



# POWER ENJOY

## *RASD*

*Requirements Analysis and Specification Document*

---

Version: 1.0.2

13th November 2016

*Edited by:*

Marzia Degiorgi - 878360

Giulia Leonardi - 877491

Valentina Ionata - 808678

# Index

<b>Index</b>	<b>2</b>
<b>1. Introduction</b>	<b>5</b>
1.1 Purpose	5
1.1.1 Goals	6
1.2 Scope	7
1.3 Definitions, Acronyms and Abbreviation	7
1.3.1 Definitions	7
1.3.2 <b>Acronyms</b>	9
1.3.3 Abbreviation	9
1.4 References	9
1.5 Overview	10
<b>2. Overall Description</b>	<b>11</b>
2.1 Product perspective	11
2.1.1 System Interfaces	11
2.1.2 User Interfaces	11
2.1.2.1 Users	11
2.1.2.1.1 Mobile interface	11
2.1.2.1.2 On-board interface	14
User Interface 9: Stop Or End Of The Ride Layout	16
2.1.2.2 Operators	16
2.1.3 Hardware Interfaces	17
2.1.4 Software Interfaces & Operating System	17
2.1.5 Operations	17
2.1.6 Site Adaption Requirement	18
2.2 Product functions	18
2.3 User characteristic	22
2.4 Constraints	22
2.4.1 Regulatory policies	22
2.4.2 Hardware limitation	22
2.4.3 Interfaces to other applications	23
2.4.4 Parallel operation	23
2.5 Assumptions and Dependencies	23
<b>3. Specific Requirements</b>	<b>24</b>
3.1 External Interface Requirements	24

3.1.1 User Interface	24
3.1.1.1 Users	24
3.1.1.1.1 Mobile interface	24
3.1.1.2 Operators	25
3.2 Functional Requirement	25
3.2.1 Scenarios	25
3.2.1.1 Sign up in the System	25
3.2.1.2 Log in and Reserve a car	26
3.2.1.3 Pick up the car	27
3.2.1.4 End of a ride	28
3.2.1.5 Use Discounts services	29
3.2.1.6 Bad behaviour and recharge	30
3.2.1.7 Delete a reservation and change profile information	31
3.2.2 Activity diagram	32
3.2.3 Use Case Model	33
3.2.3.1 Use Case : Guest	34
3.2.3.1.1 Register in the system	34
3.2.3.2 Use Case : User	37
3.2.3.2.1 Log in the system	38
3.2.3.2.2 Reserve a car	39
3.2.3.2.3 Open a car	42
3.2.3.2.4 Power up a car	43
3.2.3.2.5 Modify profile information	45
3.2.3.2.6 Delete a reservation	47
3.2.3.2.7 End a ride selecting Money Saving	48
3.2.3.2.8 Select tourist attraction and going to an address	51
3.2.3.2.9 Stop a ride temporarily	53
3.2.3.3 Use Case : System and Operator	55
3.2.3.3.1 Notify an operator after a critical situation	55
3.2.3.3.2 Notify the user of the nearest safe area because of the low battery of the car	57
3.2.3.3.4 Apply discounts	59
3.3 Class Diagram	61
3.4 Statechart Diagram	62
3.5 Software System Attributes	62
3.4.1 Reliability	62
3.4.2 Availability	62
3.4.3 Security	62
3.4.4 Maintainability	62

3.4.5 Portability	62
3.6 Other Requirements	63
<b>4. Appendices</b>	<b>63</b>
4.1 Alloy Model	63
4.1.1 Signatures	63
4.1.2 Facts	65
4.1.3 Assertions	69
4.1.4 Predicates	70
4.1.4 Graphical Representation	71
<b>5. Used Tools</b>	<b>71</b>
<b>6. Effort Spent</b>	<b>72</b>
6.1 Giulia Leonardi	72
6.2 Marzia Degiorgi	72
6.3 Valentina Ionata	72
<b>7. Revision History</b>	<b>73</b>

# 1. Introduction

The RASD document is intended for describing the general features of the digital management application for car sharing, called “**Power Enjoy**”. This document is aimed to describe system objectives, functionalities and limits, starting from the needs of the stakeholders and in a form which fit for the purpose to create an analytics documentation, useful both for the communication between the customer and the developer and for subsequent implementations too. It addresses the developers and programmers who have to implement the requirements, and more generally, all the stakeholders involved in the software conception and fulfillment.

## 1.1 Purpose

The main general objectives of the system are:

- Simplify and integrate the urban transport
- Make the city eco-friendly, fighting the pollution produced by the gasoline engines

In order to fulfill these purposes, the software has to provide the following general functionalities:

- Allow **User** to create an account

*Supplying some basic informations, the User must be able to sign up for the service and to be recognized by it every time he/she logs in, just inserting his main personal data.*

- Allow **User** to find and reserve the nearest car from its location or from a given address

*Through the mobile app, the User must be able to find the cars available in a certain radius distance from him/her, and can make his/her choice by evaluating different characteristics of each car, as the distance or the battery level.*

- Allow **User** to unlock and power up the car reserved

*Making use of the car code, located on the windscreen of each car and of the QR code received as confirmation of the occurred booking.*

- Allow **User** to pay the charged amount

*The system will automatically apply some discounts to the drivers who had an ecological behaviour (or some penalties to the ones who did not satisfy any eco-friendly*

*requirement) and will withdraw the owed amount from the credit card given by the User during the registration.*

Power Enjoy system has also to allow to the **Operators** to monitor all the parameters of the cars, necessary to their maintenance, in order to permit them to identify the car which require any kind of upkeep.

### 1.1.1 Goals

- [G1] Allows Guests to sign up providing their credential.
- [G2] Allows Users to login.
- [G3] Allows Users to find the locations of available cars within a certain radius distance from their current location or from a specified address.
- [G4] Allows to reserve a car for up to one hour.
- [G5] For a reservation whose duration exceeds the prearranged time, the system charges a fee and set the car available again.
- [G6] Allows the only User who reserved the car earlier to unlock it, when in proximity.
- [G7] Allows the User to power up the reserved car.
- [G8] Charges the User from the moment he/she starts up the engine for a given amount of money per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.
- [G9] Notifies the User of the current battery level of the car real time.
- [G10] If required, allows the User to receive informations about the main tourist attractions in proximity, while driving.
- [G11] Allows the User to locate the nearest safe areas.
- [G12] Locks automatically when the User closes the car door, after exiting the vehicle.
- [G13] Allows operators to know exactly when cars need to be recharged or an intervention of different nature.
- [G14] Allows the User to add passengers when they reserve a car.
- [G15] Allows the User to delete a reservation.
- [G16] Allows the User to modify their profile information.

The peculiarity of the system is the will to repay the users' virtuous behaviours with generous discounts, and this implies the truthfulness of the following assertions:

- [G17] Reward the User for a good behaviour with discounts.
- [G18] Penalize the User for a improper behaviour with additional charges.

## 1.2 Scope

This System is a green economy business, due to its choice of managing only electric cars, and it is also one of the most competitive services in the field of car sharing, thanks to its eco-friendly peculiarity. The software product which is going to be developed will allow citizens, who most likely have not their own car with them, to rent a car and travel with it for all the time needed.

The innovation lies on the discounts on every ride charge, in response to driver's particularly green and environmentally-friendly behaviours, which permit users to save money, respecting the world.

For the proper functioning of the system, we suppose verified the underlying properties:

- The User is at least eighteen years old and he has got a valid driving licence.
- The GPS of all the cars cannot be switched off and it is always working.
- Users' GPS location is always exact and precise.
- A User can reserve only available cars.
- Each User can reserve no more than one car a time.
- Nobody can reserve the same car that is already reserved by another User.
- Users can not open an available car just by connected to its bluetooth, without having reserved it earlier.
- When a User inserts the code on the windscreen of the car reserved in the specific fields of the application, the car doors will unlock immediately.
- The key of the car is always inside the car.
- The key of the car cannot be taken off.
- A car that does not have the right condition to be used (e.g. low battery, maintenance problems) can not be shown in the set of available cars, on the application.

## 1.3 Definitions, Acronyms and Abbreviation

### 1.3.1 Definitions

**System:** the totality of the hardware/software applications that contribute to provide the service concerned.

**Guest:** everyone who is not a User of the system; he can only sign up.

**User:** everyone who has already signed up to the system, and has:

- name
- surname
- age
- password
- email address
- phone number
- licence code

**Operator:** workers with the following tasks:

- client assistance
- recharge electric cars
- recover cars that have been parked or left with no charge not in a safe areas
- maintenance operations

**Driver:** the User who reserved a car.

**Passenger:** the companions of the Driver. They have to be specified by the driver before picking up the car. They can be Users or Guests.

**Available car:** a car which does not have any reservation yet.

**Reservation:** the possibility to stall an available car, changing its state to unavailable. This allows the only User who reserved the car to unlock it, during the Reservation Time. The reservation includes informations about the number of passengers, the location of the Car, and the Reservation time. If there are other passengers in addition to the Driver, he has to specify their cell numbers, so that they can receive a one-time code to allow the car to recognize themselves.

**Booking Confirmation:** an overview of the reservation, sent to the User. It contains a one-time code to insert on the form, when asked by the connection process to the bluetooth of the car, and the QR code for the further authentication.

**One-time code:** a pin valid up to one hour from the reservation, which enables the User to unlock the car, by inserting it in the form of the process of connection to the car bluetooth.

**QR code:** a QR code is a type of 2D bar code that is used to provide an easy and quick authentication to the system. It has to be scanned by the on-board computer, in order to



verify the identity of the Driver.

**Safe Area:** a private parking area, belonging to the company, that includes a power grid station for cars, and where could be present one or more Operators.

**Ride:** the period from the moment when a car is powered up, to the moment when the User parks the car in a safe area and exits it.

### 1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- DB: DataBase.
- DBMS: DataBase management system.
- OTC: One-Time Code.
- OS: Operating System.
- RDBMS: Relational database management system.
- DMV: Department of Motor Vehicles.

### 1.3.3 Abbreviation

<i>Abbreviation</i>	<i>Definition</i>
<b>G[i]</b>	Identifier of the goal 'i'
<b>D[i]</b>	Identifier of the domain assumption 'i'
<b>FR[i]</b>	Identifier of the functional requirement 'i'
<b>NFR[i]</b>	Identifier of the non functional requirement 'i'
<b>UC[i]</b>	Identifier of the use case 'i'
<b>SC[i]</b>	Identifier of the scenario 'i'

## 1.4 References

- ISO/IEC/IEEE 29148 date Dec 2011
- RASD sample from Oct. 20 lecture.pdf
- "Mobile Application Design and Development", Erik Wilde, UC Berkeley school of

Information,2010

- “Green Move: A platform from highly configurable, heterogeneous electric vehicle sharing”, Andrea G. Bianchessi, Gianpaolo Cugola, Simone Formentin, Angelo C. Morzenti, Carlo Ongini, Emanuele Panigati, Matteo Rossi, Sergio M. Savaresi, Fabio A. Schreiber, Letizia Tanca, Edoardo G. Vannutelli Depoli, 2014

## 1.5 Overview

This document is essentially structured in four parts:

1. In this **first section**, we provide an introduction to the general problem that we are going to develop in our software.
2. The **second section** explains which are the main functionalities of the product and its requirements, assumptions and constraints.
3. In the **third section**, we are going to deeply analyze the functional and nonfunctional requirements of our system and to explain the structure using Use Cases and Sequence Diagrams.
4. Finally in the **fourth section**, we presented the Alloy contribution, appendices and team effort.

## 2. Overall Description

### 2.1 Product perspective

The system we will release is based on both a mobile application and a software running on the on-board computer, installed in each of the Power Enjoy group's car. It will have an internal interface for the administration: the Operators could interact with the system through an interface, for mobile devices, thought on purpose for them. Other features of the same mobile application address the Users, in the same way of the interface of the on-board computers. The system is a totally self-contained product and it is not integrated with other existing systems.

#### 2.1.1 System Interfaces

The product does not provide any external interface Systems.

#### 2.1.2 User Interfaces

The users' interface has to provide different functionalities: first of all, it has to be totally responsive. Then, we have to distinguish between different kind of users, who could be Guests, Users or Operators.

##### 2.1.2.1 Users

###### 2.1.2.1.1 Mobile interface

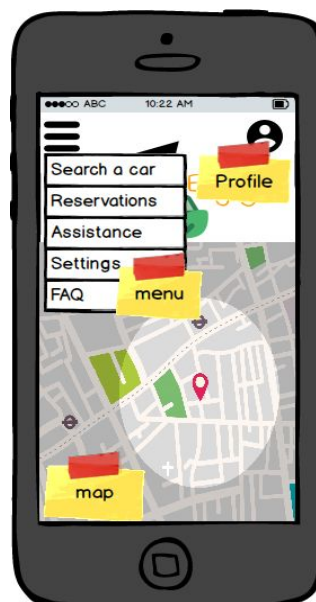
The following interfaces are the ones which address the User, set up on his/her mobile phone: in the first one, he/she has the possibility to choice if signing up, if he/she is not already registered, or if logging in.

Chosen the item "Sign up", he/she will be redirected into the second mockup, in which the user has to compile the requested fields with his/her personal information; on the other hand, choosing the item "Log in", he/she will be redirected to the form of the third mockup.



User Interface 1: Default layout for Guest( who can only sign up) or Unlogged User

The interface for Users,after the login:

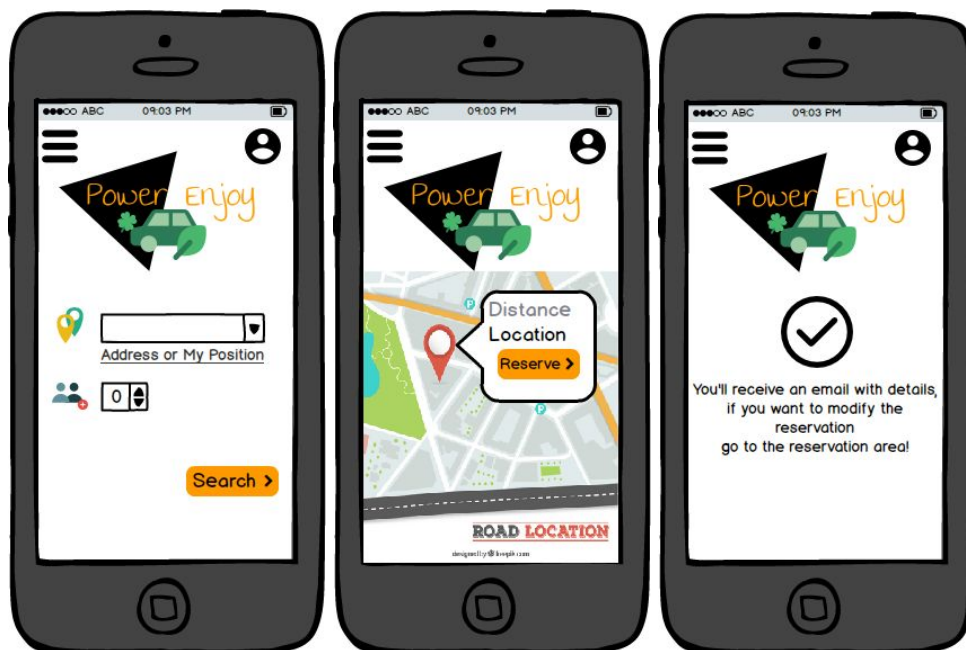


User Interface 2: User Home Layout

Once on this screen, the User can do the following basic actions:

- visualize and update account informations,
- visualize confirmed reservations,
- search a car,
- reserve a car,

as shown in the following mockups.



**User Interface 3: Search and Reserve a Car Layout**

And then the User can check his reservation on the dedicated area, in which he can do the following activities:

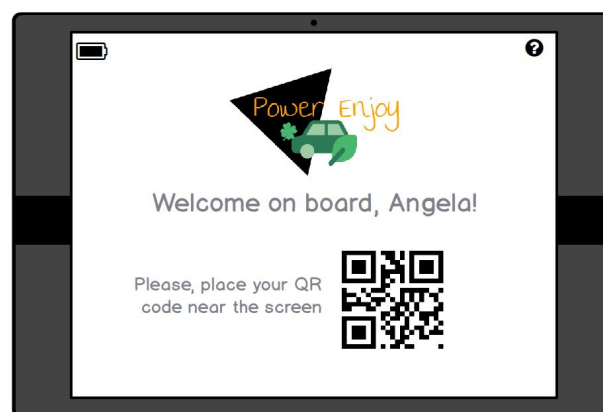
- Add some passenger.
- See the details of the reservation such as expiration time, address where pick up the car.
- Delete the reservation.
- Show OTC and QRcode.



