

# POLITECNICO MILANO 1863

Politecnico di Milano

A.A. 2016/2017

Software Engineering 2: “*PowerEnJoy*” v. 1.1

**Requirements Analysis and Specification Document**

Matteo Bresich (mat. 774366)

December 5, 2016



# Table of Content

# 1 Introduction

## 1.1 Document Purpose

This document aims to describe in a complete way the application that will be developed and the domain in which it will be performed.

The target audience of this document are analysts, project managers, developers, testers and final users.

**Important!** This document has a contract validity between the costumer and the company that will develop the application.

## 1.2 Product Scope

### 1.2.1 Actual system

Until now the car sharing company has a system where a registered client can reserve an electric car by calling a call-center. The service registration cannot be done on-line. It can only be done in the company office with a specific module. The company has a software called "*CSGestion*" that is a custom ERP, and it's used for managing all company cars. The software stores all the informations into a database.

Data such as position, battery power and occupied seats are read by an on-board computer on every car and transmitted via cellular connection to the software. All data between the software *CSGestion* and each auto are transmitted using a VPN for better security on sensitive data. All customers requests are handled by the call-center of the company. An operator after a client request reserves the auto at most for one hour. The driver have to use an electronic card ( obtained upon registration) for open the car. After the car is unlocked, the driver can turn on the machine by entering the user key (obtained upon registration) in the on-board computer and pushing the ignition button. When the driver leaves the car not in a safe area and gets out the car, can lock and unlock the car using the electronic card (Figure ??). After the use, when the driver gets out the car after leaving it in a safe area, the car will automatically lock. If the driver leaves the car in a safe area the payment of the service is automatically made by the registered client credit card and the car can be used by another customer. For whatever exception the client have to call the call-center. In case of accident it is provided that the car involved warns the system of the accident. A call-center operator will follow the practice and will notify to the company that deals with repairs. To ensure the proper functioning of the system the company hired employees (Handyman Operators) which in collaboration with the call-center recharges low battery cars and move cars from a safe area to another to ensure that in each area there is the minimum number of cars available.



Figure 1: The electronic card obtained upon registration.

### 1.2.2 Scope

*PowerEnJoy* is a new web and native mobile application thought to improve existing car sharing service. In particular *PowerEnJoy* is focused to reduce unnecessary calls to the call-center, to facilitate the work of handyman operators, to simplify user ways to access to the service (for example using the smartphone to lock/unlock the booked car or provide an on-line registration) and advertise the car sharing service.

## 1.3 Definitions, acronyms, and abbreviations

### 1.3.1 Definitions

- Visitor (or Guest): is a non-registered user.
- User (or Registered User, or Client, or Driver): is the user of the application that wants to use the car sharing service.
- Passenger: is a person that use a car of the service that is not a driver. He/She has no obligation to be registered to the service.
- Location: are the GPS coordinates to identify a place. It could be the position of a car.
- Car (or Auto, or Transport): is a machine that belongs to the car sharing company.
- Safe area (or Equipped Parking): is a parking area defined in collaboration with the municipal administration equipped with charging towers for company cars. Every parking is unattended and for safety reasons equipped with CCTV.
- CSGestion: is an existing software used by call-center operators for managing all company cars. It allows to know for every car: car status (Available, Busy or Not working), the percentage of battery charge and the number of seats occupied.
- Lock/Unlock a car: it's means to lock/unlock the car door and deny/allow to a person to get in/out the car.

### 1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- CSGestion: Car Sharing Gestion.
- ERP: Enterprise Resource Planning.
- GPS: Global Positioning System.
- VPN: Virtual Private Network.
- CCTV: Closed Circuit Television.

### 1.3.3 Abbreviations

- App: application.
- $[Gn]$ :  $n^{\text{th}}$  goal.
- $[Dn]$ :  $n^{\text{th}}$  domain assumption.
- $[Rn]$ :  $n^{\text{th}}$  functional requirement.

## 1.4 Identifying Stakeholders

The main direct stakeholders of this project are the Municipality of Milan and the car sharing company that together have funded the renovation of the transport service. When the application will be on-line other stakeholders will be some company employees and final users that will provide a feedback of the new system.

## 1.5 Identifying Actors

- Registered Users: are the final users of the car sharing service. Only they can book a car by calling the call-center or by using the web or mobile app, and use it.
- Passenger: is a registered user that can reserve a number of seats of a shared ride using the *PowerEnJoy* application.
- Handyman Operator: is a company employee that recharges low battery cars and move cars from a safe area to another to ensure that in each area there is the minimum number of cars available.

## 1.6 Goals

- [G1] The System has to encourage the user to have a good behaviour using discounts.
- [G2] The System has to discourage the user to have a bad behaviour using fees.
- [G3] The System has to ensure a good distribution of cars in the territory.
- [G4] The System has to provide the ability to share a ride.
- [G5] The Visitor shall be able to sign-up via app.
- [G6] The Registered User shall be able to log into the service.
- [G7] The Registered User shall be able to reserve a car up to one hour before via app.
- [G8] The Registered User shall be able to remove a reservation before the reservation time runs out.
- [G9] The Registered User shall be able to find the locations of available cars within a certain distance from him.
- [G10] The Registered User shall be able to find the locations of available cars from a specified address.
- [G11] The Registered User shall open/close a reserved car via app when he/she is close to it.
- [G12] The Passenger shall be able to use a car in shared mode.
- [G13] The Handyman Operator shall know the location of the car that has to be charged.
- [G14] The Handyman Operator shall report any problem with a car.

## 1.7 Proposed system

The new system will not replace the old one, it will be integrate with it. The proposed solution will be implemented in a client-server architecture and consists in a software composed by a website and a mobile application. Both parts communicate with a server which runs the business logic and accesses the DBMS. Both parts will be developed from scratch. The web version should be responsive for a better view on mobile devices. The native mobile version will be developed for iOS, Android and Windows Phone.

## 1.8 Reference Documents

- ISO/IEC/IEEE 29148-2011 Systems and software engineering - Life cycle processes - Requirements engineering.

## 1.9 Overview

- Section 1: Introduction, provides a short description of the application to be realized.
- Section 2: Overall Description, focuses on features of the software, constraints and assumptions.
- Section 3: Specific Requirements, provides requirements, typical scenarios and use cases for a more easy to read of software functionalities.

## 2 Overall description

### 2.1 Product perspective

*PowerEnJoy* is a web and mobile application which is integrated with an other existing system. The application does provide some APIs for integration with future projects.

### 2.2 Product functions

The function summary can be taken directly from the section in the first part of this document *Scope* (see section ??) and *Goals* (see section ??).

### 2.3 Constraints

#### 2.3.1 Regulatory Policies & Safety and security

The system must be in accordance with local laws and policies. It must ask the user for the permission to acquire, store and process personal data and web cookies. The system must offer the possibility to delete all personal data.

#### 2.3.2 Hardware limitations

*PowerEnJoy* requires a computer or a smartphone connected to the internet. The mobile app requires the GPS module. The following list highlights some minimal conditions under which the system must operate:

	Requirements for Mobile App
Operating system	Android 4.4.2 or higher, iOS 9 or higher, Windows phone 8 or higher
CPU	dual-core @1.5Ghz
Memory	1GB
Other	3G connection @1,5 Mb/s

	Requirements for Web App
Operating system	Windows, MacOS, Linux
CPU	single-core @3Ghz
Memory	1GB
Other	800x600 resolution

#### 2.3.3 Reliability requirements

The system must have a minimum availability of 99%.

#### 2.3.4 Criticality of the application

The system is not used in critical applications for life.

#### 2.3.5 Interfaces to other applications

*PowerEnJoy* requires to access the Internet, requires Google Maps APIs to provide map visualizations and map-related services and requires *CSGestion* APIs for integrate the new application with the old system.

#### 2.3.6 Parallel operation

*PowerEnJoy* supports parallel operations from different users.

## 2.4 Assumptions

- Cars available and bookable are always working.
- Bookable Cars are always connected to the company VPN.
- Every bookable Car always provides correct data such as: location, battery status and occupied seats.
- A car reservation can be cancelled by the user before start the ride.
- Payments are charged at the end of each ride.
- If the driver and all passenger get out the car and the car location is in a safe area the ride ended.
- Passengers can share a ride payments only if they are registered to the service.
- The division of payment between registered users is by weighted average to reserved seats.
- The old payment system can be changed for the addition of discounts and fees.
- The service is regarded as a metropolitan transport service. It is against the rules to get away for more than 30km from the city center.
- The company's employee is registered to the system at the time of hiring.
- After a ride, the user can connect the auto to the charging system, but it is not required.
- A user who has reserved a car and decided to share the ride have to wait all passengers who have reserved a seat at least until before that the reservation expires.
- The location of safe areas are known and recorded in the system.
- A registered user has only one active reservation at a time: he/she can reserve a car or a number of seats of a shared ride at a time.

