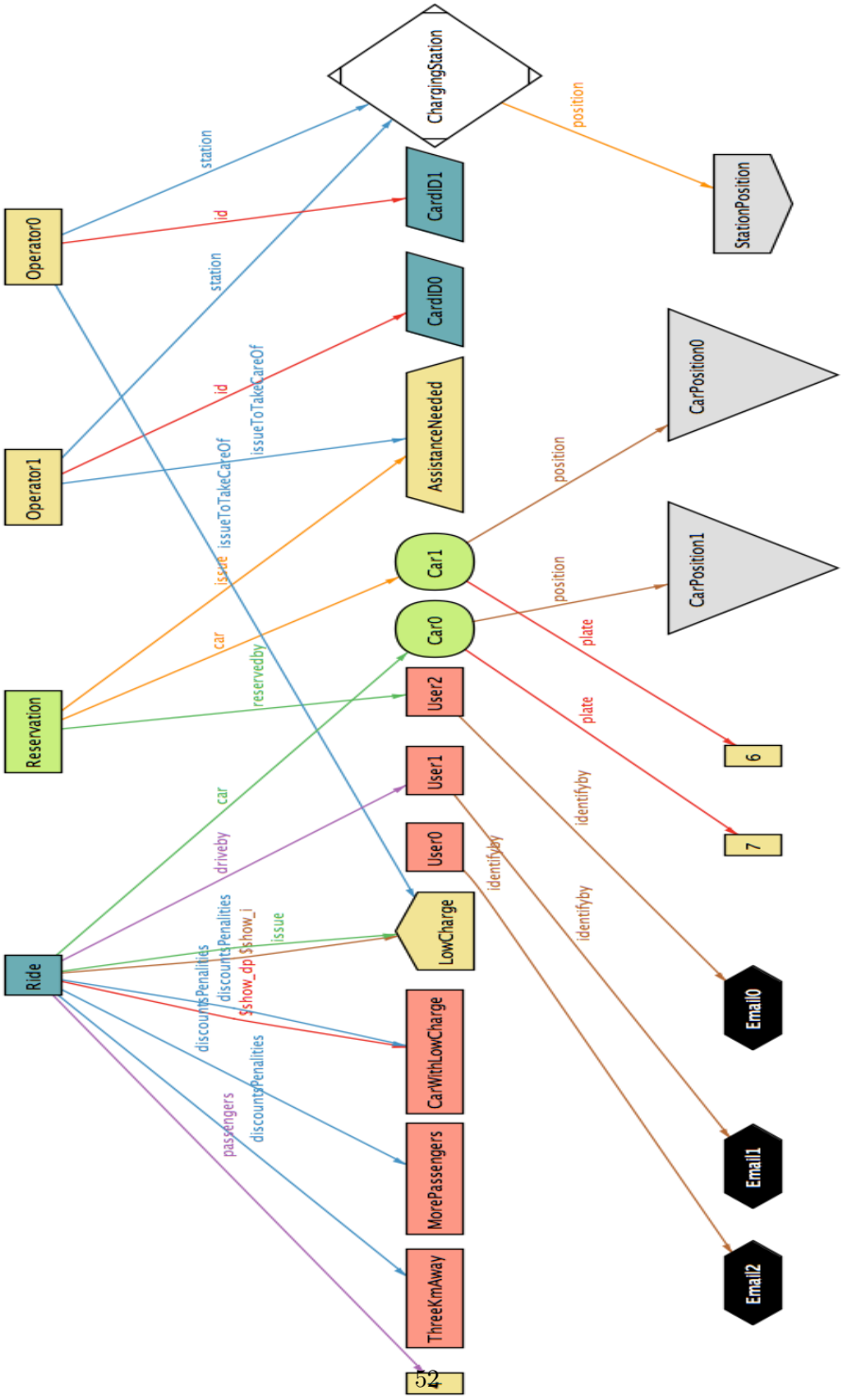
```
assert noLowChargeIssueOnPluggedInCars {
no r:Ride |some dp:CarWithLowCharge  | some i: LowCharge
|
some dp2 : CarPluggedIn |
(i in r.issue || dp in r.discountsPenalities) && (dp2 in r.discountsPenalities)
}
assert noLowChargeIssueChargedInCars {
no r:Ride |some dp:CarWithLowCharge  | some i: LowCharge
|
some dp2 : ChargedCar |
(i in r.issue || dp in r.discountsPenalities) && (dp2 in r.discountsPenalities)
}

check noLowChargeIssueChargedInCars
check noLowChargeIssueOnPluggedInCars
check everyOperatorDoesHisJob
check everyOperatorDoesHisJob2
check everyRideLowChargeIssueHasALowChargePenality
run show for 5
```

Alloy Result

6 commands were executed. The results are:

#1: No counterexample found. noLowChargeIssueChargedInCars may be valid.
#2: No counterexample found. noLowChargeIssueOnPluggedInCars may be valid.
#3: No counterexample found. everyOperatorDoesHisJob may be valid.
#4: No counterexample found. everyOperatorDoesHisJob2 may be valid.
#5: No counterexample found. everyRideLowChargeIssueHasALowChargePenality may be valid.
#6: **Instance found.** show is consistent.

5.2 World Generated



6 Other Features

6.1 Future Development

We are thinking about extending the perimeter of the safe area to the whole region, and possibly applying it to other regions. We also plan to implement a more innovative and computationally efficient method to let the system check whether a car is parked within 3 km of distance from the nearest safe area , using a special signal created by emitters placed into charging stations instead of using the GPS sensor.

6.2 Hours of work

Joshua Nicolay Ortiz Osorio

- 26/10/2016 : 1h30m
- 28/10/2016 : 2h30m
- 29/10/2016 : 1h
- 1/11/2016 : 2h
- 4/11/2016 : 2h
- 5/11/2016 : 1h
- 8/11/2016 : 3h
- 9/11/2016 : 3h
- 10/112016 : 3h
- 11/11/2016 :2h30m
- 12/11/2016 : 4h
- 13/11/2016 : 4h

Michelangelo Medori

- 26/10/2016 : 1h30m
- 27/10/2016 : 2h
- 29/10/2016 : 2h30m
- 1/11/2016 : 2h
- 4/11/2016 : 2h
- 5/11/2016 : 1h

- 8/11/2016 : 3h
- 10/112016 : 3h
- 11/11/2016 :2h30m
- 12/11/2016 : 4h
- 13/11/2016 : 4h

6.3 Used Tools

- Visio: for Use Case Diagram, Sequence Diagrams, State Diagram and Activity Diagram
- Balsamiq : to make mock-ups
- Latex : to write the document and create .pdf
- Github: for version controller
- Alloy Analizer 4.2 . to prove the consistency of our model
- Dropbox : to share documents online

6.4 Revision History

Version	**Date**	**Author(s)**	**Summary**
1.0	11/11/2016	Joshua Nicolay Ortiz Osorio and Michelangelo Medori	initial release
2.0	20/12/2016	Joshua Nicolay Ortiz Osorio and Michelangelo Medori	minor fixes
3.0	05/02/2017	Joshua Nicolay Ortiz Osorio and Michelangelo Medori	more txt assumption, fixed requirements and class diagram
4.0	06/02/2017	Joshua Nicolay Ortiz Osorio and Michelangelo Medori	added new requirements, fixed text assumptions and use case
5.0	07/02/2017	Joshua Nicolay Ortiz Osorio and Michelangelo Medori	new mockups added, fixed UX and BCE diagrams, added more text assumption