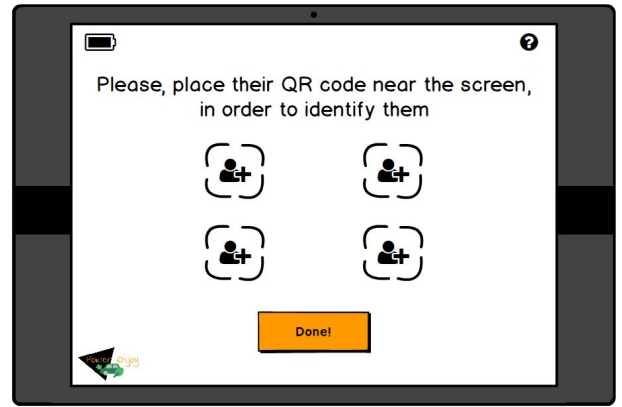
**User Interface 4: Personal Reservation Area Layout**

#### 2.1.2.1.2 On-board interface

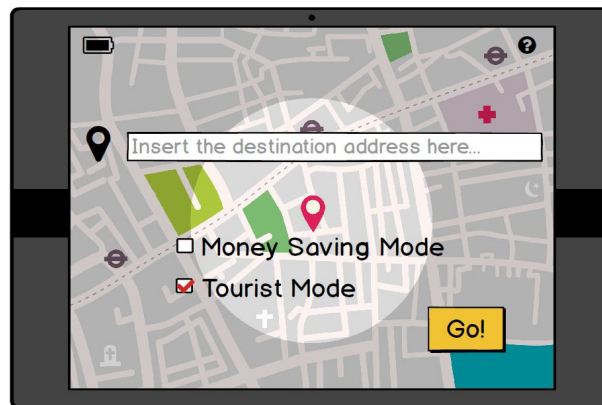
This is the interface of the software set up on the on-board computer of the cars. It will allow Users to choose a particular modality, as the “Tourist” one or the “Money Saving Mode”, to specify an address in order to be guided up to there, to keep under control the charge and the battery level of the car, while driving.



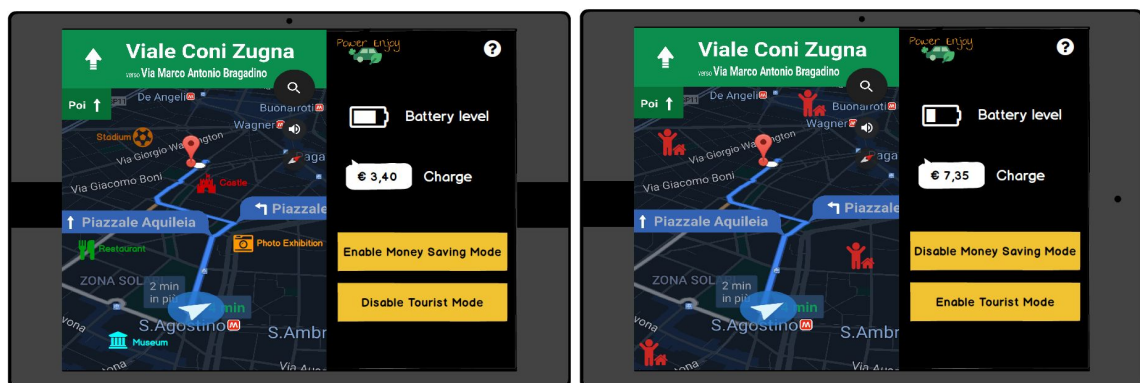
**User interface 5: Welcome and Authentication Layout**



User interface 6: Passengers' Authentication Layout



User interface 7: Search and Modalities Layout



User Interface 8: Tourist Mode and Money Saving Mode Layout



**User Interface 9: Stop Or End Of The Ride Layout**

#### 2.1.2.2 Operators

The Operators have to access the mobile application with:

- Operator's code
- password



**User Interface 10: Operator Login**

After having logged in the mobile application, the Operators can see the location of the cars, which are into the zone they are entrusted with, on the map. When the system notifies the need of the recovery of a certain car, the Operator receives the notification



on his mobile application - which shows him where the car is located and its need - and confirms that the recovery has been done successfully, by rendering the car available again with a click.



**User Interface 11: Operator Home layout and Notification Layout**

### 2.1.3 Hardware Interfaces

This system does not have any hardware interfaces.

### 2.1.4 Software Interfaces & Operating System

<i>Database management system</i>	
<i>Name</i>	MySQL
<i>Mnemonic</i>	MySQL
<i>Specification number</i>	Community Edition
<i>Version number</i>	15.7.16
<i>Source</i>	<a href="https://dev.mysql.com/downloads/mysql/">https://dev.mysql.com/downloads/mysql/</a>

### 2.1.5 Operations

The system can be used by Users, Guests and Operators. The operations are those described in the product functions, written below.

Users can mainly find the cars available around them and reserve them; Guests can be just passengers of a trip reserved by an User; Operators must be able to interact with the system so as to be able to take care of the cars.

Furthermore, the system has also to allow to the administrator to manage some basic features. He has to be able to change the number of cars available and safe areas (the company could decide to extend the number of vehicles and facilities at all times), to insert a new Operator into the system, every time a new one is employed, and to redetermine the urban boundaries, which any car can not exceed. Additionally, the administrator of the system has to periodically monitor the use of RAM memory and the processor, in order not to compromise the software functionality, due to an overloaded server.

### 2.1.6 Site Adaption Requirement

For installing the software is necessary to have installed in the selected server the following item:

- DBMS
- RDBMS

It's sufficient for the Users to have any kind of mobile device (iOS, Android or Windows Phone). The QR reader has to be set up on the on-board computer of each car.

## 2.2 Product functions

### → Functional Requirements

**[G1]** Allows Guests to sign up providing their credential.

- **[FR 1.1]** The system must verify that the email address has not to be registered yet on the DBMS.
- **[FR 1.2]** The system must verify that the payment method inserted is valid.
- **[FR 1.3]** The system allows Guests to register themselves, by inserting their personal informations.
- **[FR 1.4]** Guests must be able to see only the "Sign up and Log in" layout.

**[G2]** Allows Users to login.

- **[FR 2.1]** The system must verify that the entered username and password are already registered into the DBMS.

- **[FR 2.2]** If username and password are correct , the system allows the User to log in.

**[G3]** Allows Users to find the locations of available cars within a certain radius distance from their current location or from a specified address.

- **[FR 3.1]** The system must know the position of all the cars that are available.
- **[FR 3.2]** The system must show the Users the position of the available cars within a certain radius distance from their position.
- **[FR 3.3]** The system must show the Users the position of the available cars within a certain radius distance from the address inserted by him/her.

**[G4]** Allows to reserve a car for up to one hour.

- **[FR 4.1]** Since the User pushes the reservation button, the system must set as unavailable the car reserved for all the other Users.
- **[FR 4.2]** The system must allocate an OTC to the reservation.

**[G5]** For a reservation whose duration exceeds the prearranged time, the system charges a fee and set the car available again.

- **[FR 5.1]** If after one hour the User has not opened the car yet, the system has to charge him a fee of 1€.
- **[FR 5.2]** When the reservation time is up, the system must set the car available again.

**[G6]** Allows the only User who reserved the car earlier to unlock it, when in proximity.

- **[FR 6.1]** The system must allow the User to find the car bluetooth signal.
- **[FR 6.2]** The system must allow the User to connect to the car bluetooth signal.
- **[FR 6.3]** The system must verify that the OTC inserted by the User matches with the one expected.
- **[FR 6.4]** The system must unlock the car, if the OTC matches.

**[G7]** Allows the User to power up the reserved car.

- **[FR 7.1]** The system must check if the QR code of the Driver matches with the one allocated to him/her during the reservation process.
- **[FR 7.2]** The system must check the QR code of all the passengers to apply a price reduction of 10%, if there are at least two other passengers into the car.
- **[FR 7.3]** The system must allow the User to power up the car just pushing the “Start and Stop” button, if the User has been already authenticated.

**[G8]** Charges the User from the moment he/she starts up the engine for a given amount of money per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.

- **[FR 8.1]** When the User leaves the car, the system must ask the User if he wants to stop or to end the ride.
- **[FR 8.2]** The system has to show on the screen the current charges real time.
- **[FR 8.3]** When the User ends the ride the system must charge him for a given amount of money per minute, applying the related discounts and penalties.

**[G9]** Notifies the User of the current battery level of the car real time.

- **[FR 9.1]** The system must show the current charge of the car battery on the screen of the on-board computer.
- **[FR 9.1]** The System must show an alarm when the battery is particularly low, and show the nearest safe areas from the current position of the car.

**[G10]** If required, allows the User to receive informations about the main tourist attractions in proximity, while driving.

- **[FR 10.1]** The system must show on the map of the on-board device the most fascinating and popular tourist attractions, as museums, exhibitions, monuments or buildings interesting from an architectural point of view, while he/she is moving in proximity of them.
- **[FR 10.2]** If the User clicks on an attraction, the system has to display a brief description of it.

**[G11]** Allows the User to locate the nearest safe areas.

- **[FR 11.1]** The system has to show on the map of the on-board device the safe areas available, as long as the User is driving within a certain distance from them.
- **[FR 11.2]** In case of emergency, the system has to notify immediately the problem to the User and to point him/her the nearest safe areas where to stop and let the car be repaired.
- **[FR 11.3]** If the User chooses the “Money Saving Mode”, the system must show, on the screen of the on-board device, all the safe areas in proximity of the Driver’s route, until the “Money Saving Mode” is enabled.

**[G12]** Locks automatically when the User closes the car door, after exiting the vehicle.

- **[FR 12]** The system must lock the car after the User has expressed the will of ending or stopping temporarily the ride, has left the vehicle and has closed the doors.

**[G13]** Allows operators to know exactly when cars need to be recharged or an intervention of different nature.

- **[FR 13.1]** The system must monitor the condition of the car, in order to be able to see if the car needs to be charged or repaired.

- **[FR 13.2]** The system has to send a request for intervention to the operator to whom was assigned the area in which the car is.

**[G14]** Allows the User to add passengers when they reserve a car.

- **[FR 14.1]** The User must be able to add at most 4 passengers to the reservation.
- **[FR 14.2]** The system must send a QR code to all the passengers, in order to be recognized for a certain the ride.

**[G15]** Allows the User to delete a reservation.

- **[FR 15]** After the reservation, the system must permit the User to delete it.

**[G16]** Allows the User to modify their profile information.

- **[FR 16]** The System must enable the User to change its personal data, that could be obsolete.

**[G17]** Reward the User for a good behaviour with discounts.

- **[FR 17.1]** The system must apply a discount of 10%, if the passengers are more or equal than 3.
- **[FR 17.2]** The system must apply a discount of 20%, if the car is left with more than 50% of battery.
- **[FR 17.3]** The system must apply a discount of 30%, if the User brings the car in a safe area and puts it in charge.
- **[FR 17.4]** If the User parked in an area using “Money Saving” option, the system must apply a discount.

**[G18]** Penalize the User for a improper behaviour with additional charges.

- **[FR 18.1]** The system must apply a penalty of 30%, if the car is left with less than 20% of battery.
- **[FR 18.2]** The system must apply a penalty of 30%, if the car is left at more than 3 km from the nearest station.
- **[FR 18.3]** The system must charge the associated User for every exceptional penalty due to that User.

## → **Nonfunctional requirements**

- **[NFR 1.1]** The System must store passwords in a safe way.
- **[NFR 1.2]** The email address used for the registration must be formally correct.
- **[NFR 1.3]** The system has access to the DB of the DMV, in order to verify the authenticity of the driving licence.
- **[NFR 3.1]** The GPS of the cars must be on 24 hours a day.

- [NFR 4.1] The system must be able to count down the minutes since the reservation, up to one hour.
- [NFR 4.2] The OTC is delivered to the User in a safe way.
- [NFR 4.3] The QR code is delivered to the User and to the Passengers in a safe way.
- [NFR 6.1] The car bluetooth signal is available 24 hours a day.
- [NFR 10.1] The system must update its maps every day.

## 2.3 User characteristic

The main characteristic of our users should be:

- basic knowledge in using simple mobile and web application

The main characteristic of our users must be:

- owning a valid driving licence
- owning a mobile device

## 2.4 Constraints

### 2.4.1 Regulatory policies

The System should require the following permissions to the User:

- Identity
- Contacts
- GPS Location
- SMS
- Information about Wi-Fi connection

### 2.4.2 Hardware limitation

**Mobile application:**

- 3G connection
- Storage capacity
- GPS
- Bluetooth connection

**On-board application:**

- Storage capacity
- GPS
- Bluetooth connection
- Camera
- 3G connection

### 2.4.3 Interfaces to other applications

1. **Interface with the DBMSs:** the system will interface with the DBMSs installed on the server.
2. **Interface with the email gateway provider:** the system has to interact with the email gateway provider of the Users in order to send them the confirmation email, after every reservation.
3. **Interface with the push service, via own APIs:** the system has to send push notifications to the Operators, in order to inform them about the necessity to recover a certain car and where it is located.
4. **Interface with the car system:** the system has to be able to access to the data of the car and to share some informations with its system.

### 2.4.4 Parallel operation

The server has to support operations by the mobile application of different Users and different Operators, at the same time.

## 2.5 Assumptions and Dependencies

The following assumptions have been taken into account:

- There is enough space in the server where the application is installed for the daily updates of the maps.
- Accurate car locations are known by GPS.
- When a destination address is inserted, the map with the route will be available in at most 30 seconds.
- Operators define a car available only when all its issues are resolved.
- As soon as the mobile device tries to pair with the car by bluetooth, a screen on the mobile device asks the User to insert the correct OTC.
- By putting the mobile device with the QR code on its screen near the camera of the on-board computer, the QR reader will scan the QR and authenticate the User/Passenger.
- The internet connection works correctly both on the mobile device and on the on-board computer.
- The map updates contain the location of the current tourist attractions.
- A car which is set available again, is shown the Users among the free cars on the mobile application instantly.

- The User can not hold the car more than 24 hours.
- A squared section of the territory is assigned to each Operator
- Each squared section is identified by the coordinates of its centre and the length of its sides

## 3. Specific Requirements

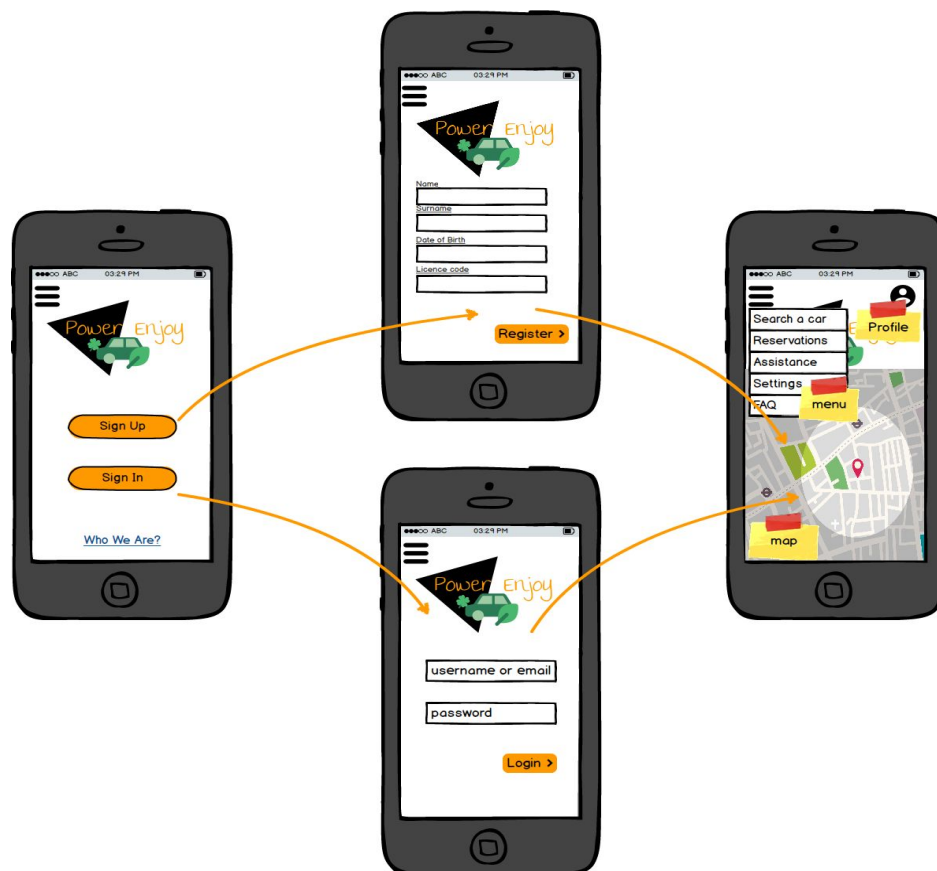
### 3.1 External Interface Requirements

#### 3.1.1 User Interface

##### 3.1.1.1 Users

##### 3.1.1.1.1 Mobile interface

The following storyboard shows how a Guest and a User can respectively reach the home page of the “Power Enjoy” mobile application.



Storyboard 1: Storyboard of the user interface



### 3.1.1.2 Operators

The following storyboard shows the flow from the log in of an operator up to the notification alarm from the system.



**Storyboard 2: Storyboard of the operator interface**

## 3.2 Functional Requirement

### 3.2.1 Scenarios

#### 3.2.1.1 Sign up in the System

Code	SC001
Description	Describes how an unregistered user can sign up for the system.
Goal	[G1] Allows Guests to sign up providing their credential.
Assumption	The User is not registered.
Angela wants to visit a new shopping center in the suburb of Milan, but the public transport is	

not the best choice, because of its sporadic rides. Therefore, she is looking for a car sharing service application that will allow her to stay in the shopping center all the time that she wants, without concerns about the return. During this research, she finds the new service **Power Enjoy**, and after having read about the service and the resources, she decides to download it and sign up.

For the registration, the System provides a form with different fields: username, name, surname, email address, payment credential, licence code, birth information, and optional personal information to complete the profile. Hence, Angela fills in all the fields and pushes the registration button. The system confirms the registration and sends to Angela an email with a secure password that allows her to login into the system from there.