## 3 Specific requirements

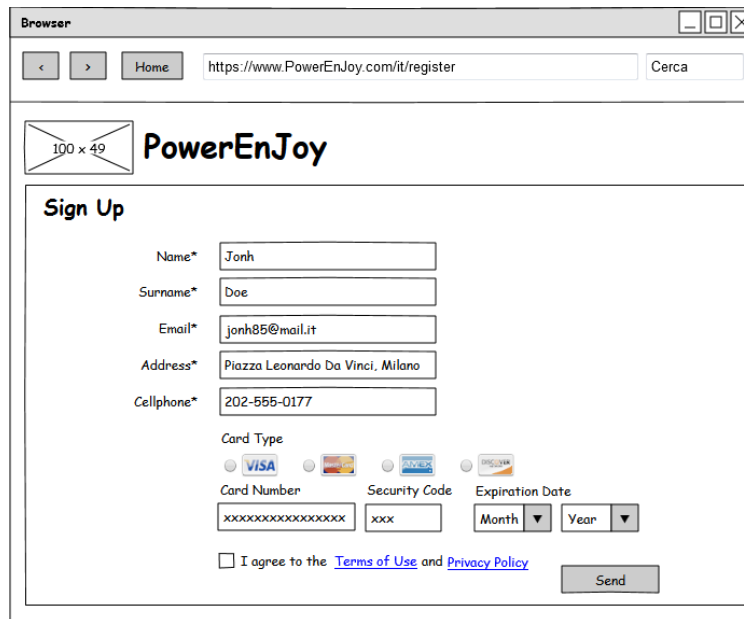
### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

Below there are some mock-ups of the user interface. Mock-ups are only for the main features of the system. The following is only a preview. It is not final and during development it is necessary to work closely with the customer.



**3.1.1.1 Registration** This page presents the registration form for the user.



The screenshot shows a web browser window with the address bar displaying `https://www.PowerEnJoy.com/it/register`. The page features the PowerEnJoy logo and a "Sign Up" section. The form includes fields for Name, Surname, Email, Address, and Cellphone, each with a red asterisk indicating it is required. Below these fields are radio buttons for Card Type (VISA, MasterCard, AMEX, PREPAID) and input fields for Card Number, Security Code, and Expiration Date (Month and Year dropdowns). A checkbox for agreement to Terms of Use and Privacy Policy is present, along with a "Send" button.

Browser

< > Home `https://www.PowerEnJoy.com/it/register` Cerca

100 x 49 **PowerEnJoy**

**Sign Up**

Name\*

Surname\*

Email\*

Address\*

Cellphone\*

Card Type

☒ VISA ☐ MasterCard ☐ AMEX ☐ PREPAID

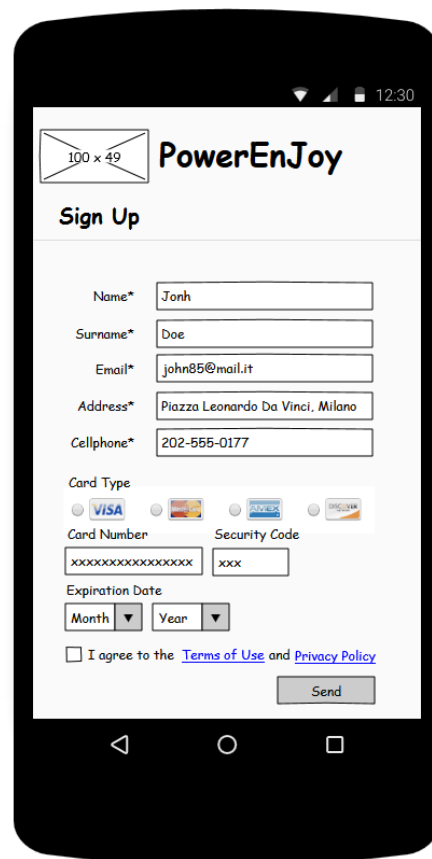
Card Number  Security Code

Expiration Date

Month  Year

☐ I agree to the [Terms of Use](#) and [Privacy Policy](#)

Send



The screenshot shows a mobile phone screen displaying the same PowerEnJoy registration form. The form is adapted for the mobile screen size, with the same fields and layout as the desktop version. The phone's status bar at the top shows the time as 12:30 and various icons. The Android navigation bar is visible at the bottom.

100 x 49 **PowerEnJoy**

**Sign Up**

Name\*

Surname\*

Email\*

Address\*

Cellphone\*

Card Type

☒ VISA ☐ MasterCard ☐ AMEX ☐ PREPAID

Card Number  Security Code

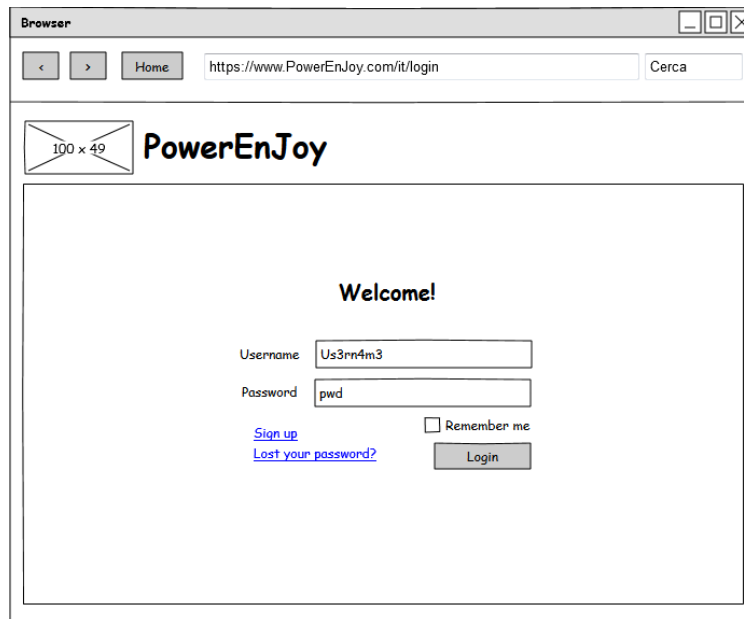
Expiration Date

Month  Year

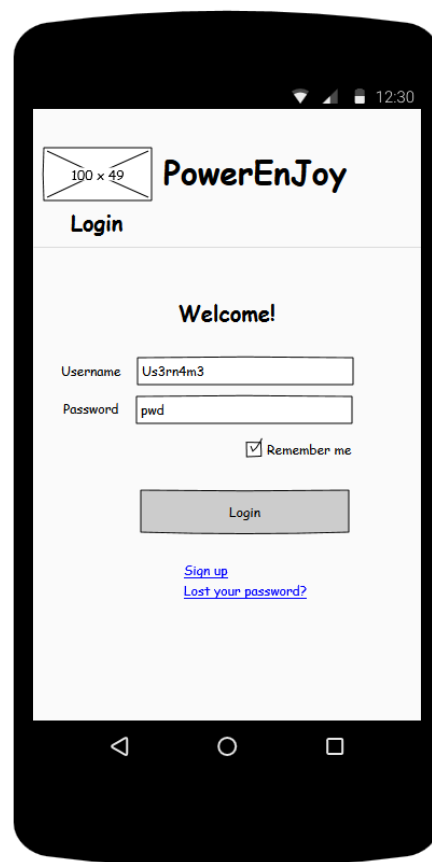
☐ I agree to the [Terms of Use](#) and [Privacy Policy](#)

Send

**3.1.1.2 Login** This page presents the login form for the user.

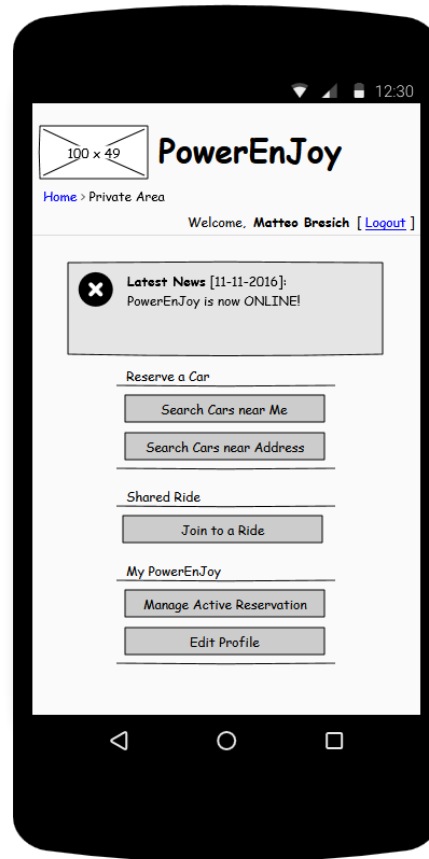
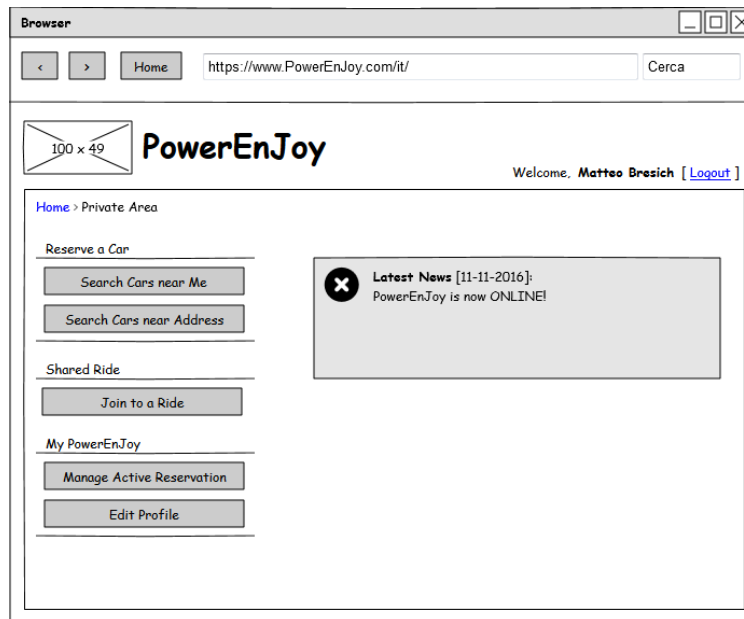


A screenshot of a web browser window titled "Browser". The address bar shows "https://www.PowerEnJoy.com/it/login" with a search button labeled "Cerca". The page features the "PowerEnJoy" logo and a "Welcome!" message. The login form includes fields for "Username" (containing "Us3nn4m3") and "Password" (containing "pwd"). There are links for "Sign up" and "Lost your password?", a "Remember me" checkbox, and a "Login" button.

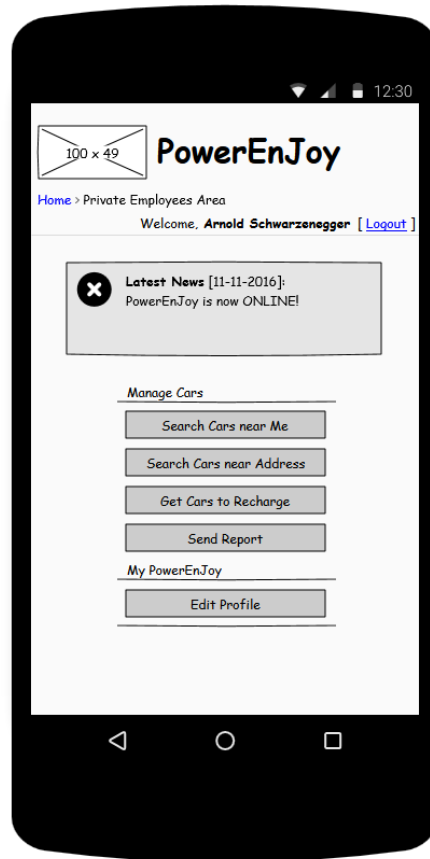
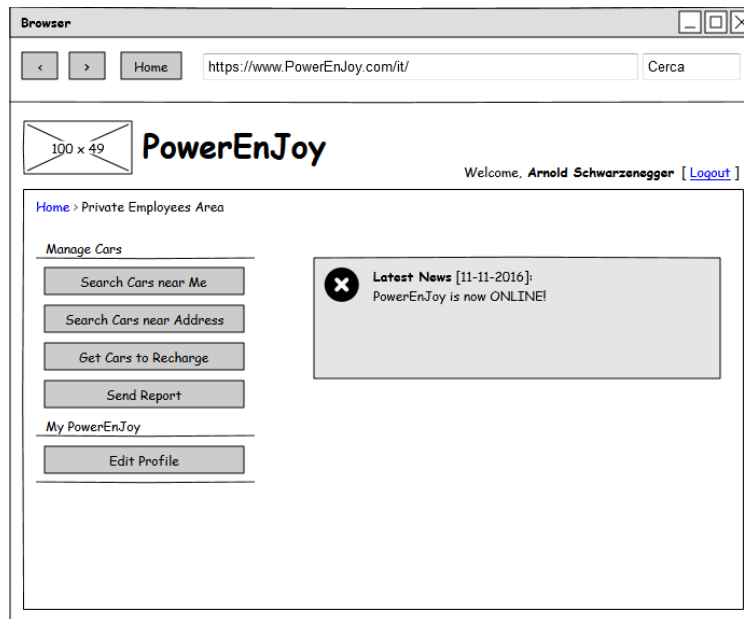


A screenshot of a mobile phone displaying the PowerEnJoy login page. The phone's status bar at the top shows the time as 12:30. The app header includes the "PowerEnJoy" logo and the word "Login". The "Welcome!" message is centered. The login form has "Username" (Us3nn4m3) and "Password" (pwd) fields. The "Remember me" checkbox is checked. A "Login" button is positioned above the "Sign up" and "Lost your password?" links.

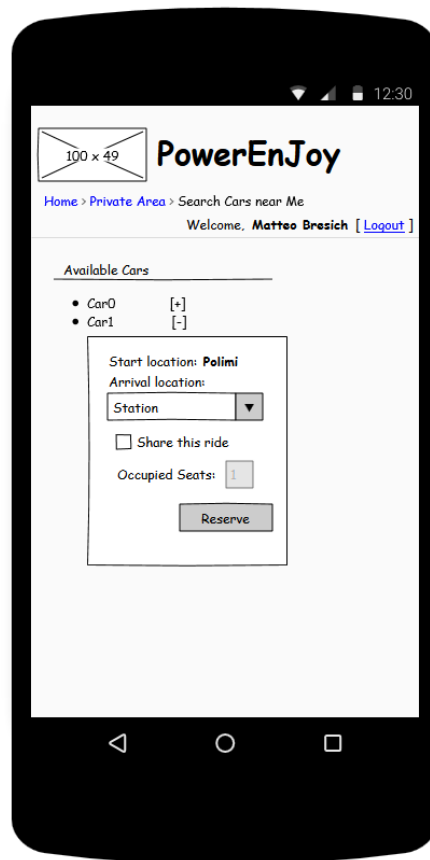
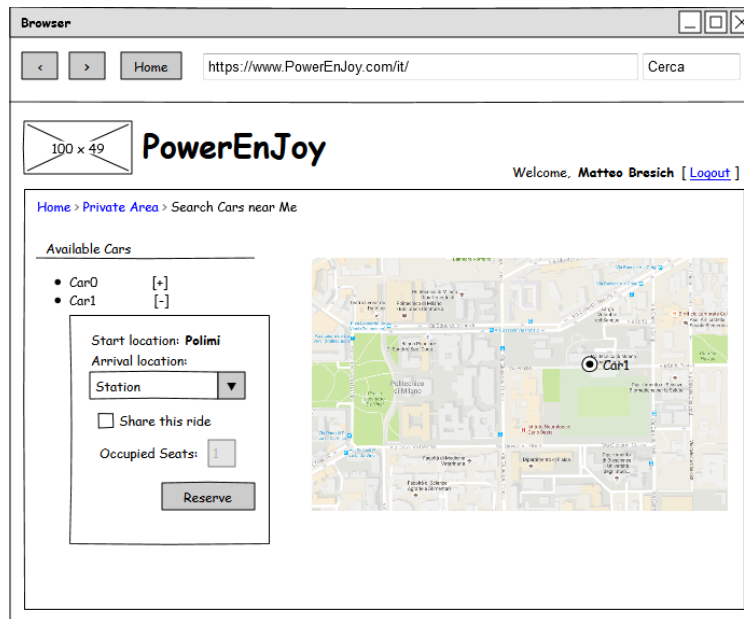
**3.1.1.3 Private Area** This page presents the private area for the registered user.



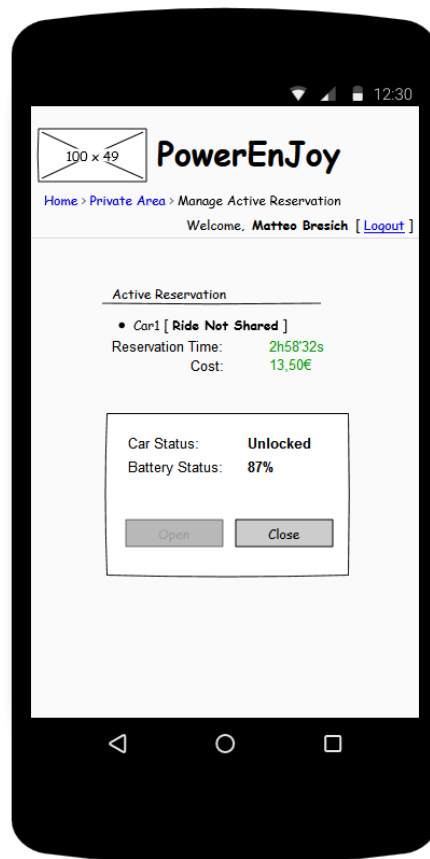
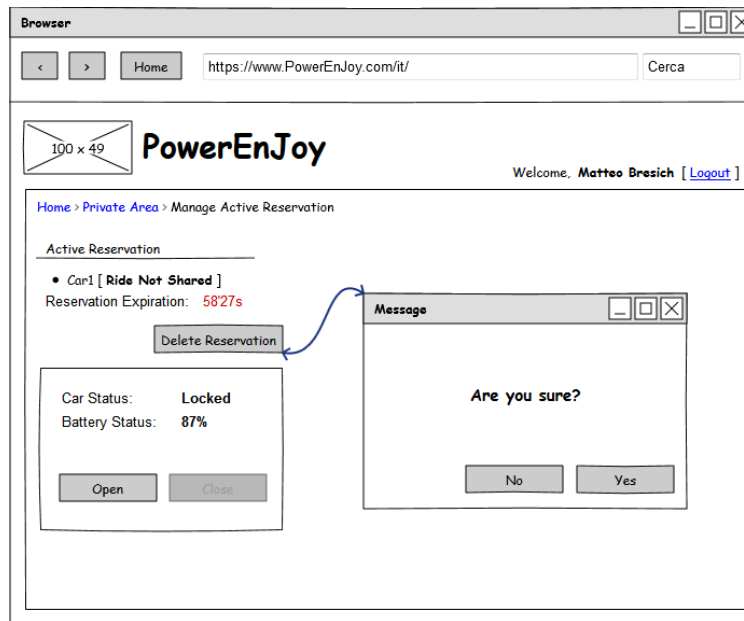
**3.1.1.4 Private Employees Area** This page presents the private area for the Handyman Operator.