Now, Angela can access to the services provided by the system, and she's ready to try them for her shopping day.

### 3.2.1.2 Log in and Reserve a car

Code	SC002
Description	Describes how a user can reserve a car.
Goal	<p>[G2] Allows Users to log in.</p> <p>[G3] Allows Users to find the locations of available cars within a certain radius distance from their current location or from a specified address.</p> <p>[G4] Allows to reserve a car for up to one hour.</p>
Assumption	The User is already registered in the system.
<p>The day after the registration, Angela logs in the system with her email address and password. Then, she looks curiously all the application services and how they work.</p> <p>It is 5 p.m and she is ready to reserve a car for her shopping day. Therefore, she clicks on the menu and selects the item "Search a car".</p> <p>At this point, the application asks to fill in two fields: the address, or current location, and the number of passengers. She decides to search the nearest car from her position and to travel alone: the application asks her to activate the GPS location. Once she has done that, she clicks on the "Search" button. Now, the app displays a map with two available cars near Angela: one which is distance 1 km from her, and the other one which is distance 500 m - it is indicated for both of them the exact address. Angela chooses the nearest one and pushes the "Reserve" button.</p> <p>The system confirms the reservation and sends to Angela an email with further details. Angela has only one hour to pick up the car: she has to hurry if she do not want to lose her reservation!</p>	

### 3.2.1.3 Pick up the car

Code	SC003
Description	Describes how open and start a ride with a car.
Goal	[G6] Allows the only User who reserved the car earlier to unlock it, when in proximity. [G7] Allows the User to power up the reserved car.
Assumption	The User is already registered in the system. The User is already logged in the system. The User has already reserved a car.

Angela found the address of the reserved car and reached it. Now, she is next to the car and launches the application, goes to the reservation area and follows the instructions step to step. At first, she switches on the bluetooth connection of her mobile phone, searches among the devices available to pair close to her the one named "PowerEnjoy345"- since "345" is the car number communicated to her through the confirmation via email - and clicks on it. A screen asking for her OTC appears. Angela does not remember the correct OTC linked to her current reservation, therefore she checks it on her personal area of the Power Enjoy mobile application. Now, she is able to insert the right OTC: as soon as filled out the field, the car in front of her unlocks itself. Then, Angela gets in it. The display of the car welcomes Angela and asks her to identify herself by putting the QR code sent to her during the reservation near the screen. She opens again the reservation area on her mobile application, clicks on the QR code and shows it to the screen of the on-board computer. The car display accepts it and Angela is finally able to power up the engine, just by pushing the "Start and Stop" button near the steering wheels.

Before beginning to drive, she looks carefully at the display and notice the main functionalities that it offers: battery level of the car, real time charge, the possibility of adding the passengers declared during the reservation, if necessary. Then she notices the possibility to enable two different modalities of use: the "Money Saving Mode" and the "Tourist One". Since she would like to save as much money as possible, she puts her tick on the "Money Saving Mode", inserts the address of the shopping centre and starts her ride.

#### 3.2.1.4 End of a ride

Code	SC004
Description	Describes the end of a ride.
Goal	<p>[G8] Charges the User from the moment he/she starts up the engine for a given amount of money per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.</p> <p>[G9] Notifies the User of the current level of battery of the car real time.</p> <p>[G10] Locks automatically when the User closes the car door, after exiting the vehicle.</p>
Assumption	<p>The User has already reserved a car.</p> <p>The User has already picked up and started the car.</p>
<p>Angela is driving to the new shopping centre and she is very excited. When she arrives, she stops in the parking area and the car display asks her to select a choice between: “stop” and “end the ride”. She selects “stop”, and exits the car. After few seconds, the car automatically locks and she heads towards the shopping center.</p> <p>After three hours, she comes back and opens the car through the application, as she did the first time. She selects on the display to restart the car. The car powers up and Angela has a look at the current amount charges to her of the car, and at the battery level, selects the “Money Saving Mode”, insert her address and begins to drive.</p> <p>When she arrives in the safe area nearest to her home, indicated to her by the modality chosen, she stops the car and selects “end of the ride” : the car stops to charge her. She exits the car and it automatically locks itself.</p> <p>Angela is really satisfied with the service and the final charge, for such a long time, is lower than the one of any other car sharing she has experienced so far. She heads towards home, while thinking that she will certainly use the service again.</p>	

### 3.2.1.5 Use Discounts services

Code	SC005
Description	Describes how a User can use discount services.
Goal	[G12] Allows Users to add passengers when they reserve a car. [G15] Reward the User for a good behaviour with discounts.
Assumption	The User is already registered in the System. The User is already logged in the system. The User has already reserved a car.

Angela is so satisfied with her first experience with the Power Enjoy service that she decides to reserve an other car to go for a ride to the city with some friends. In order to do that, she goes, as the previous time, in the reservation area of the application, and clicks the “Add Passenger” button. The application asks Angela to specify the email address for each passenger, in order to allow the system to send the personal QR code to them and to permit to them to be identified by the car for the ride.

Angela adds two passengers, Laura and Mark, and heads towards to pick up the car. When she arrives to the safe area where it’s parking, she authenticates herself by providing her OTC and QR code, she powers up the engine and goes to pick up her friends. Laura and Mark are waiting for Angela at their home, and they look forward to spend sometimes with her.

When Angela arrives to her friends’ house, they get in the car. Angela selects twice “Add Passenger” on the display and each friend shows the corresponding QR Code to the on-board camera. The System recognizes them and notifies Angela that a discount of 10% will be applied to the total charge, at the end of the ride.

Angela and her friends are ready to leave: they want to have a tour of the main attractions of the city, but Angela loves shopping, and she does not want to miss the opportunity to go around the multiplicity of shop of Milan. Therefore, Angela inserts as destination the city-centre and enables the “Tourist Mode”, in order to be notified of all the most interesting monuments and museums.

During the ride, they decided to stop to “San Siro”, because of the Mark’s passion for Milan football teams, to the church “Santa Maria delle Grazie” in order to visit the amazing “Cenacolo Vinciano” and then to the “Brera art gallery”. When they, finally, arrive to the city centre the car has less than 50% of charge: therefore, Angela decides to leave the car in a safe area and recharge it.

At the end of the ride, Angela has two discounts: one applied for the passengers and the other, of 30%, because she has plugged the car by means of the power grid. The system automatically

charges the debit card of Angela and they move close to the shop window of the first store.

### 3.2.1.6 Bad behaviour and recharge

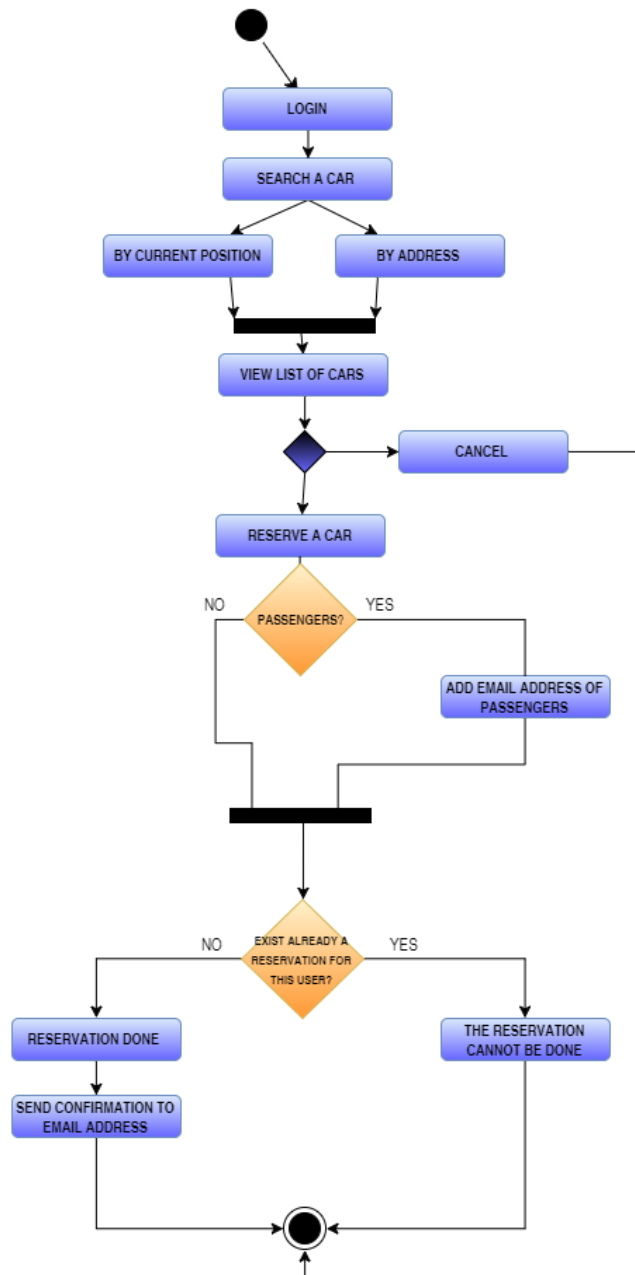
Code	SC006
Description	Describes how a User can assume a bad behaviour, and what is the response of the system.
Goal	<p>[G11] Allows operators to know exactly when cars need to be recharged or an intervention of different nature.</p> <p>[G5] For a reservation whose duration exceeds the prearranged time, the system charges a fee and set the car available again.</p> <p>[G16] Penalize the User for a improper behaviour with additional charges.</p>
Assumption	<p>The User is already registered in the System.</p> <p>The User is already logged in the system.</p> <p>The User has already reserved a car.</p>
<p>Mark, Angela's friend, having experienced the Power Enjoy service thanks to her, has decided to register and to reserve a car. But he is lazy, the time passes and he does not succeed in reaching the reserved car in time.</p> <p>When the hour expires, the system charge him with a fee of 1€ on his credit card.</p> <p>Mark has to meet a friend of his and it's really late: he reserves an other car and reaches it immediately. Finally he meets his friend and the afternoon passes; then he picks up again the car to go home.</p> <p>While he is driving lost in his mind, he realizes that he has used more than 80% of battery. So when he ends the ride, the system charges 30% more to the final amount.</p> <p>Mark, exhausted of this unlikely day, goes home.</p> <p>The system notifies the associated Operator about the need of an intervention of recovery, to recharge the car.</p>	

### 3.2.1.7 Delete a reservation and change profile information

Code	SC007
Description	Describes how an User can interact with his/her account and how can modify a reservation.
Goal	[G13] Allows User to delete a reservation. [G14] Allows User to modify their profile information.
Assumption	The User is already registered in the system. The User is already logged in the system.
<p>Angela has to meet Laura for a drink in a new pub in Milan. Therefore, she reserves a car with the Power Enjoy company, because she knows that is the most comfortable option.</p> <p>After a while, Laura calls Angela and says to her that she has just registered in the system and has already booked a car to pick her up. Angela agrees, and try to delete her reservation. She goes to the reservation area and delete her booking in time, so that she will not pay any extra fee.</p> <p>Angela remembers also that she has to change her payment method because her old credit card is expired. She goes into her application again, selects "Account Information" and modifies the associated data.</p> <p>"Tuuuuu Tuuuuu", a clacson. Oh it is Laura! It is time to go now!</p>	

### 3.2.2 Activity diagram

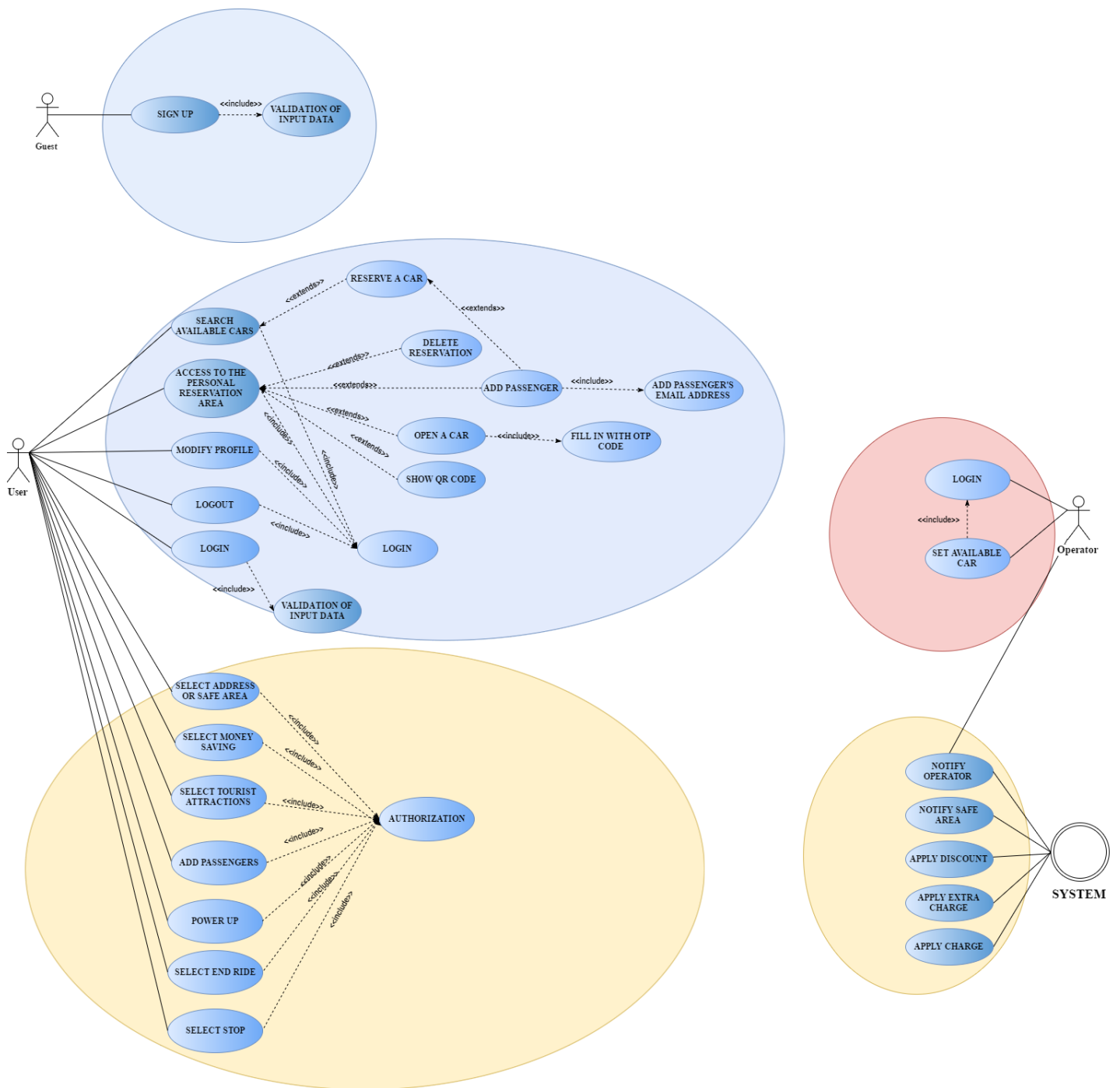
The purpose of the system is the management of an electric car sharing software. Then, here we represent the activity diagram for create a reservation of a car.



**Activity Diagram 1: Creating a reservation of a car**



### 3.2.3 Use Case Model



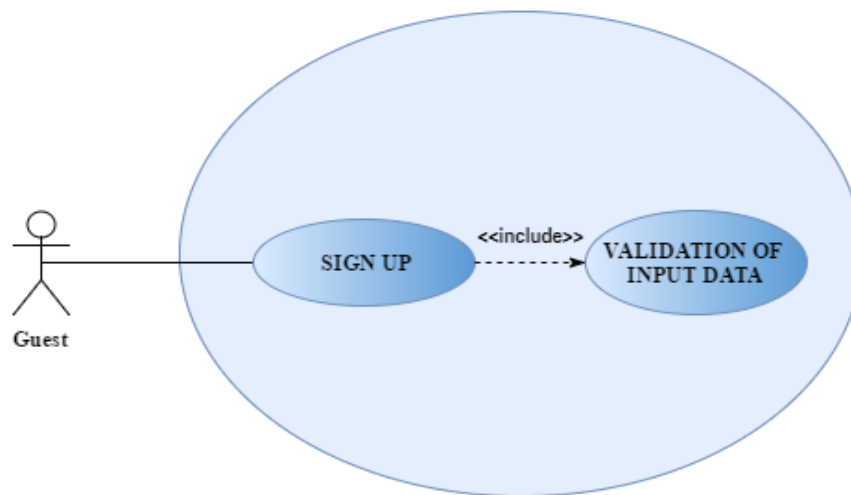
**Use Case 1: Model**

We can derive some use cases from the previous scenarios, First of all we can distinguish three main actors:

- **Guest**
- **User**
- **Operator**
- **System**

The system is explicit to underline some specific and particular functionalities.