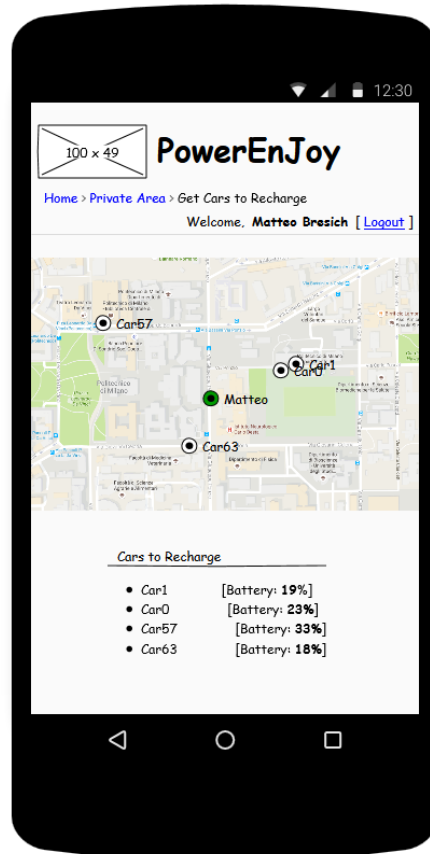
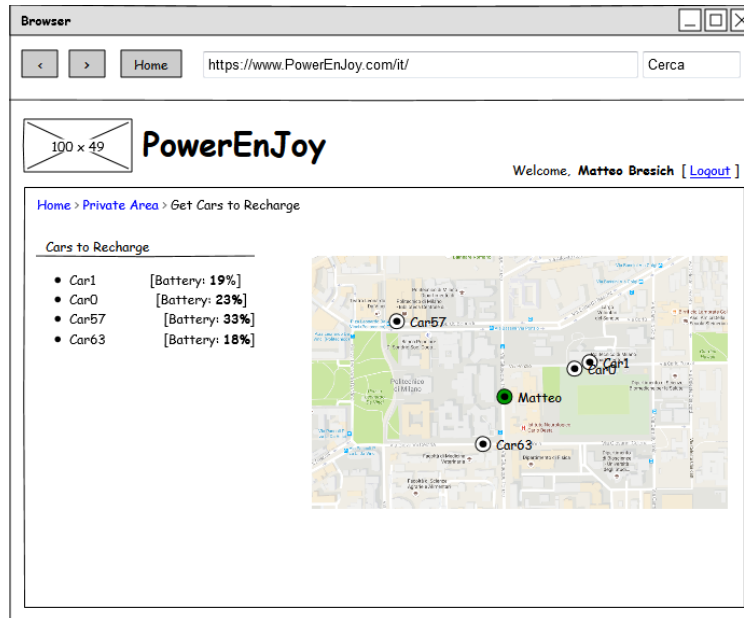
**3.1.1.5 Car Reservation Page** This page presents the car booking page for the registered user.



**3.1.1.6 Reservation Management Page** Follows the managment reservation page for the registered user.



**3.1.1.7 Get Cars to Recharge Page** This page presents cars that have to be connect to the charging station for the Handyman Operator.



### 3.1.1.8 Edit Profile

Follows the edit profile page for the registered user.

The screenshot shows a web browser window titled "Browser" with the address bar displaying "https://www.PowerEnJoy.com/it/". The page header includes the PowerEnJoy logo, a placeholder for a 100 x 49 image, and a welcome message: "Welcome, Matteo Bresich [Logout]". The breadcrumb trail is "Home > Private Area > Edit Profile". The "My Profile" section contains the following fields and buttons:

- Name\*: Matteo
- Surname\*: Bresich
- Email\*: matteo.bresich@mail.polimi.it
- Address\*: Piazza Leonardo Da Vinci, Milano
- Cellphone\*: 202-555-0177
- Payment\*: Change Payment Method
- Save

The screenshot shows the same "Edit Profile" page as the desktop version, but displayed on a smartphone screen. The layout is adapted for mobile, with the same fields and buttons visible: Name\* (Matteo), Surname\* (Bresich), Email\* (matteo.bresich@mail.polimi.it), Address\* (Piazza Leonardo Da Vinci, Milano), Cellphone\* (202-555-0177), Payment\* (Change Payment Method), and a Save button. The status bar at the top shows the time as 12:30.



### 3.1.2 Hardware Interfaces

*PowerEnJoy* does not require additional Hardware interface.

### 3.1.3 Communication Interfaces

*PowerEnJoy* uses the TCP transport protocol and HTTP/HTTPS over SSL application layer protocol to guarantee the security of data transmission.

## 3.2 Functional Requirements

### 3.2.1 [G1] The System has to encourage the user to have a good behaviour using discounts.

- [D1] The old payment system can be changed for adding discounts and fees.
- [D2] The cost of the service is calculated per hour (ex. 3,5h).
- [D3] The car provides occupied spots with weight sensors under seats.
- [R1] If the System detects that the user took at least two other passengers onto the car for at least half of the trip (in time term when the machine is in motion), the system applies a discount of 10% on the last ride.

### 3.2.2 [G2] The System has to discourage the user to have a bad behaviour using fees.

- [D1] The old payment system can be changed for adding discounts and fees.
- [D2] The old system has APIs to generate a payment.
- [D3] Access to APIs is provided only using the HTTPS protocol.
- [R1] If a car is not picked-up within one hour from the reservation the user pays a fee of 1 EUR.

### 3.2.3 [G3] The System has to ensure a good distribution of cars in the territory.

- [D1] The old system keeps track of the current cars' locations.
- [D2] The old system generate which cars has to be moved, the initial and the final location for the auto.
- [R1] The System has to generate a message to an Handyman Operator that contains: which car has to be moved, the initial and the final location for the auto.

### 3.2.4 [G4] The System has to provide the ability to share a ride.

- [D1] The old system keeps track of the current cars' locations.
- [D2] The old system has APIs to get available shared ride and to get for each ride available seats.
- [D3] Access to APIs is provided only using the HTTPS protocol.
- [R1] The System has to show a list of available shared ride.
- [R2] The System has to show ride details such as: initial location, final location, occupied seats.

**3.2.5 [G5] The Visitor shall be able to sign-up via app.**

- [D1] Phone number, email address and payments method for the registration are mandatory and must be valid.
- [R1] The User cannot register more than once.
- [R2] Unregistered users can only see information pages of the service, the sign up page and login page.
- [R3] The User must fill in all the mandatory fields, accept terms of use and the privacy policy of the service to successfully complete the registration process.
- [R4] If the User successfully complete the registration process the electronic card for unlock booked cars will be sent to the address of residence.

**3.2.6 [G6] The Registered User shall be able to log into the service.**

- [D1] Subscribers with the old system are already available in the database.
- [R1] If the Registered User hasn't the correct login information cannot access the system.
- [R2] If the Registered User provides correct credentials the system will grant him/her the access.

**3.2.7 [G7] The Registered User shall be able to reserve a car up to one hour before via app.**

- [R1] A car can be reserved by only one registered user at a time.
- [R2] A Registered User can choose if enable a shared ride or not.
- [R3] A Registered User can reserve only one car at time.
- [R4] A Registered User can reserve a car only if the last payment was done properly.

**3.2.8 [G8] The Registered User shall be able to remove a reservation before the reservation time runs out.**

- [R1] The Registered User shall be able to remove a reservation before the reservation time runs out.

**3.2.9 [G9] The Registered User shall be able to find the locations of available cars within a certain distance from him.**

- [D1] The GPS coordinates are always available and correct.
- [R1] The Registered User can search available cars that are no more than 4km distant from her/him simply providing his/her own position.

**3.2.10 [G10] The Registered User shall be able to find the locations of available cars from a specified address.**

- [D1] The GPS coordinates are always available and correct.
- [R1] The Registered User can search available cars that are no more than 4km distant from the address provided.

**3.2.11 [G11] The Registered User shall open/close a reserved car via app when he/she is close to it.**

- [D1] Every car can be locked and unlocked from the onboard computer.
- [D2] Every car has a bluetooth module connected to the onboard computer.
- [D3] The old system has APIs to locked and unlocked a selected car.
- [D4] Access to APIs is provided only using the HTTPS protocol.
- [R1] A Registered User can open and close a car previously reserved using the mobile app.
- [R2] A Registered User must to be close to the car not more than 5m.
- [R3] The Registered User's smartphone must have the application installed and the permissions for use the bluetooth.

**3.2.12 [G12] The Passenger shall be able to use a car in shared mode.**

- [R1] A Passenger can reserves a number of seats of a selected shared ride.
- [R2] A Passenger can make one reservation at a time.
- [R3] A Passenger after the booking must arrive at the place of departure before the estimated departure. The estimated departure is calculated from the difference between the car prenotation time and the current time.

**3.2.13 [G13] The Handyman Operator shall know the location of the car that has to be charged.**

- [D1] Every Handyman Operator has a business smartphone provided by the company with the app.
- [D2] The old system has APIs to get battery status for each available car.
- [D3] The old system has APIs to get a list of cars that have to be attach to the charging system.
- [D4] The old system keeps track of the current cars' locations.
- [R1] A Handyman Operator can get what are the cars that have to be attach to the charging system.
- [R2] A Handyman Operator can get where are the cars that have to be attach to the charging system.

**3.2.14 [G14] The Handyman Operator shall report any problem with a car.**

- [D1] Every Handyman Operator has a business smartphone provided by the company with the app.
- [R1] A Handyman Operator can send a report if there are any problem with cars.