### 3.2.3.1 Use Case : Guest

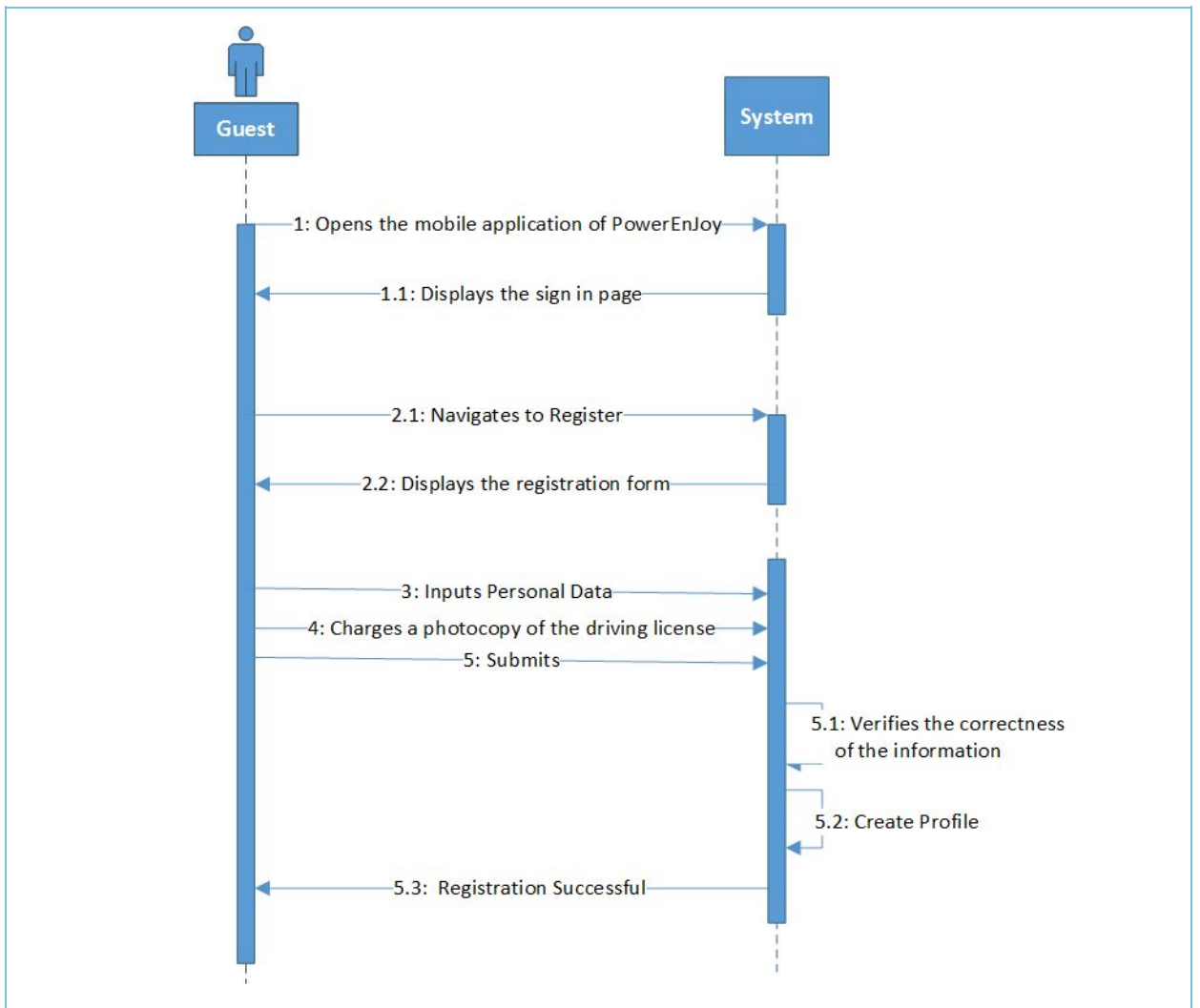


#### Use Case 2: Guest

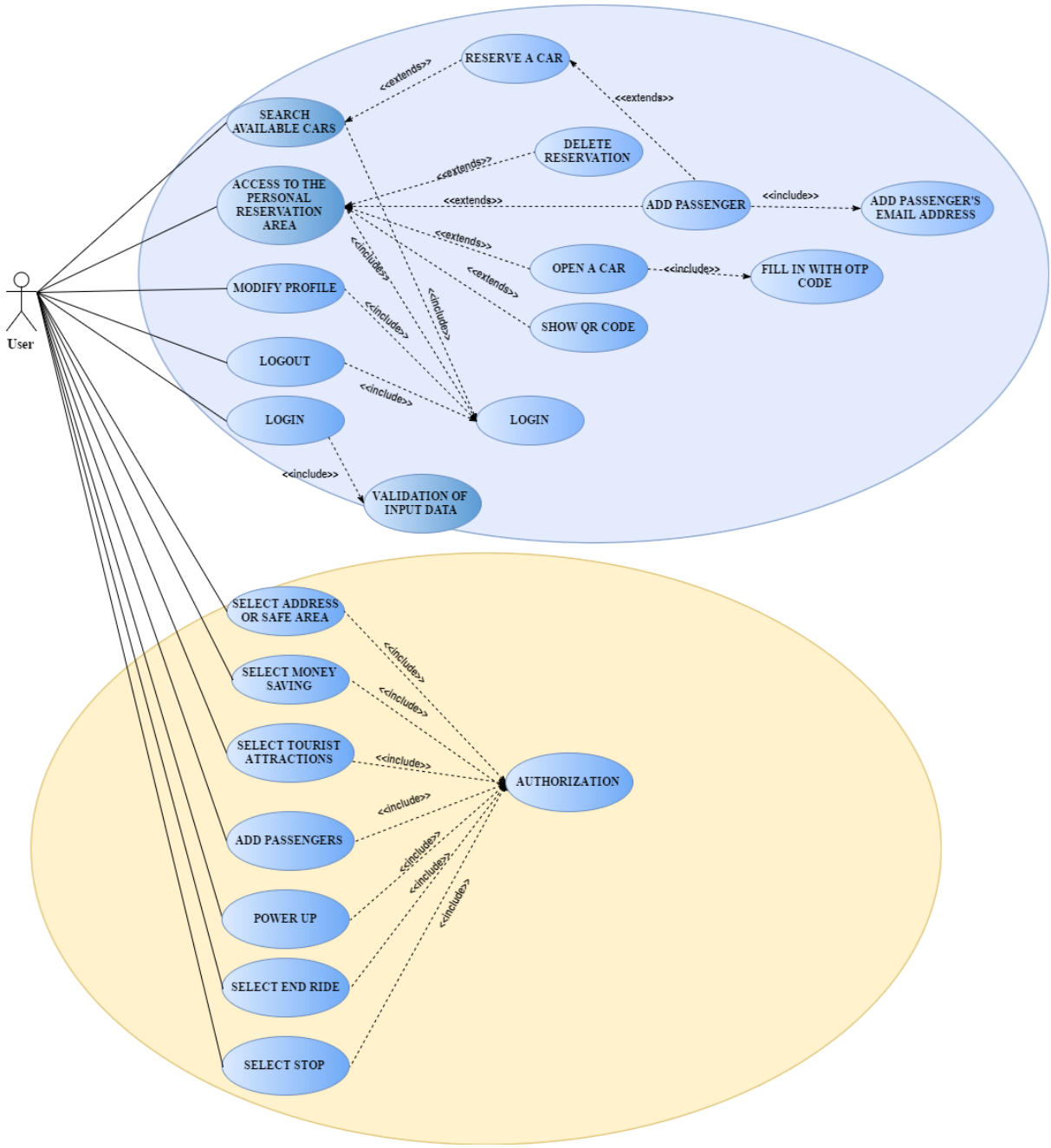
##### 3.2.3.1.1 Register in the system

Register in the system	
Code	UC1
Goal	[G1] Allows Guests to sign up providing their credential.
Requirement	[FR 1.1] The system must verify that the email address has not to be registered yet on the DBMS. [FR 1.2] The system must verify that the payment method inserted is valid.

	<p>[FR 1.3] The system allows Guests to register themselves, by inserting their personal informations.</p> <p>[FR 1.4] Guests must be able to see only the “Sign up and Log in” layout</p>
Assumption	The Guest is not registered in the system.
Actors	Guest
Entry Condition	The Guest downloads the “Power Enjoy” mobile application and opens it.
Exit condition	All the information fields are successfully filled and saved.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The Guest download the “Power Enjoy” mobile application.</li> <li>2. The Guest opens the mobile application.</li> <li>3. The Guest selects the registration option.</li> <li>4. The system provide the registration form.</li> <li>5. The Guest fills in the fields of the form and save information.</li> <li>6. The System controls the correctness of the inserted inputs.</li> <li>7. The System confirmed the registration and displays the homepage.</li> </ol>	
Exception	Mandatory data are not correct.
Non Functional Requirement	<p>[NFR 1.1] The System must store passwords in a safe way.</p> <p>[NFR 1.2] The email address used for the registration must be formally correct.</p> <p>[NFR 1.3] The system has access to the DB of the DMV, in order to verify the authenticity of the driving licence.</p>
Sequence Diagram	



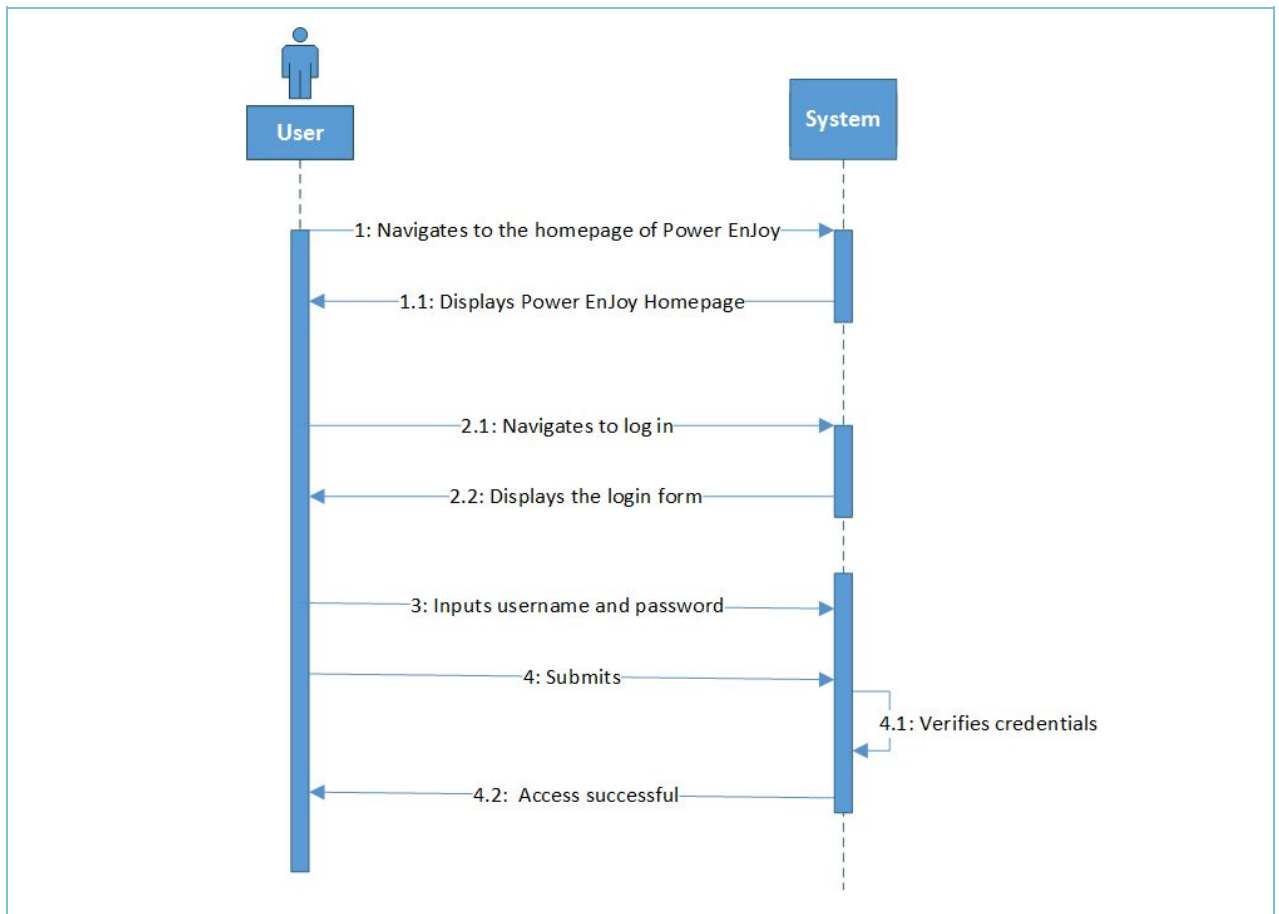
3.2.3.2 Use Case : User



Use Case 3: User

### 3.2.3.2.1 Log in the system

Log in the System	
Code	UC2
Goal	[G2] Allows User to Log in
Requirement	<p>[FR 2.1] The system must verify that the entered username and password are already registered into the DBMS.</p> <p>[FR 2.2] If username and password are correct , the system allows the User to log in.</p>
Assumption	The User is registered in the system.
Actors	User
Entry Condition	The user opens the “Power Enjoy” mobile application.
Exit condition	The informations required(username or email and password) are correct.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User opens the mobile application “Power Enjoy”.</li> <li>2. The User selects the “Sign in” option.</li> <li>3. The User inputs username/email address and password.</li> <li>4. The System verifies correctness of username and password.</li> <li>5. The System displays the homepage of the application.</li> </ol>	
Exception	Wrong inputs filled in by the User.
Non Functional Requirement	
Sequence Diagram	



### 3.2.3.2.2 Reserve a car

Reserve a car	
Code	UC3
Goal	<p>[G3] Allows Users to find the locations of available cars within a certain radius distance from their current location or from a specified address.</p> <p>[G4] Allows to reserve a car for up to one hour.</p> <p>[G14] Allows the User to add passengers when they reserve a car.</p>
Requirement	<p>[FR 3.1] The system must know the position of all the cars that are available.</p> <p>[FR 3.2] The system must show the Users the position of the available cars within a certain radius distance from their position.</p>

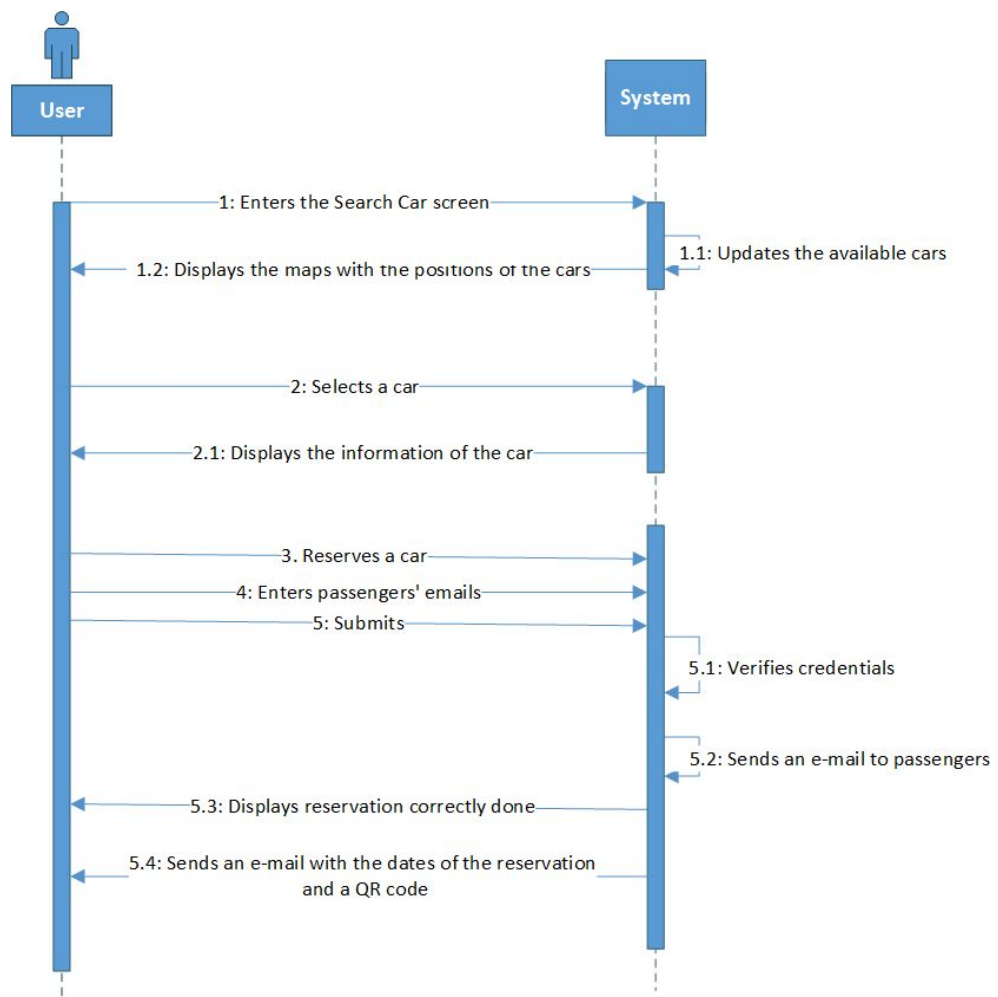
	<p>[FR 3.3] The system must show the Users the position of the available cars within a certain radius distance from the address inserted by him/her.</p> <p>[FR 4.1] Since the User pushes the reservation button, the system must set as unavailable the car reserved for all the other Users.</p> <p>[FR 4.2] The system must allocate an OTC to the reservation.</p> <p>[FR 14.1] The User must be able to add at most 4 passengers to the reservation.</p> <p>[FR 14.2] The system must send a QR code to all the passengers, in order to be recognized for a certain the ride.</p>
Assumption	<p>The User is registered.</p> <p>The User has logged in.</p> <p>The User navigates in the homepage.</p>
Actors	User
Entry Condition	The User navigates to the search page.
Exit condition	The User has reserved a car.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User selects the "Search a car" option.</li> <li>2. The User inserts the input informations and clicks the search button.</li> <li>3. The System displays a map with all the available car compare to the input data.</li> <li>4. The User selects a car and reserves it.</li> <li>5. If the User has specified the presence of passengers, he has to fills in the information about them.</li> <li>6. The system confirms the operation, sends an email with a one time code and a QR code, and also saves these informations in the personal reservation area of the user.</li> </ol>	
Exception	<p>-The User cancels the operation.</p> <p>- Any cars in the specified location with a certain radius distance is available.</p>
Non Functional Requirement	<p>[NFR 3.1] The GPS of the cars must be on 24 hours a day.</p> <p>[NFR 4.1] The system must be able to count down the minutes since the reservation, up to one hour.</p>



[NFR 4.2] The OTC is delivered to the User in a safe way.