### 3.3 Scenarios

#### 3.3.1 Scenario 1

Chiara is an university student and a registered user to the service. Every day she uses public transportation to reach the school in the center of Milan. During the year because of the public transportation's strikes she is forced to find an alternative way to arrive at her destination. Just the other week when she had to go to college for an exam, when she arrives to Cadorna station because of the metro service strike, decided to use the app *PowerEnJoy* installed on her smartphone. After checking that there is a car available in the equipped parking nearest her, she book it and went to the equipped parking to get to her destination. The car chosen is fully charged. After using it for the most of the day Chiara brings back the car to the Cadorna's equipped parking. Once she parked, the system detects the end of the race and automatically computes the fare and applies a discount on this ride of 20% because the battery car charge is more than 50%. The transaction is performed automatically by the user's method of payment already chosen

#### 3.3.2 Scenario 2

Antonio wants to spend the night at the club. He is registered to the car sharing service. He book a car by calling the call center. He gets the car at the equipped parking close to home by opening it with his smartphone, he goes to take Sofia and Elena. After a fun evening the group recovers the car to the disco's parking and return home. After bringing home all, Antonio brings the car to the initial equipped parking. Once he parked, the system detects the end of the race and automatically computes the fare and applies a discount on this ride of 10% because for more than 50% of the movement time, Antonio led at least two other passengers with him. The transaction is performed automatically by Antonio's method of payment already chosen

#### 3.3.3 Scenario 3

Four friends live near Lambrate's station. They want to go to the San Siro stadium for a concert. Andrea is registered to the car sharing service. After checking that there is a car in Lambrate's station he goes to catch it. The group spent 30 minutes to arrive at the stadium. After the concert only Alessandro come back with Andrea. They spent 45 minutes because of an accident. At the end of the ride the system automatically computes the fare and doesn't apply a discount on this ride of 10% because for more than 50% of the movement time, there are only Andrea and Alessandro. The transaction is performed automatically by Andrea's method of payment already chosen

#### 3.3.4 Scenario 4

Two group have to go from the Centrale's station to the Garibaldi's Station (both are equipped parking). Adrian, Sonia and Alessandro are in the first group instead Claudio and Silvia are in the second group. Adrian and Claudio are registered to the service. Adrian reserves a car for a shared ride. Claudio looks for a shared ride and finds Adrian's ride and reserves the remaining two seats. When they arrive to Garibaldi's Station the ride is over and the system automatically computes the fare and applies a discount on this ride of 10% because for more than 50% of the movement time, the car was sold out. The fare is shared between Adrian and Claudio (Adrian:  $\frac{3}{5}$  of the total amount, Claudio:  $\frac{2}{5}$  of the total amount). The transaction is performed automatically by the users' method of payment already chosen

#### 3.3.5 Scenario 5

Alessia leaves the car at the Bovisa's equipped parking, despite the fact that the car needs to be recharged, she has no time to connect the car to the charging system.

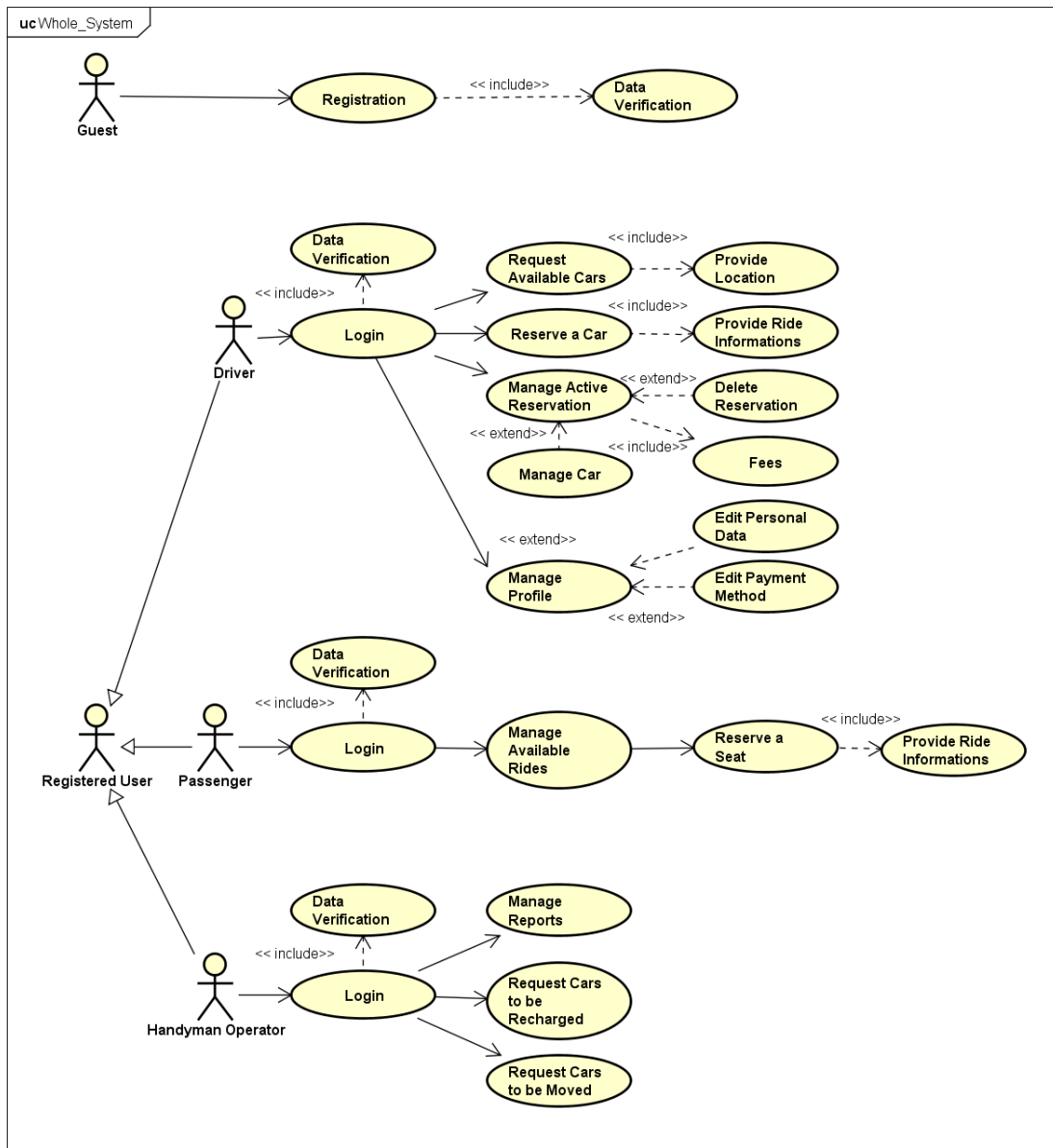
Daniele, an handyman operator, using the *PowerEnJoy* application is checking if there are cars that need to be recharged. The car number 5 needs to be connect. Daniele reach the car and connect it.

### 3.3.6 Scenario 6

Roberta, Maria and Luigi that have token their *PowerEnJoy* 's cars in other equipped parking leave them in the equipped parking A. The system realizes that there are too much cars in the equipped parking A and sends a message to the handyman operator Fabio. The message says: "Move car 5 from the equipped parking A to the equipped parking B". Fabio moves the car to the indicated location.