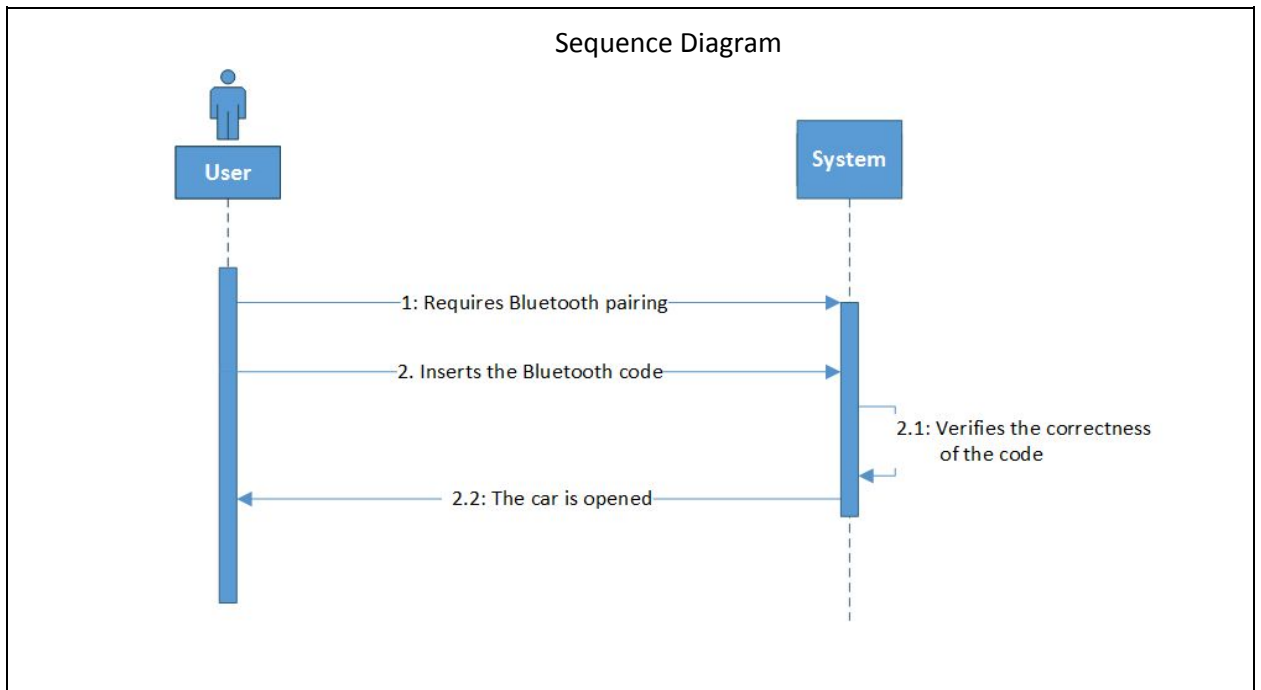
[NFR 4.3] The QR code is delivered to the User and to the Passengers in a safe way.

### Sequence Diagram



### 3.2.3.2.3 Open a car

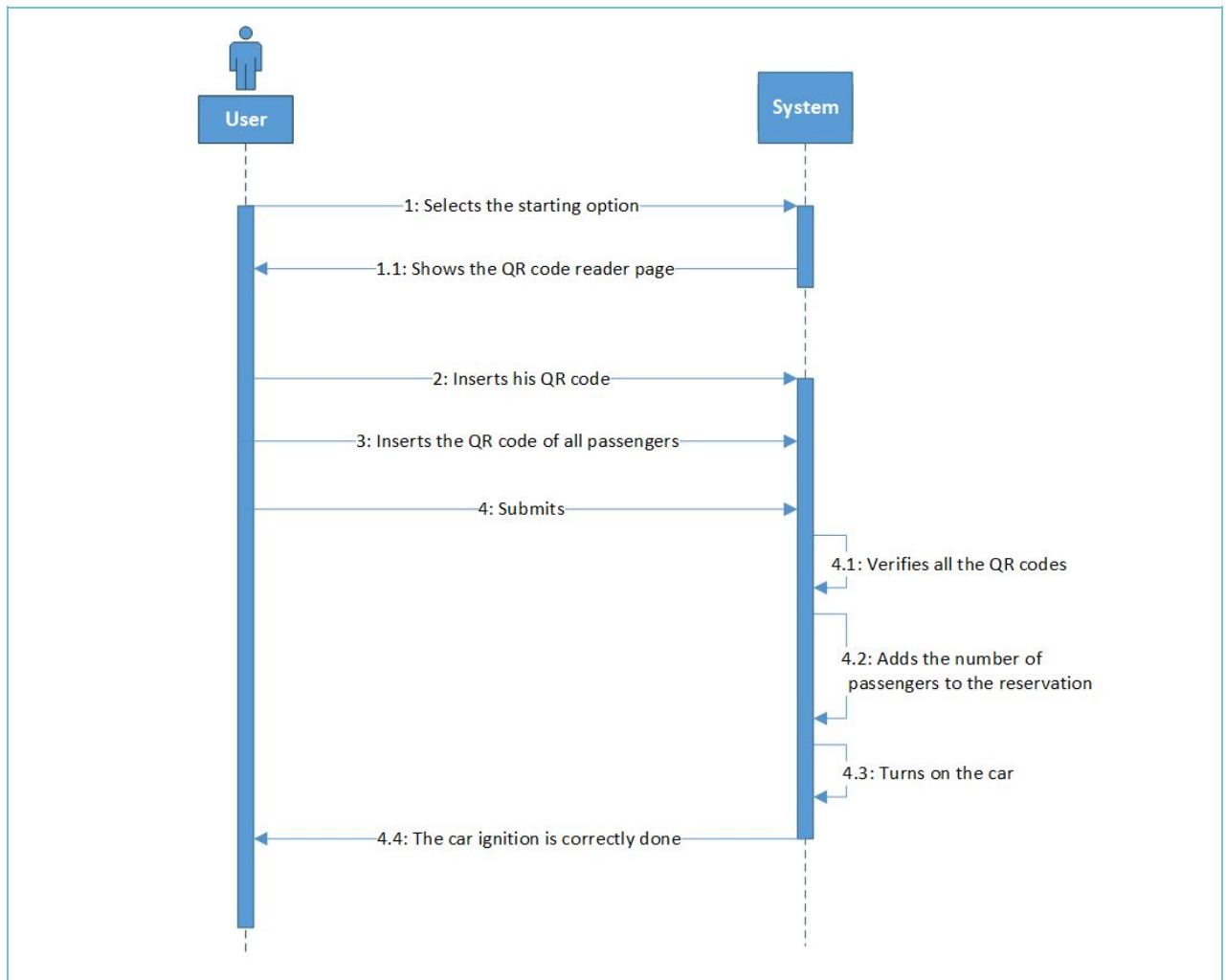
Open a car	
Code	UC4
Goal	[G6] Allows the only User who reserved the car earlier to unlock it, when in proximity.
Requirement	<p>[FR 6.1] The system must allow the User to find the car bluetooth signal.</p> <p>[FR 6.2] The system must allow the User to connect to the car bluetooth signal.</p> <p>[FR 6.3] The system must verify that the OTC inserted by the User matches with the one expected.</p> <p>[FR 6.4] The system must unlock the car, if the OTC matches.</p>
Assumption	<p>The User is registered.</p> <p>The User has logged in.</p> <p>The User has already reserved a car.</p>
Actors	User
Entry Condition	The User is in proximity of the reserved car and he has navigated to the personal reservation area in the mobile application "Power Enjoy", to take the code car.
Exit condition	The car unlocks.
Flow Of Events	
<ol style="list-style-type: none"> <li>The User checks the one time code of the reservation in his personal area on mobile application.</li> <li>The User requires the bluetooth pairing and inserts the code.</li> <li>The system unlocks the reserved car.</li> </ol>	
Exception	The User inserted a wrong code to open the car.
Non Functional Requirement	[NFR 6.1] The car bluetooth signal is available 24 hours a day.



#### 3.2.3.2.4 Power up a car

Power up the reserved car	
Code	UC5
Goal	<p>[G7] Allows the User to power up the reserved car.</p> <p>[FR 7.1] The system must check if the QR code of the Driver matches with the one allocated to him/her during the reservation process.</p> <p>[FR 7.2] The system must check the QR code of all the passengers to apply a price reduction of 10%, if there are at least two other passengers into the car.</p> <p>[FR 7.3] The system must allow the User to power up the car just pushing the “Start and Stop” button, if the User has been already authenticated.</p>

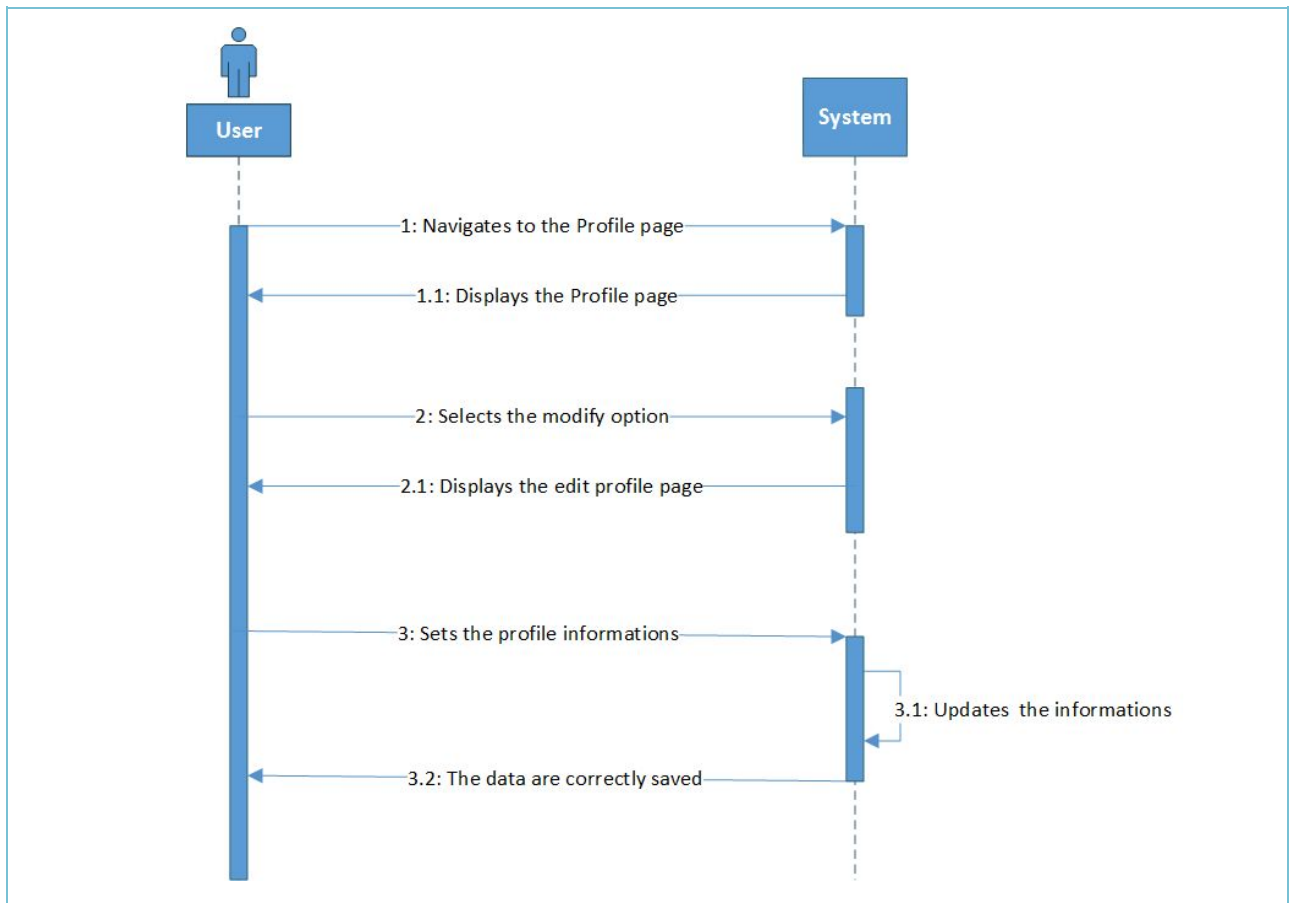
Requirement	<p>[FR 7.1] The system must check if the QR code of the Driver matches with the one allocated to him/her during the reservation process.</p> <p>[FR 7.2] The system must check the QR code of all the passengers to apply a price reduction of 10%, if there are at least two other passengers into the car.</p> <p>[FR 7.3] The system must allow the User to power up the car just pushing the “Start and Stop” button, if the User has been already authenticated.</p>
Assumption	<p>The User is registered.</p> <p>The User has logged in the system.</p>
Actors	User
Entry Condition	The User has opened the car with the associated code and he is entered in his personal area.
Exit condition	The car recognizes the QR Code and authorized the User powering up the car.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User takes the QR code in the reservation area.</li> <li>2. The User selects the starting option on the display of the car.</li> <li>3. The User places the QR code in front of the display.</li> <li>4. The User also inserts the QR codes of passengers.</li> <li>5. The System authorizes the User to power up the car by the “Start and Stop Button”.</li> </ol>	
Exception	
Non Functional Requirement	
Sequence Diagram	



### 3.2.3.2.5 Modify profile information

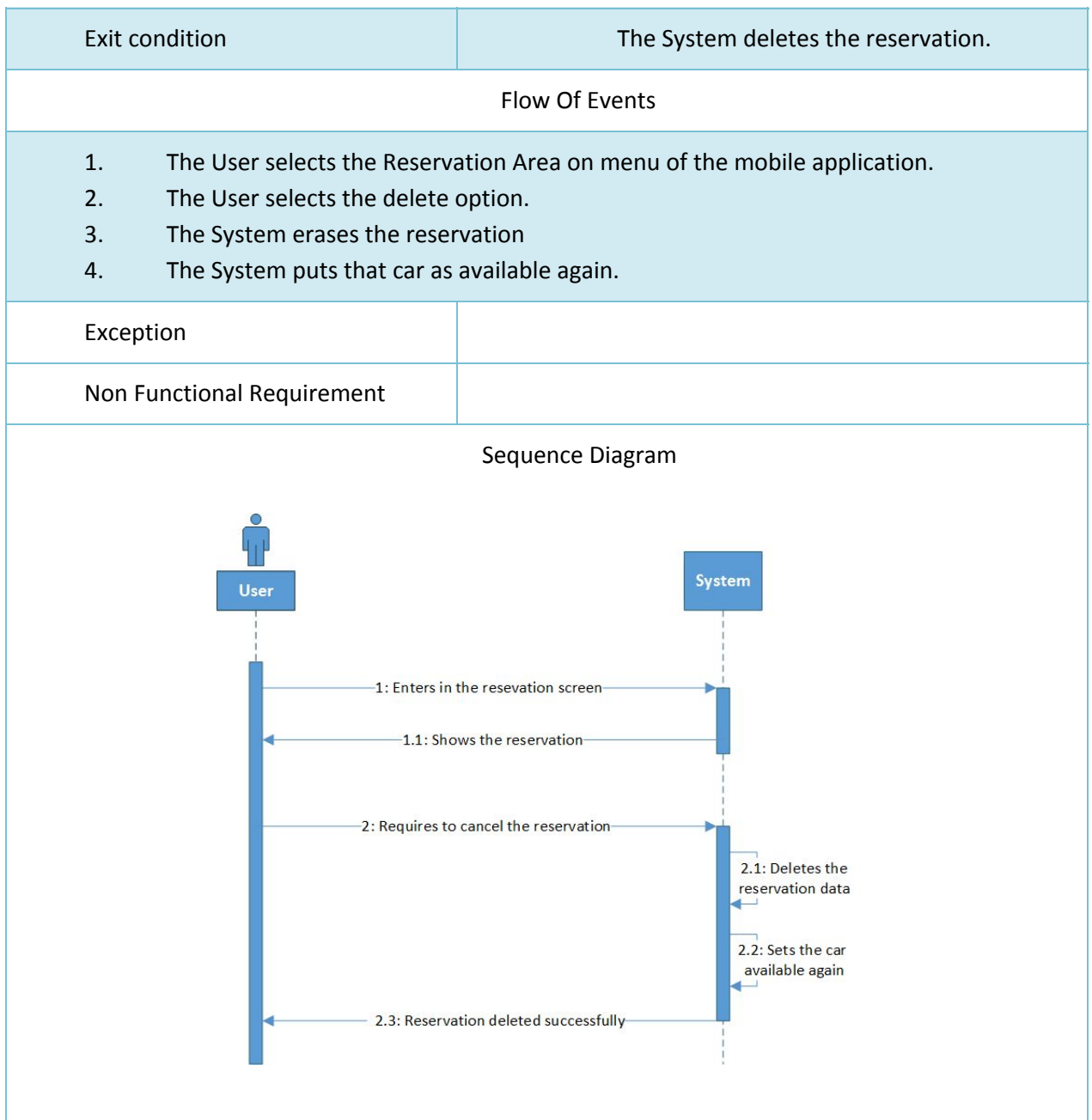
Modify profile information	
Code	UC6
Goal	[G16] Allows the User to modify their profile information.
Requirement	[FR 16] The System must enable the User to change its personal data, that could be obsolete.
Assumption	The User is registered. The User has logged in.

	The User is navigating the mobile application "Power Enjoy".
Actors	User
Entry Condition	The User has logged and navigates to the Profile page.
Exit condition	All the modified informations are saved.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User selects the profile in the homepage of the mobile application.</li> <li>2. The User selects the modify option in the page.</li> <li>3. The System provides the profile information form.</li> <li>4. The User changes the inputs information.</li> <li>5. The System verifies the correctness.</li> <li>6. The User saves them.</li> </ol>	
Exception	<ul style="list-style-type: none"> <li>- The User inserts wrong informations in mandatory fields.</li> <li>- The User does not change anything.</li> </ul>
Non Functional Requirement	
Sequence Diagram	



#### 3.2.3.2.6 Delete a reservation

Delete a reservation	
Code	UC7
Goal	[G15] Allows the User to delete a reservation.
Requirement	[FR 15] After the reservation, the system must permit the User to delete it.
Assumption	The User is registered. The User has logged in. The User has reserved a car.
Actors	User
Entry Condition	The User navigates in the personal Reservation area to see the details of his/her reservation.



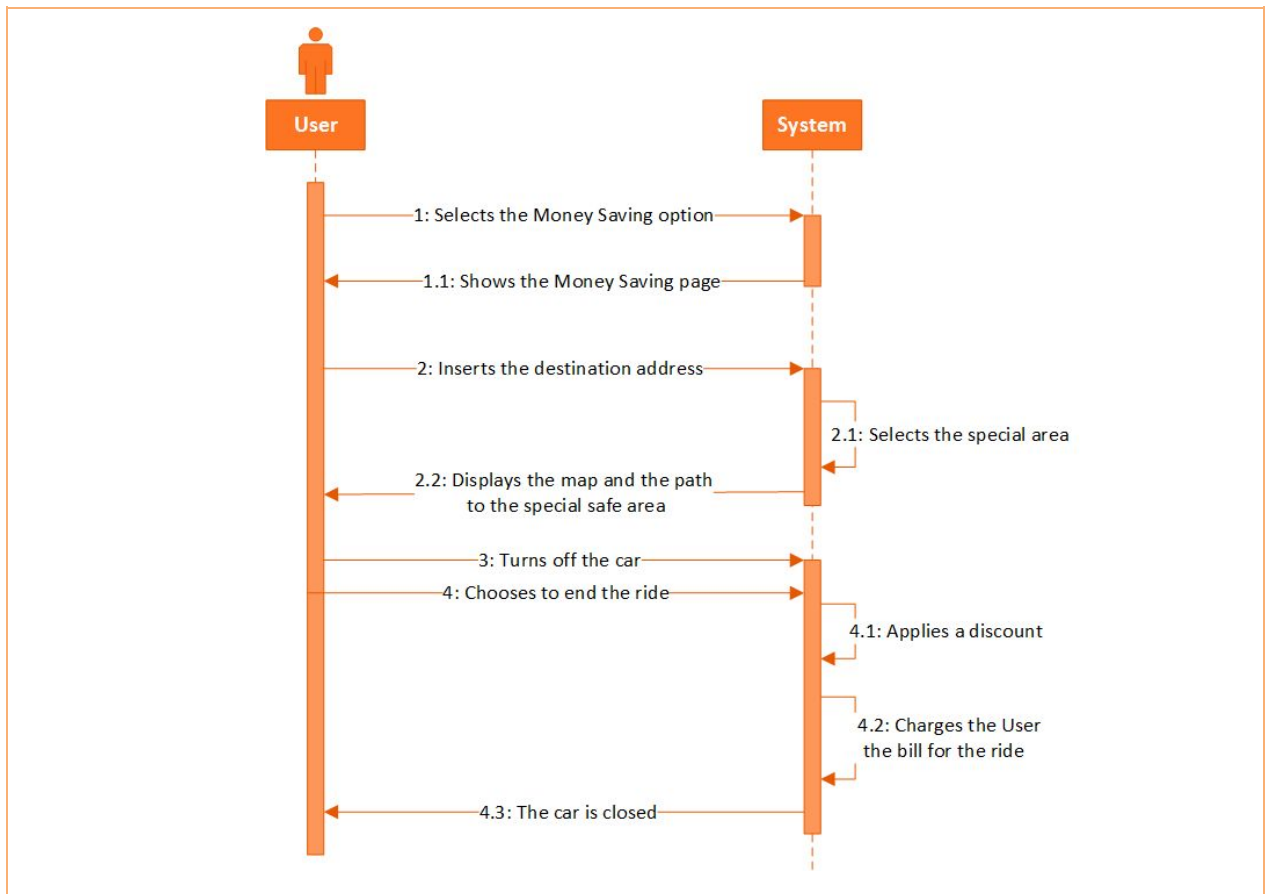
### 3.2.3.2.7 End a ride selecting Money Saving

Select money save and end a ride	
Code	UC8
Goal	[G8] Charges the User from the moment he/she starts up the engine for a given amount of money



	<p>per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.</p> <p>[G9] Notifies the User of the current battery level of the car real time.</p> <p>[G11] Allows the User to locate the nearest safe areas.</p> <p>[G12] Locks automatically when the User closes the car door, after exiting the vehicle.</p>
Requirement	<p>[FR 8.1] When the User leaves the car, the system must ask the User if he wants to stop or to end the ride.</p> <p>[FR 8.2] The system has to show on the screen the current charges real time.</p> <p>[FR 8.3] When the User ends the ride the system must charge him for a given amount of money per minute, applying the related discounts and penalties.</p> <p>[FR 9] The system must show the current charge of the car battery on the screen of the on-board computer.</p> <p>[FR 11.1] The system has to show on the map of the on-board device the safe areas available, as long as the User is driving within a certain distance from them.</p> <p>[FR 11.2] In case of emergency, the system has to notify immediately the problem to the User and to point him/her the nearest safe areas where to stop and let the car be repaired.</p> <p>[FR 11.3] If the User chooses the “Money Saving Mode”, the system must show, on the screen of the on-board device, all the safe areas in proximity of the Driver’s route, until the “Money Saving Mode” is enabled.</p> <p>[FR 12] The system must lock the car after the User has expressed the will of ending the ride, has left the vehicle and has closed the doors.</p>
Assumption	<p>The User is registered.</p> <p>The User has logged in.</p> <p>The User has reserved a car.</p>

	<p>The User has opened the car.</p> <p>The User powered up the car.</p>
Actors	User
Entry Condition	The User selects money saving option on the display of the car.
Exit condition	The User ends the ride in the indicated safe area.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User selects “Money Saving” option on the display of the car.</li> <li>2. The User inserts the destination address.</li> <li>3. The System calculates the safe area nearest to the destination of the User and more useful to ensure a uniform distribution of cars.</li> <li>4. The System shows the special safe area where the User can park.</li> <li>5. The User reaches the area and ends the ride.</li> <li>6. The System applies a discount on the final charge.</li> <li>7. The System locks the car.</li> <li>8. The System charge the user on his/her payment method.</li> </ol>	
Exception	<p>-The User cancels the operation.</p> <p>-The User does not reach the selected area.</p> <p>-The User does not have enough money on his/her payment method.</p>
Non Functional Requirement	
Sequence Diagram	

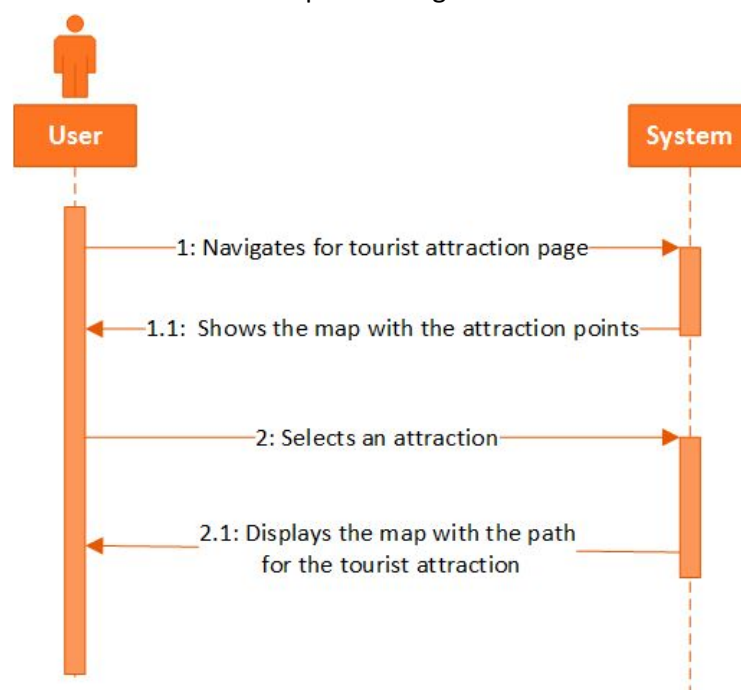


### 3.2.3.2.8 Select tourist attraction and going to an address

Select tourist attraction and go to an address	
Code	UC9
Goal	[G10] If required, allows the User to receive informations about the main tourist attractions in proximity, while driving.
Requirement	<p>[FR 10.1] The system must show on the map of the on-board device the most fascinating and popular tourist attractions, as museums, exhibitions, monuments or buildings interesting from an architectural point of view, while he/she is moving in proximity of them.</p> <p>[FR 10.2] If the User clicks on an attraction, the system has to display a brief description of it.</p>
Assumption	The User is registered.

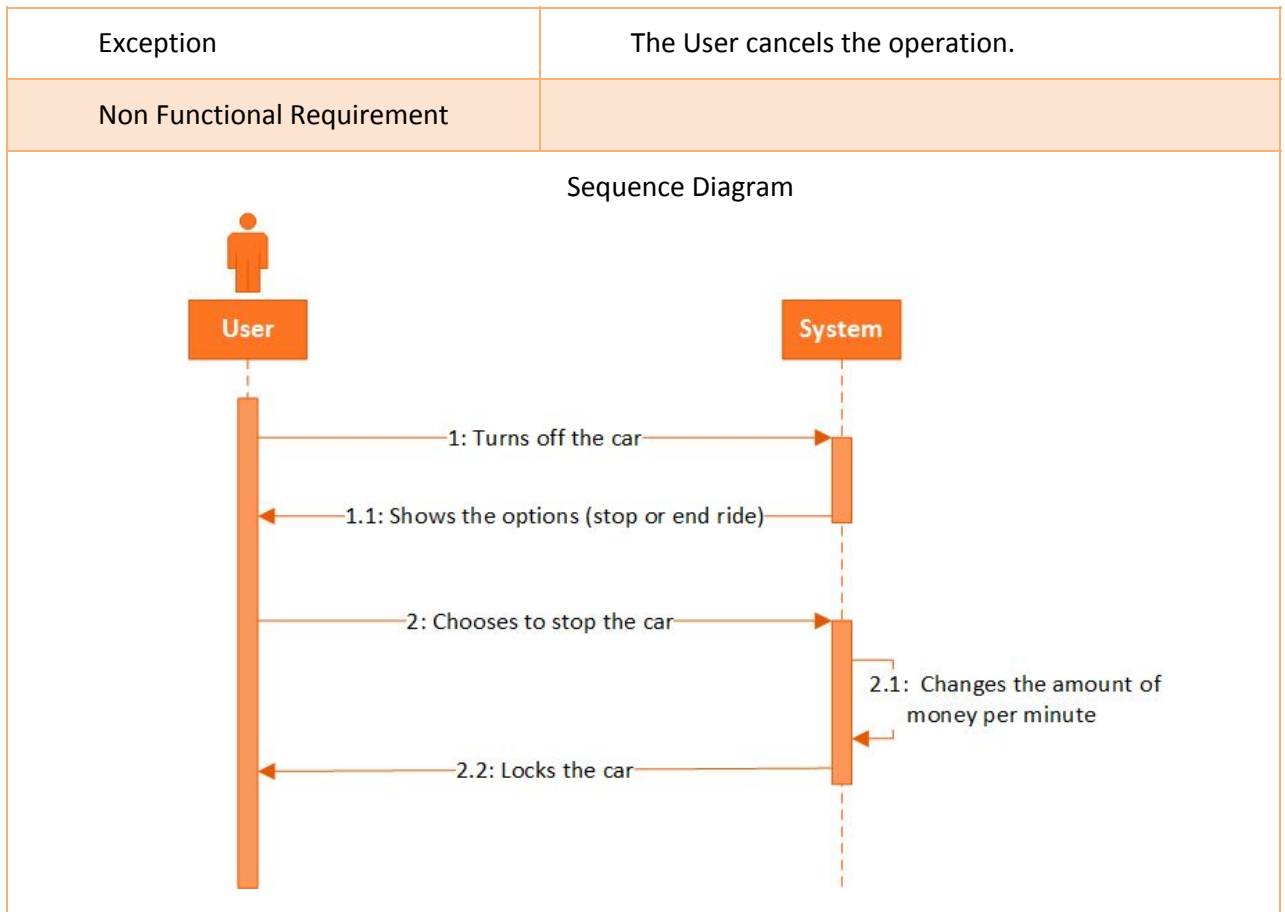
	<p>The User has logged in.</p> <p>The User has reserved a car.</p> <p>The User has opened the car.</p> <p>The User powered up the car.</p>
Actors	User
Entry Condition	The User selects Tourist attraction on the display of the car and then select the navigator to an address.
Exit condition	The User arrived to the address.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User select tourist attraction on the display of the car.</li> <li>2. The System shows the map with the attraction points.</li> <li>3. The User chooses a point on the map.</li> <li>4. The System start the navigator to the selected address.</li> <li>5. The User arrives to the address.</li> </ol>	
Exception	The User cancels the navigator option.
Non Functional Requirement	[NFR 10.1] The system must update its maps every day.

Sequence Diagram

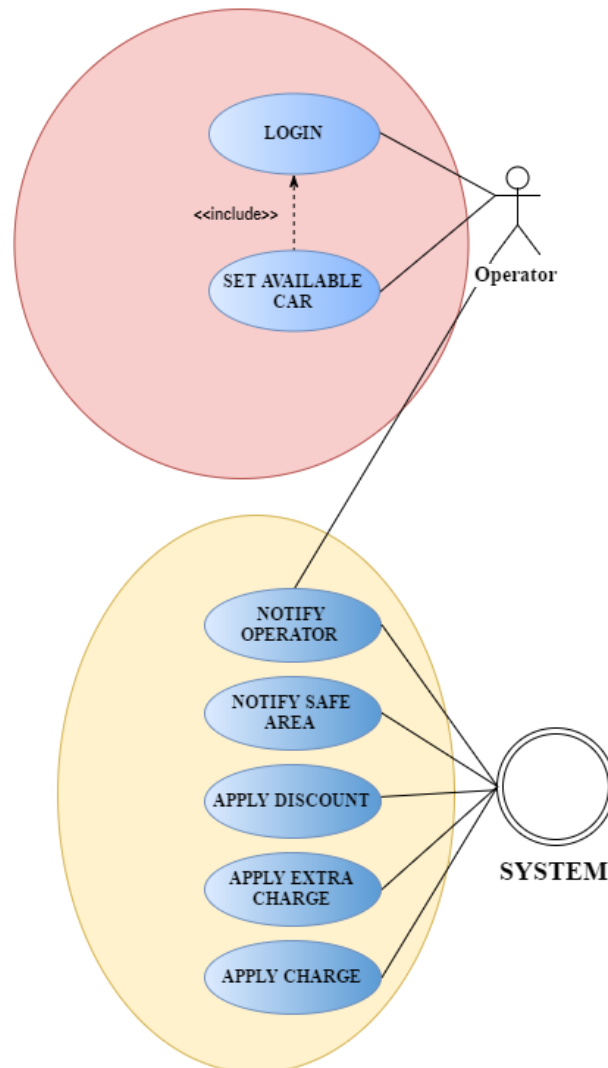


### 3.2.3.2.9 Stop a ride temporarily

Stop a car during a ride	
Code	UC10
Goal	<p>[G12] Locks automatically when the User closes the car door, after exiting the vehicle.</p> <p>[G8] Charges the User from the moment he/she starts up the engine for a given amount of money per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.</p>
Requirement	<p>[FR 12] The system must lock the car after the User has expressed the will of ending or stopping temporarily the ride, has left the vehicle and has closed the doors.</p> <p>[FR 8.1] When the User leaves the car, the system must ask the User if he wants to stop or to end the ride.</p>
Assumption	<p>The User is registered.</p> <p>The User has logged in.</p> <p>The User has reserved a car.</p> <p>The User has opened the car.</p> <p>The User powered up the car.</p>
Actors	User
Entry Condition	The User stops and select the stop option on the display of the car.
Exit condition	The System locks the car.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User stops the car.</li> <li>2. The User select the stop option on the display of the car.</li> <li>3. The User exits the car.</li> <li>4. The System locks the car.</li> </ol>	