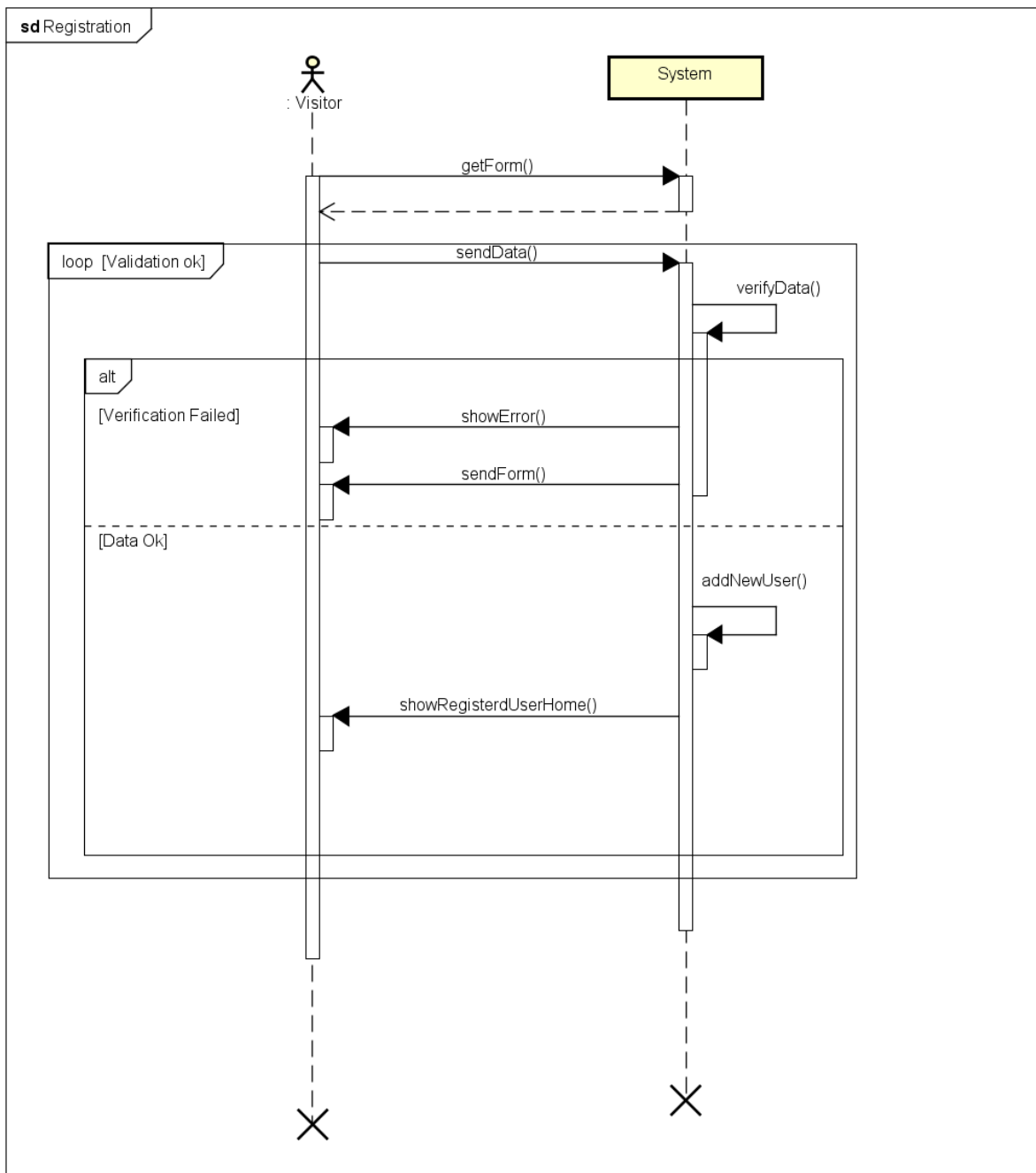
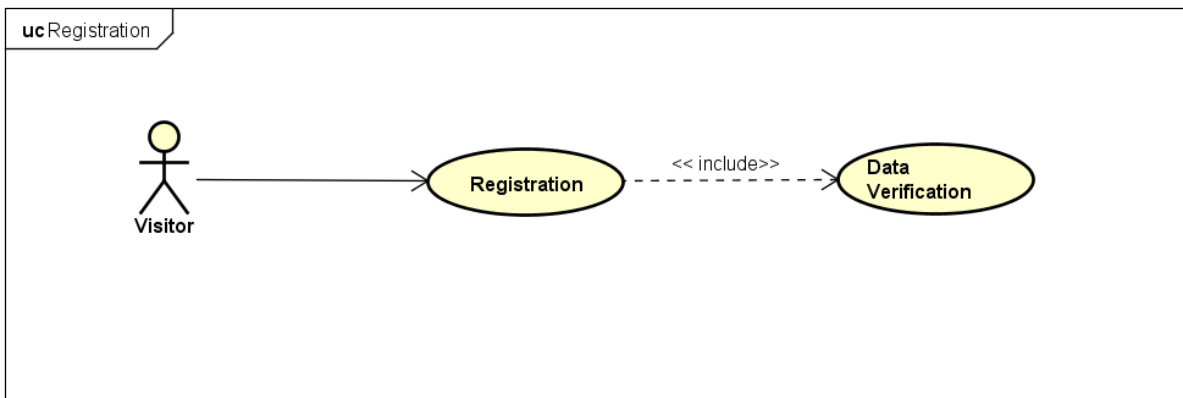
## 3.4 UML Models

### 3.4.1 Use Case



#### 3.4.1.1 Visitor registers to *PowerEnJoy*

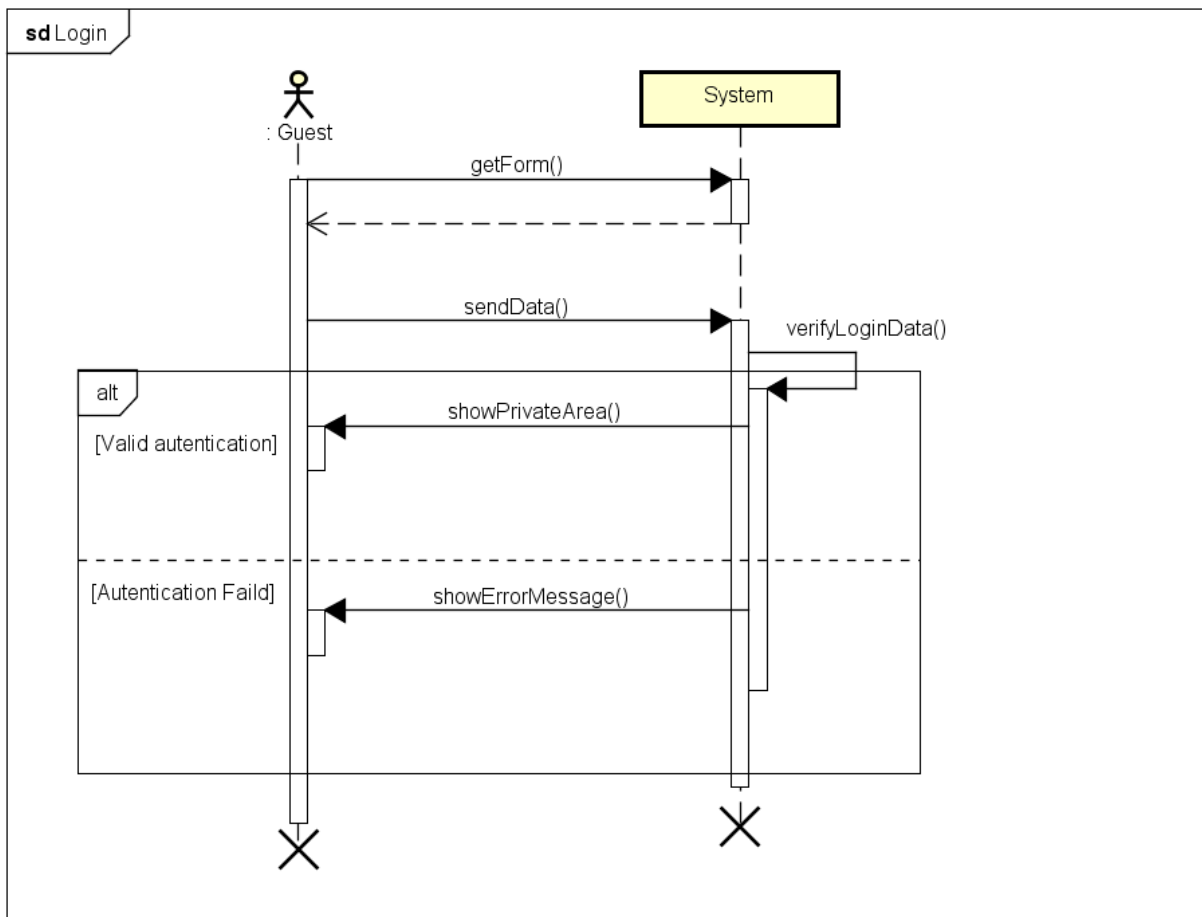
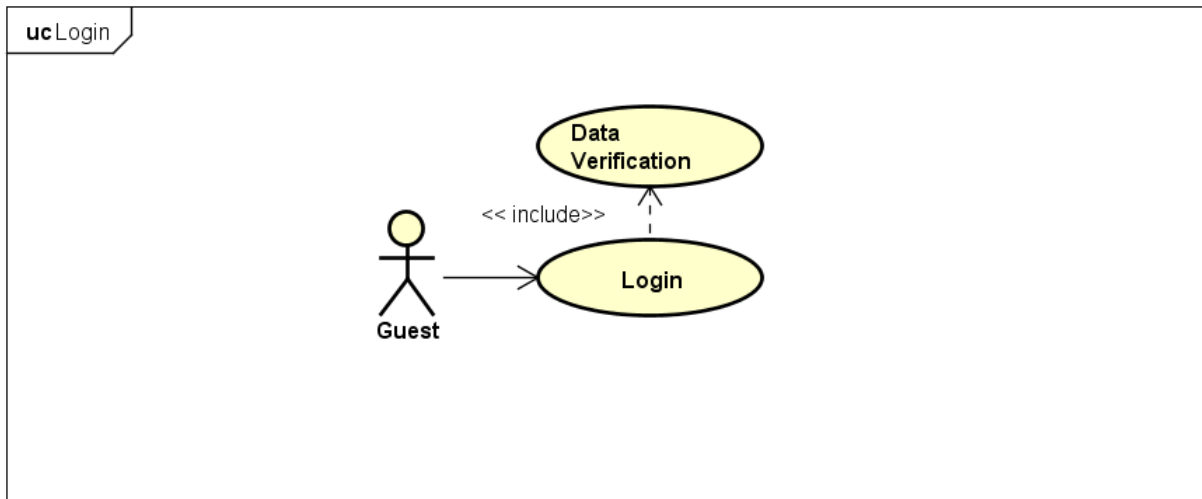
Actor	Visitor
Goal	[G5]
Precondition	None
Event Flow	<ol style="list-style-type: none"><li>1. Visitor on the home page clicks on "Register".</li><li>2. Visitor fills all required fields.</li><li>3. Visitor clicks on the "Send" button.</li><li>4. The application saves data in the database.</li></ol>
Postcondition	Visitor completes the registration process, becomes a registered user and obtains the username and the password generated by the system through email.
Exceptions	<ol style="list-style-type: none"><li>1. The visitor is already a registered user.</li><li>2. Data not valid or missing.</li><li>3. The email chosen is already associated to another user.</li></ol>
Exception Handling	All exceptions are handled alerting the visitor of the problem through an error message.