### 3.2.3.3 Use Case : System and Operator



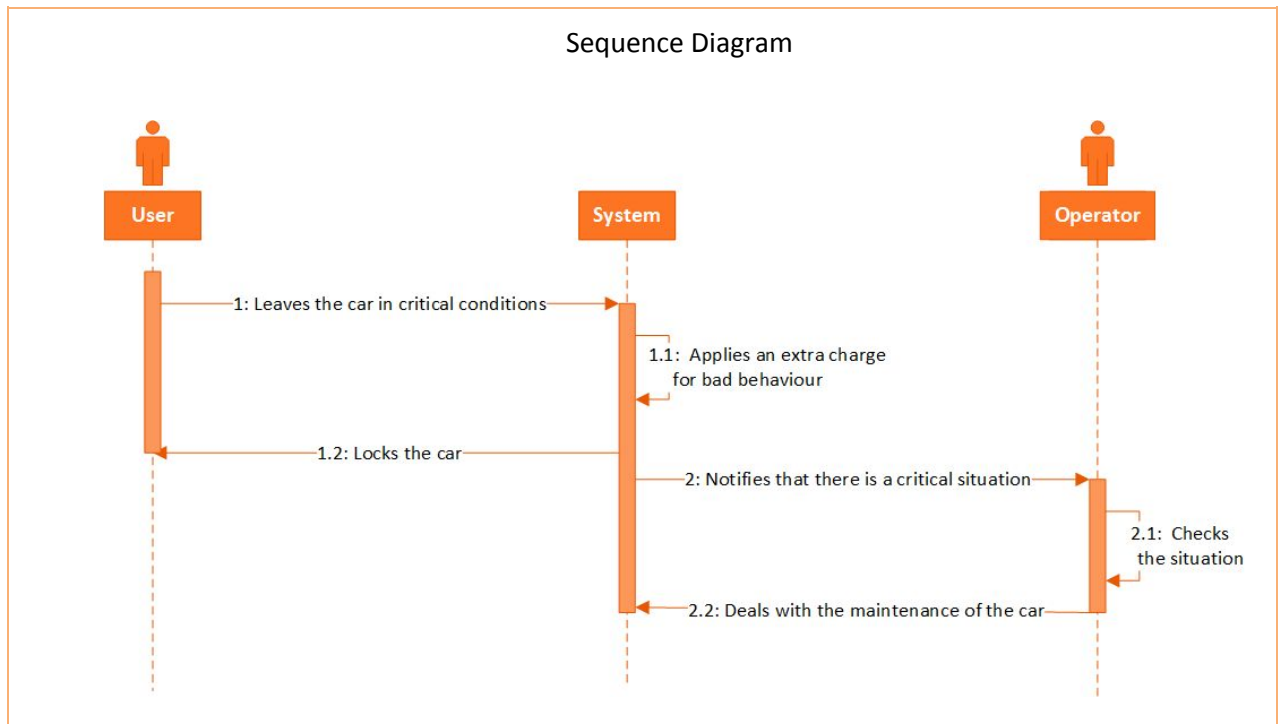
#### Use Case 4: System and Operator

##### 3.2.3.3.1 Notify an operator after a critical situation

Notify an operator for recovering a car	
Code	UC11
Goal	[G13] Allows operators to know exactly when cars need to be recharged or an intervention of different nature.

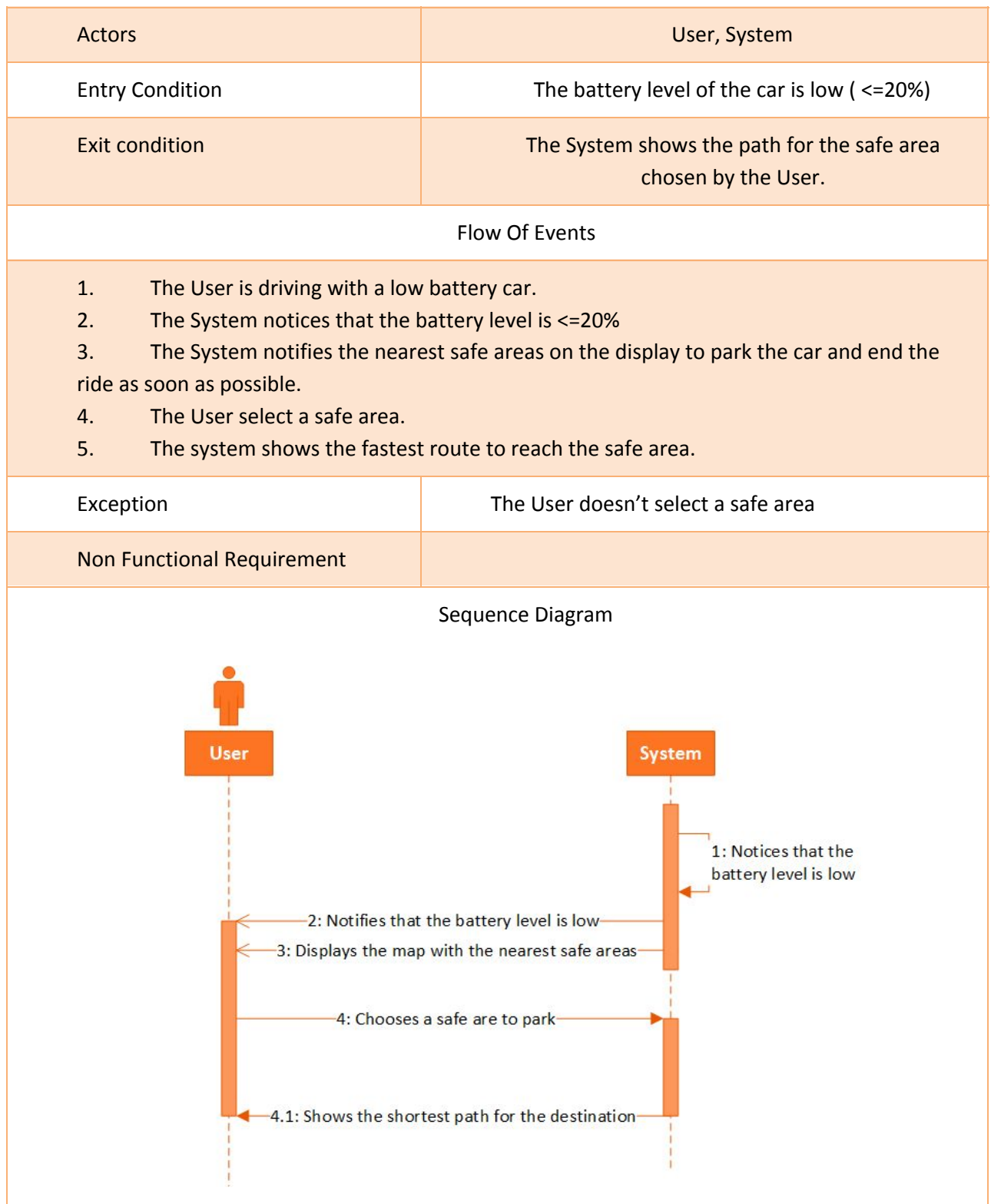
	[G18] Penalize the User for a improper behaviour with additional charges.
Requirement	<p>[FR 13.1] The system must monitor the condition of the car, in order to be able to see if the car needs to be charged or repaired.</p> <p>[FR 13.2] The system has to send a request for intervention to the operator to whom was assigned the area in which the car is.</p> <p>[FR 18.1] The system must apply a penalty of 30%, if the car is left with less than 20% of battery.</p> <p>[FR 18.2] The system must apply a penalty of 30%, if the car is left at more than 3 km from the nearest station.</p> <p>[FR 18.3] The system must charge the associated User for every exceptional penalty due to that User.</p> <p>[FR 8.3] When the User ends the ride the system must charge him for a given amount of money per minute, applying the related discounts and penalties.</p>
Assumption	The User ended a ride and the car has a low charge or the User ended a ride outside a safe area.
Actors	User, System, Operator
Entry Condition	The User leaves the car in an outside area or with a low charge.
Exit condition	The Operator has dealt with the maintenance of the car.
Flow Of Events	
<ol style="list-style-type: none"> <li>1. The User has left a reserved car in critical conditions.</li> <li>2. The System apply an extra charge for bad behaviour to the User.</li> <li>3. The System notify the associated operator for recover the car.</li> <li>4. The Operator receives the notification and checks the situation.</li> <li>5. The Operator deal with the maintenance of the car.</li> </ol>	
Exception	The notification fails.
Non Functional Requirement	





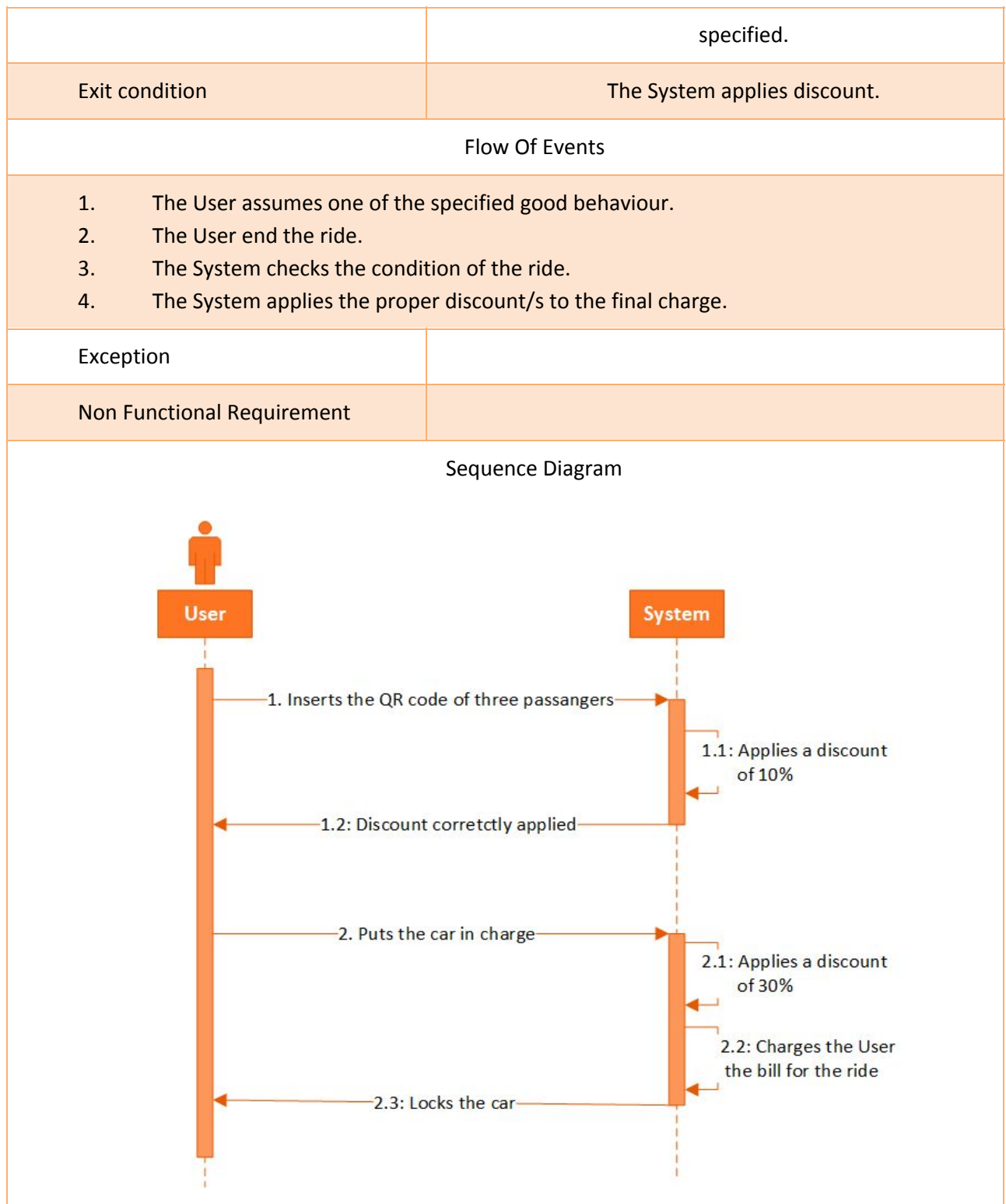
### 3.2.3.3.2 Notify the user of the nearest safe area because of the low battery of the car

Notify a User the nearest safe area	
Code	UC14
Goal	[G9] Notifies the User of the current battery level of the car real time.
Requirement	<p>[FR 9.2] The System must show an alarm when the battery is particularly low, and show the nearest safe areas from the current position of the car.</p> <p>[FR 11.2] In case of emergency, the system has to notify immediately the problem to the User and to point him/her the nearest safe areas where to stop and let the car be repaired.</p>
Assumption	<p>The User has reserved a car.</p> <p>The User entered the car.</p> <p>The User powered up the car.</p> <p>The car has a low level of battery.</p>

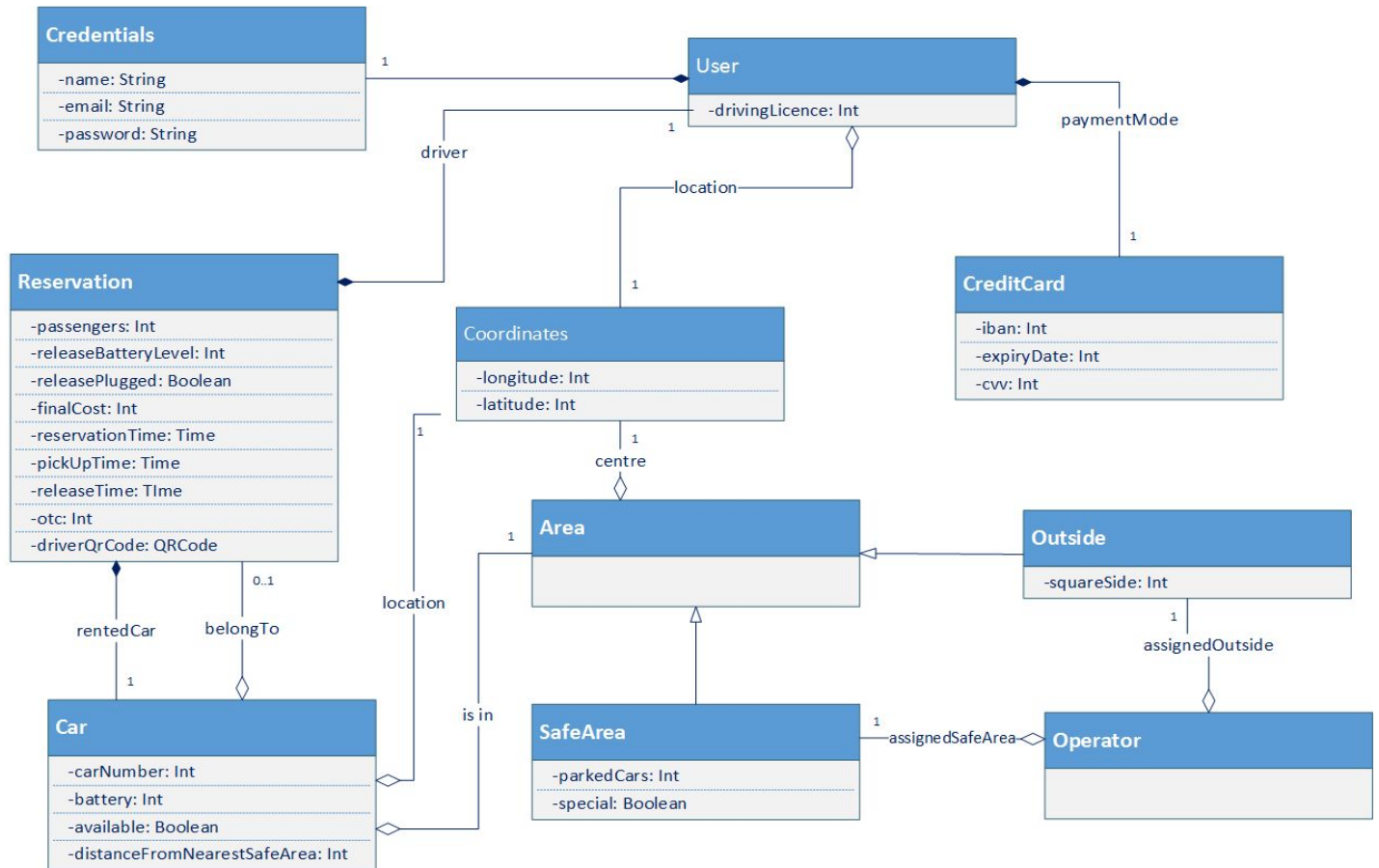


#### 3.2.3.3.4 Apply discounts

Apply discounts	
Code	UC15
Goal	<p>[G17] Reward the User for a good behaviour with discounts.</p> <p>[G8] Charges the User from the moment he/she starts up the engine for a given amount of money per minute, up to the moment when he exits the vehicle and also notifies the User of the current charges real time.</p>
Requirement	<p>[FR 17.1] The system must apply a discount of 10%, if the passengers are more or equal than 3.</p> <p>[FR 17.2] The system must apply a discount of 20%, if the car is left with more than 50% of battery.</p> <p>[FR 17.3] The system must apply a discount of 30%, if the User brings the car in a safe area and puts it in charge.</p> <p>[FR 17.4] If the User turns on the “Money Saving” option, the system must indicate to him/her specific safe areas where to park, in order to receive a discount.</p> <p>[FR 8.3] When the User ends the ride the system must charge him for a given amount of money per minute, applying the related discounts and penalties.</p>
Assumption	<p>The User has reserved a car.</p> <p>The User has opened the car.</p> <p>The User has powered up the car.</p> <p>The User assumes one of the good behaviour specified.</p>
Actors	User, System
Entry Condition	The User assumes one of the good behaviour

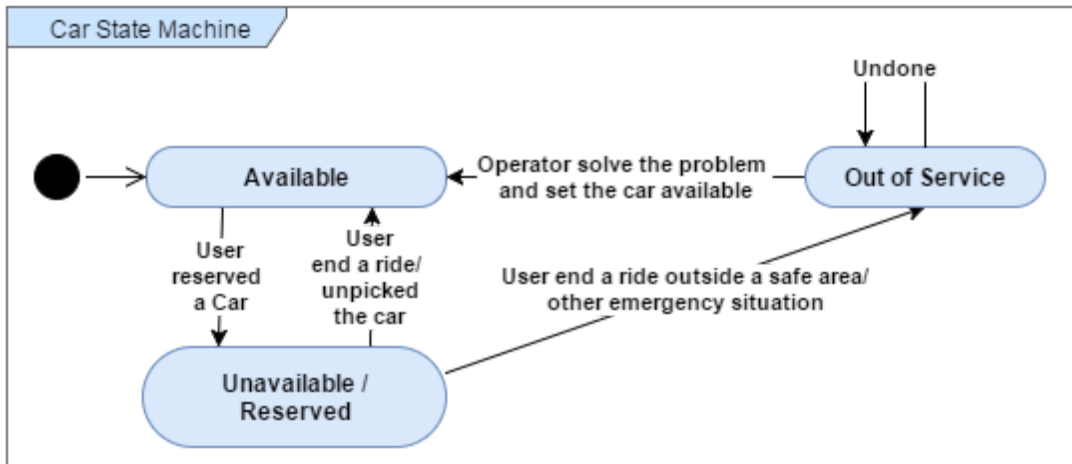


### 3.3 Class Diagram



**Class Diagram: System UML**

## 3.4 Statechart Diagram



State Chart Diagram: State of the vehicles

## 3.5 Software System Attributes

### 3.4.1 Reliability

The system must ensure the truthfulness of the data shown to the Users and effectiveness of the reservations.