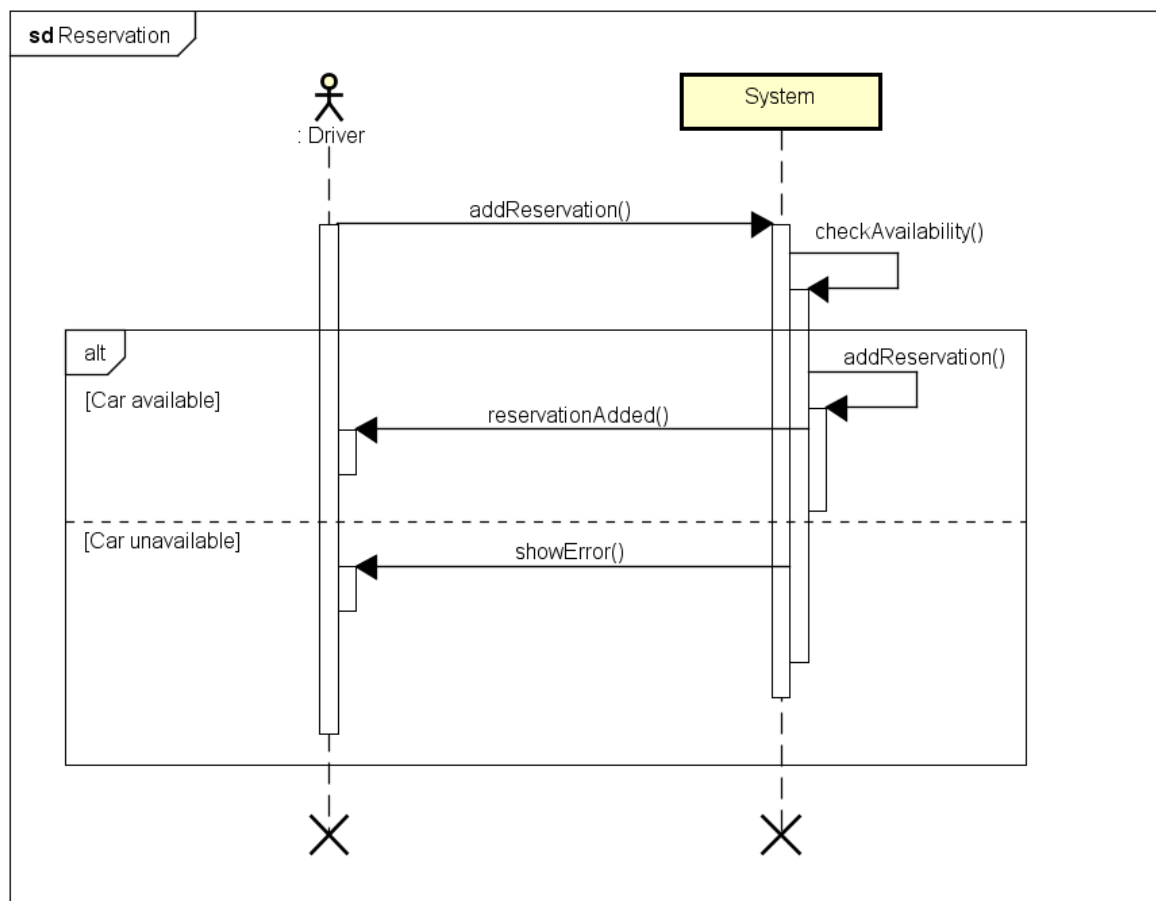
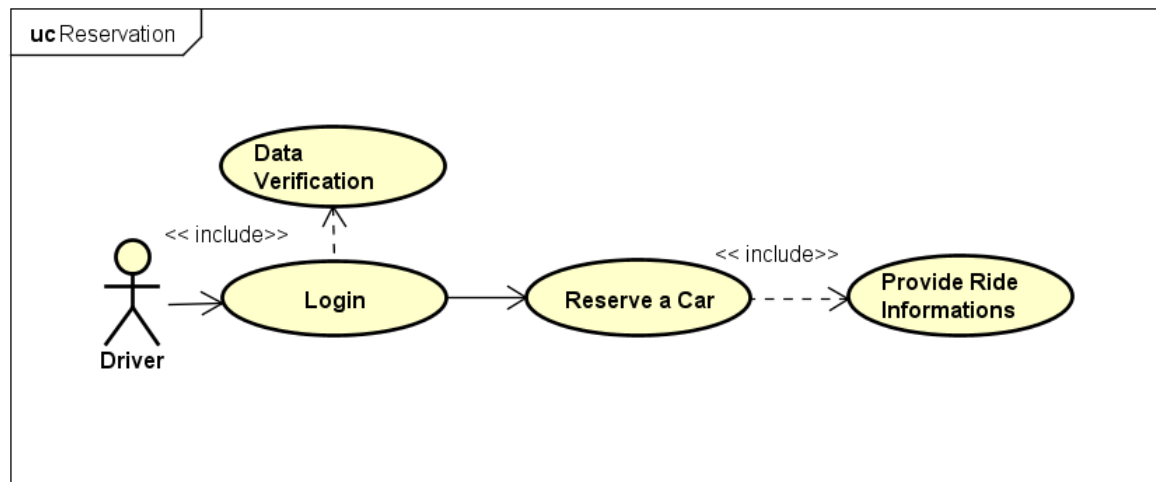
#### 3.4.1.2 Visitor login to *PowerEnJoy*

Actor	Visitor
Goal	[G6]
Precondition	The Visitor must be already registered.
Event Flow	<ol style="list-style-type: none"><li>1. Visitor navigates to the login page.</li><li>2. Visitor fills all required fields.</li><li>3. Visitor clicks on the "Login" button.</li></ol>
Postcondition	The application verifies the credentials and if correct shows the home page for the registered user and promotes guest to registered user.
Exceptions	<ol style="list-style-type: none"><li>1. Data not valid or missing.</li></ol>
Exception Handling	All exceptions are handled alerting the visitor of the problem through an error message.



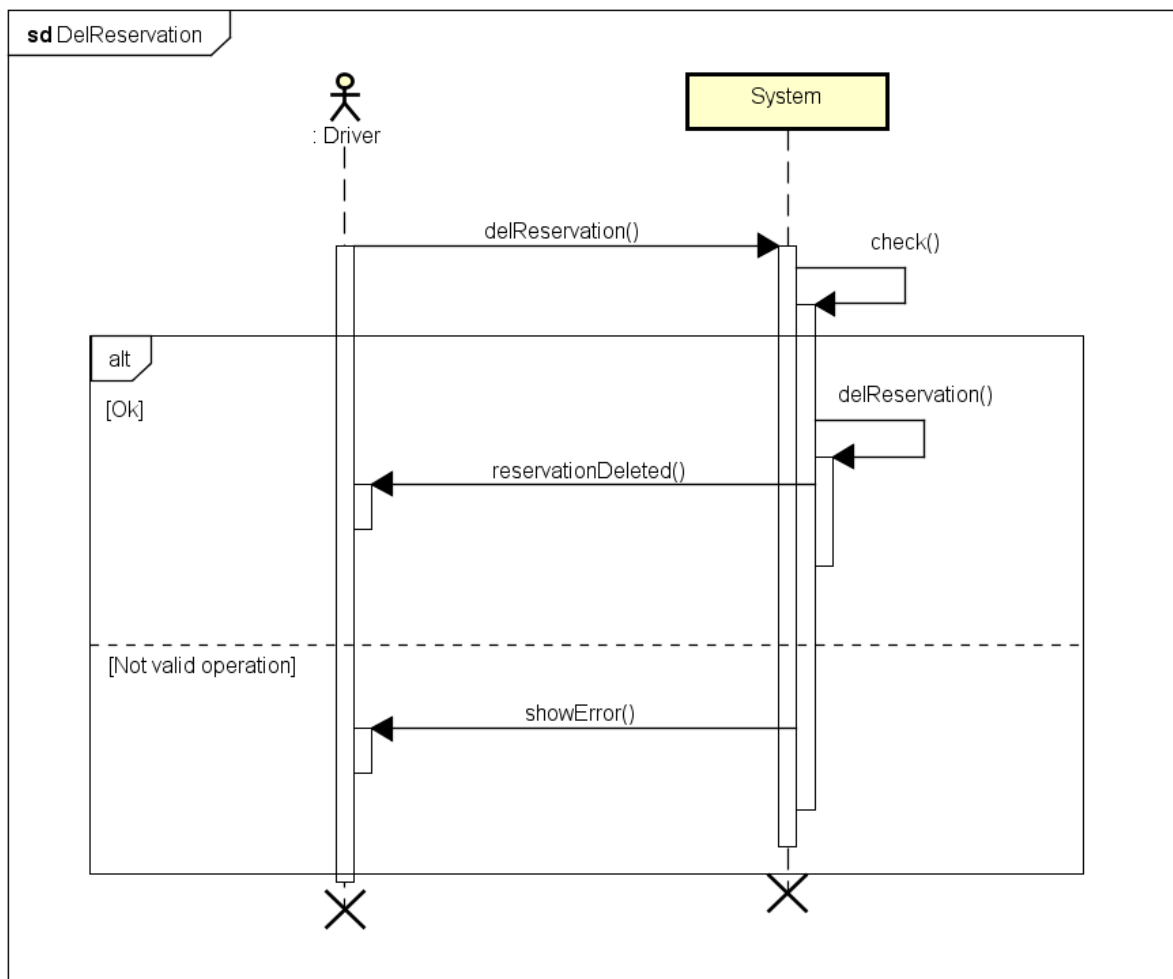
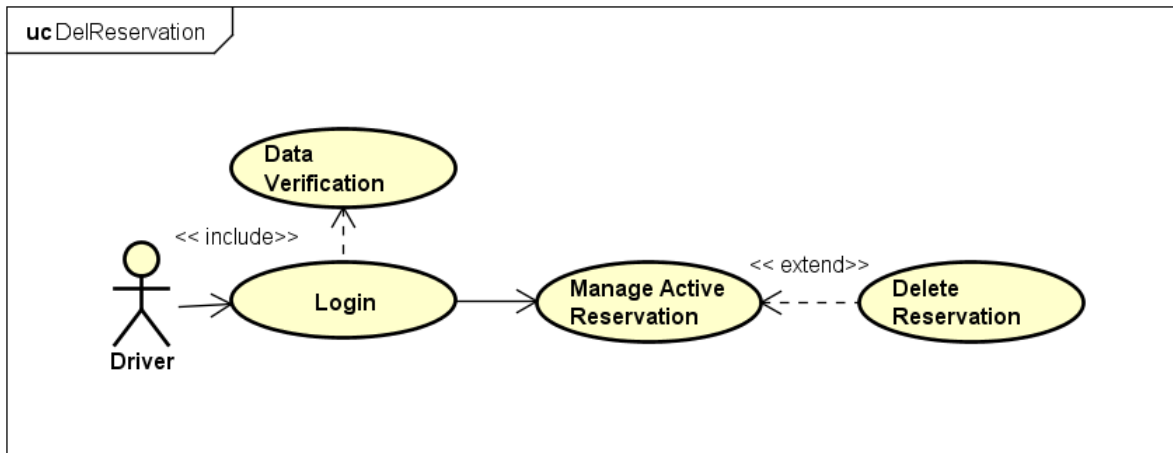
### 3.4.1.3 Registered User reserve a car using *PowerEnJoy*