### 3.4.2 Availability

The System should be available in every time 24 hours a day, 7 days a week.

### 3.4.3 Security

The System must provide the secure storage of Users' passwords and personal information, like payments methods, and also must protect payments transactions. This can be achieve with secure protocols.

### 3.4.4 Maintainability

All data about Users must be stored consistently in the DB, and also, the System has to provide durability properties. This should be achieved with backup operations.

### 3.4.5 Portability

This software must run in all the systems.

## 3.6 Other Requirements

The interfaces have to be user-friendly and suitable for every kind of User. They must maximize the number of functionalities and interactions between Users and System, by minimizing the number of clicks necessary to the employment. They have to totally avoid misleading fulfillments.

# 4. Appendices

## 4.1 Alloy Model

### 4.1.1 Signatures

```
//User
sig User {
  credentials: one Credentials,
  drivingLicence: one Int,
  location: one Coordinates,
  paymentMode: one CreditCard,
}
drivingLicence > 0

//Reservation
sig Reservation{
  driver: one User,
  passengers: one Int,
  rentedCar: one Car,
  reservationTime: one Time,
  pickUpTime: one Time,
  releaseTime: one Time,
  releaseBatteryLevel: one Int,
  releasePlugged: one Bool,
  finalCost: one Int,
  otc: one Int,
  driverQRCode: one QRCode
}
passengers >= 0 and passengers <= 4
releaseBatteryLevel >= 0 and releaseBatteryLevel <= 100
finalCost > 0
}
```

```

//Car
sig Car{
  carNumber: one Int,
  isIn: one Area,
  location: one Coordinates,
  belongTo: lone Reservation,
  available: Bool,
  battery: one Int,
  distanceFromNearestSafeArea: one Int //kilometres
}{
  carNumber >= 0 and
  battery >= 0 and
  battery <= 100
  distanceFromNearestSafeArea > 0
}

//Area
abstract sig Area{
  centre: one Coordinates,
}

sig SafeArea extends Area{
  parkedCars: one Int,
  special: one Bool
}{
  parkedCars > 0
}

sig Outside extends Area{
  squareSide: one Int
}{
  squareSide > 0
}

//Operator
sig Operator{
  assignedSafeArea: one SafeArea,
  assignedOutside: one Outside
}

//Coordinates
sig Coordinates{
  longitude: one Int,
  latitude: one Int
}{
  latitude >= -9 and latitude <= 9
}

```



```
    longitude >= 0 and longitude <= 18
}
```

```
//Credit card
sig CreditCard{
    iban: one Int,
    expiryDate: one Int,
    cvv: one Int
}{
    iban > 0
    cvv > 0
    expiryDate > 0
}
```

```
//Time
sig Time{
    hour: one Int,
    minute: one Int
}{
    hour >= 00 and hour <= 23 and
    minute >= 00 and minute <= 59
}
```

```
sig CurrentTime extends Time{}
```

```
//Credentials
sig Credentials{
}
```

```
//QR code
sig QRCode{
}
```

## 4.1.2 Facts

```
//Singleton current time
fact singletonCurrentTime{
    #CurrentTime = 1
}
```

```
//An OTC code corresponds only to one reservation
fact oneTimeCode{
    all r1, r2: Reservation |
        (r1 != r2)
        implies r1.otc != r2.otc
}
```

```

}

//A driver's qr code must be different for two different reservations
fact differentQrCodes{
  all r1, r2: Reservation |
    (r1 != r2)
    implies (r1.driverQrCode != r2.driverQrCode)
}

//A car reserved corresponds just to a certain reservation
fact carLinkedToReservation{
  no c:Car, r:Reservation |
    (c.belongTo=r
      and r.rentedCar !=c)
}

//Different cars can be reserved simultaneously only by different users
fact uniqueCarReservation{
  all c1,c2: Car |
    (c1!=c2)
    implies c1.belongTo != c2.belongTo
}

//Can not exist two simultaneous reservations with the same driver or the same car
fact uniqueReservation{
  all r1,r2 :Reservation |
    (r1 !=r2)
    and (r1.driver = r2.driver
      or r1.rentedCar = r2.rentedCar)
    implies
      (r1.reservationTime != r2.reservationTime
        and r1.pickUpTime != r2.pickUpTime
        and r1.releaseTime != r2.releaseTime)
}

//Different credit cards have different ibans
fact differentCreditCards{
  all c1,c2 : CreditCard |
    (c1!=c2)
    iff (c1.iban != c2.iban)
}

//If a car is in an area, it has the same location of the area
fact carInAnArea{
  all c:Car, a:Area |
    (c.isIn = a)
    implies (c.location =a.centre)
}

```

```

}

//If there is a reservation with a certain car, that car is not available
fact carReserved {
  all r: Reservation, c: Car |
    (r.rentedCar = c)
    implies (c.available=False)
}

//Users must have different credentials and driving licences
fact differentCredentials{
  all u1, u2: User |
    (u1!=u2)
    implies ( (u1.credentials != u2.credentials)
    and (u1.drivingLicence != u2.drivingLicence))
}

//Not available car
fact notAvailableCar{
  all c: Car |
    (c.battery<20
    or c.distanceFromNearestSafeArea>3)
    implies (c.available= False)
}

//Cars with different ID
fact differentNumberOfCar{
  all c1,c2: Car |
    (c1!=c2)
    implies(c1.carNumber != c2.carNumber)
}

//The area assigned to an operator can not be assigned to an other one
fact OneOperatorPerArea{
all o1, o2:Operator | (o1!=o2)
implies ((o1.assignedSafeArea centre != o2.assignedSafeArea.centre)
    and (o1.assignedOutside.centre != o2.assignedOutside.centre)
    and (o1.assignedOutside.squareSide!=o2.assignedOutside.squareSide))
}

//Cars with some necessities are recovered by the operator responsible for the area in which the car is
fact RecoverOperationOutside{
  all o:Operator, c:Car, out:Outside |
    (c.battery<=20 or c.isIn=out)
    implies (o.assignedOutside=c.isIn)
}

```

```

fact RecoverOperationSafeArea{
  all o:Operator, c:Car, sa:SafeArea |
    (c.battery<=20 or c.isIn=sa)
    implies (o.assignedSafeArea=c.isIn)
}

//Each operator has a different area to take after
fact uniqueOperator{
  all o1,o2:Operator |
    (o1!=o2)
    implies ((o1.assignedSafeArea != o2.assignedSafeArea)
      and (o1.assignedOutside != o2.assignedOutside))
}

//Each area has only one operator
fact oneOperatorPerArea{
  one o: Operator |
    all sa: SafeArea, out: Outside |
      (o.assignedSafeArea = sa
        and o.assignedOutside = out)
}

//Two different safe areas have different coordinates
fact differentSafeAreas{
  all sa1, sa2: SafeArea |
    (sa1 != sa2)
    iff (sa1.centre != sa2.centre)
}

//Two different outside areas have different coordinates
fact differentOutsides{
  all out1, out2: Outside |
    (out1 != out2)
    iff (out1.centre != out2.centre)
}

//Every outside is linked to a save area
fact outsideLinkedToSafeArea{
  all o: Operator |
    o.assignedOutside.centre = o.assignedSafeArea.centre
}

//Pickup time has to be major than reservation time
fact reservationPickUpConstraints1{
  all r: Reservation |

```

```

        (r.reservationTime.hour < r.pickUpTime.hour)
        or ((r.reservationTime.hour = r.pickUpTime.hour)
            and (r.reservationTime.minute < r.pickUpTime.minute))
    }

    //Time for the reservation
    fact atMostOneHourOfReservation{
        all r: Reservation |
            (r.pickUpTime.hour < r.reservationTime.hour+1)
            or ((r.pickUpTime.hour = r.reservationTime.hour +1)
                and (r.pickUpTime.minute < r.reservationTime.minute))
    }

    //Release time must major than pickup time
    fact pickupBeforeRelease{
        all r: Reservation |
            (r.pickUpTime.hour < r.releaseTime.hour)
            or ((r.pickUpTime.hour = r.releaseTime.hour)
                and (r.pickUpTime.minute < r.releaseTime.minute))
    }

```

### 4.1.3 Assertions

```

    //Checks that a car already reserved is set as unavailable
    assert availability{
        all ct: CurrentTime, t: Time, c: Car, r: Reservation |
            (r.rentedCar = c)
            and (r.reservationTime = t)
            and moreThanOrEqual[ct, t]
            implies c.available = False
    }

    //Checks that a car whose reservation exceeds the prearranged time of one hour, is automatically set as
    available again, when the time expires
    assert reservationExpired{
        all r: Reservation, ct: CurrentTime | moreOrEqualToOneHour[r.reservationTime, ct]
        implies
            r.rentedCar.available = True
    }

    //Checks that different cars do not belong to the same reservation
    assert noUbiquity{
        no disj c1, c2 : Car |
        all r: Reservation |
            c1.belongTo = r
            and c2.belongTo = r
    }

```

```

}

//Checks that a car which is distant more than 3 kilometres from the nearest safe area or whose battery is
lower than 20% can not be reserved
assert notAvailable{
  all c: Car | c.battery < 20 or c.distanceFromNearestSafeArea > 3
    implies c.available = False
}

```

#### 4.1.4 Predicates

```

pred moreThanOrEqual [t1, t2: Time]{
  t1.hour = t2.hour
  implies t1.minute >= t2.minute
  else t1.hour >= t2.hour
}

pred moreOrEqualToOneHour[t1, t2: Time]{
  (t2.hour = t1.hour+1) and (t1.minute>=t2.minute)
  or (t2.hour > t1.hour+1)
}

```

```

pred show {
  #User = 1
  #Reservation =1
  #Car = 1
}

```

```

pred showTwoUsers{
  #User = 2
  #Reservation = 2
  #Car=2
}

```

```

pred showConflict{
  #User = 2
  #Reservation = 2
  #Car=1
}

```

```

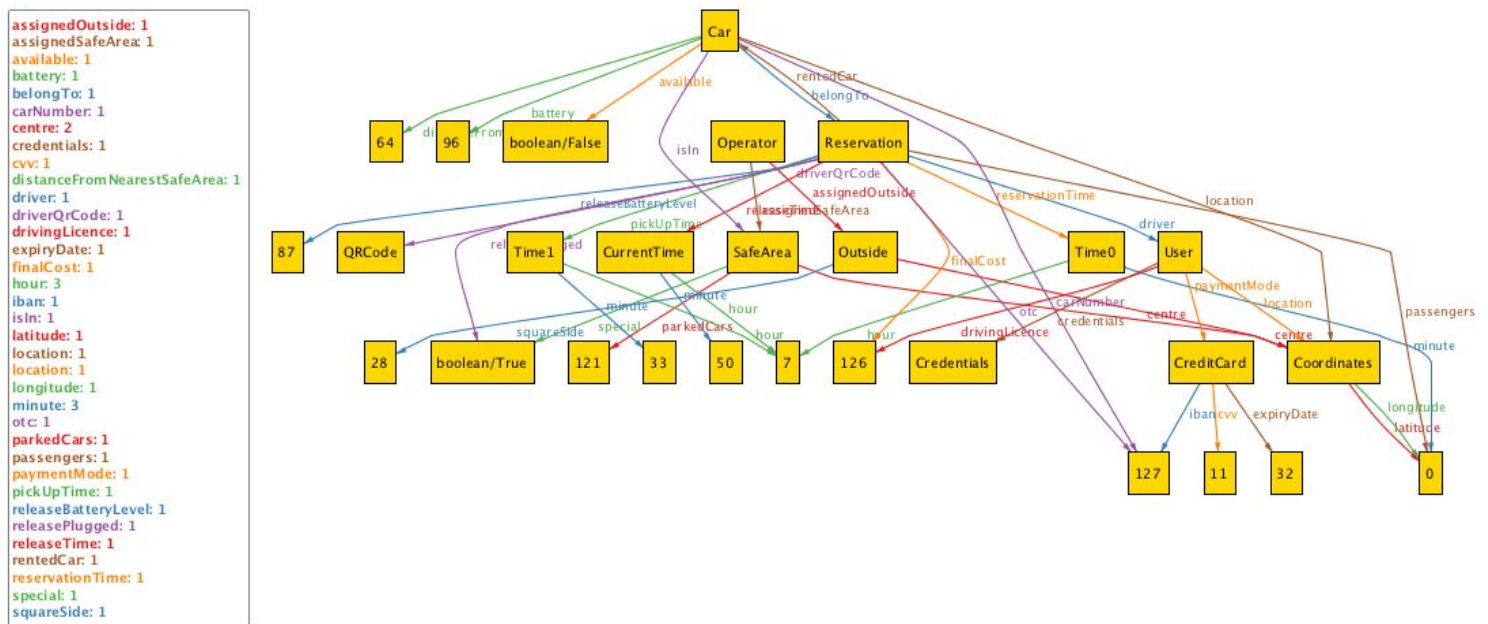
//check availability for 8 Int
//check reservationExpired for 8 Int
//check noUbiquity for 8 Int

```

```
//check notAvailable for 8 Int
run show for 8 Int
//run showTwoUsers for 8 Int
//run showConflict for 8 Int
```

#### 4.1.4 Graphical Representation

One of the possible graphical representations of the alloy model previously described:



#### Alloy Graphical Representation

## 5. Used Tools

The main tools we used to create this RASD document are:

- Google Drive Documents: to write and redact this document;
- Visio: to create the diagrams;
- Draw.io : to create diagrams;
- Alloy Analyzer 4.2: to prove the consistency of our model.

## 6. Effort Spent

### 6.1 Giulia Leonardi

➤	30/10/2016	4h
➤	04/11/2016	3h
➤	05/11/2016	5h
➤	06/11/2016	3h
➤	07/11/2016	4h
➤	09/11/2016	5h
➤	10/11/2016	5h
➤	11/11/2016	5h
➤	12/11/2016	8h
➤	13/11/2016	9h

### 6.2 Marzia Degiorgi

➤	30/10/2016	5h
➤	31/10/2016	3h
➤	01/11/2016	4h
➤	04/11/2016	2h
➤	05/11/2016	6h
➤	06/11/2016	7h
➤	07/11/2016	3h
➤	10/11/2016	6h
➤	11/11/2016	4 / 5h
➤	12/11/2016	1h

### 6.3 Valentina Ionata

➤	30/11/2016	3h
➤	01/11/2016	2h
➤	02/11/2016	1h
➤	03/11/2016	4h
➤	05/11/2016	2h
➤	06/11/2016	1h
➤	07/11/2016	3h
➤	08/11/2016	2h
➤	09/11/2016	3h
➤	10/11/2016	3h
➤	11/11/2016	2h
➤	12/11/2016	9h
➤	13/11/2016	7h



## 7. Revision History

Version	Date	Description
v 1.0.0	13/11/2016	Initial Release
v 1.0.1	14/1/2017	Updated layout Added Statechart Diagram Resolved lexical mistakes
v 1.0.2	03/02/2017	Formal updates