Actor	Registered User
Goal	[G7]
Precondition	The Registered User must be already logged in.
Event Flow	<ol style="list-style-type: none"><li>1. The Registered User search an available car within a certain distance from him or from a specified address.</li><li>2. The Registered User selects the desired car from a list.</li><li>3. The Registered User click on the "Reserve" button.</li><li>4. The Registered User can choose if share his/her ride.</li><li>5. The Registered User, in the case of shared ride, have to provide the arrival location and occupied seats.</li></ol>
Postcondition	The application reserve the selected car for one hour. A timer stars to check the hour when the reservation it's confirmed.
Exceptions	<ol style="list-style-type: none"><li>1. Already reserved car</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.



#### 3.4.1.4 Registered User delete a reservation using *PowerEnJoy*

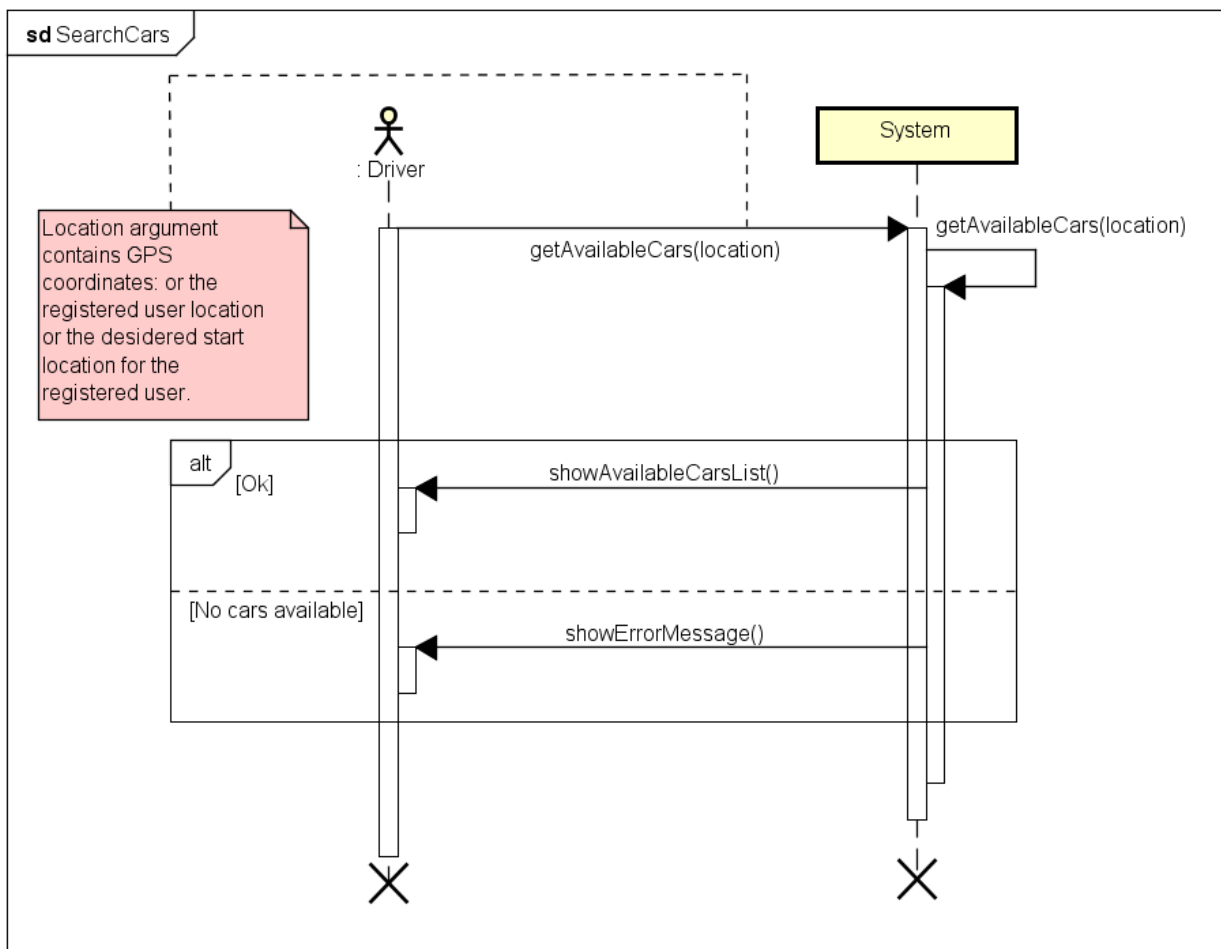
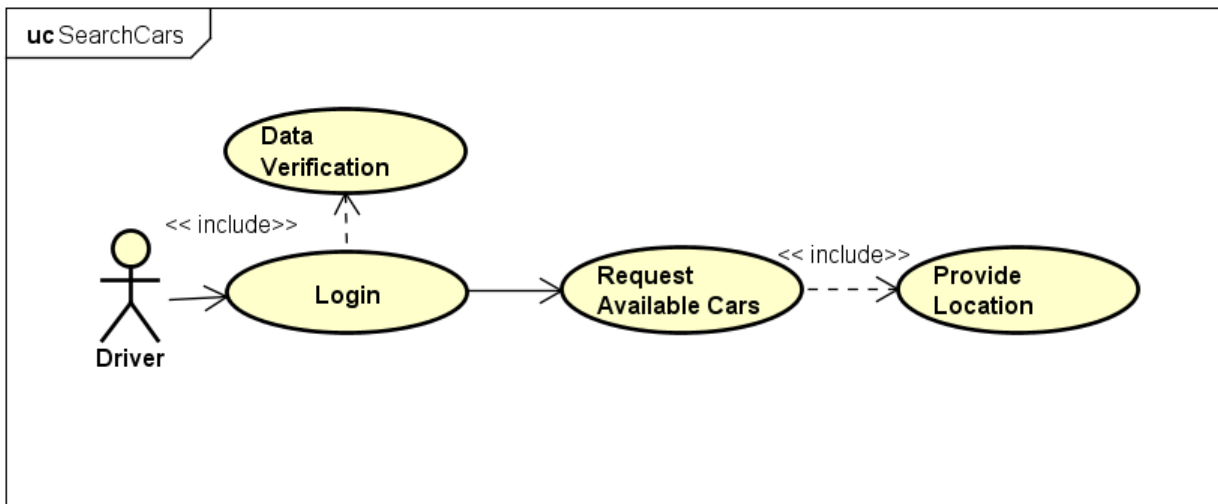
Actor	Registered User
Goal	[G8]
Precondition	The Registered User must be already logged in. The Registered User must have made a reservation. The reservation time is not exceeded.
Event Flow	<ol style="list-style-type: none"><li>1. The Registered User select from the menu "Manage Active Reservation".</li><li>2. The Registered User selects the unique active reservation.</li><li>3. The Registered User click on the "Delete Reservation" button.</li><li>4. The Registered User at the message "Are you sure?" clicks on the "Yes" button.</li></ol>
Postcondition	The application cancel reservation and changes the state of the car in available.
Exceptions	<ol style="list-style-type: none"><li>1. Reservation is exceeded.</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.



#### 3.4.1.5 Registered User search available cars using *PowerEnJoy*

Actor	Registered User
Goal	[G9]
Precondition	The Registered User must be already logged in. The Registered User must provide his/her location.
Event Flow	1. The Registered User select from the menu "Search Cars Near Me".
Postcondition	The application retrieve all available cars near user location and show them in a list.
Exceptions	1. No cars available.
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.

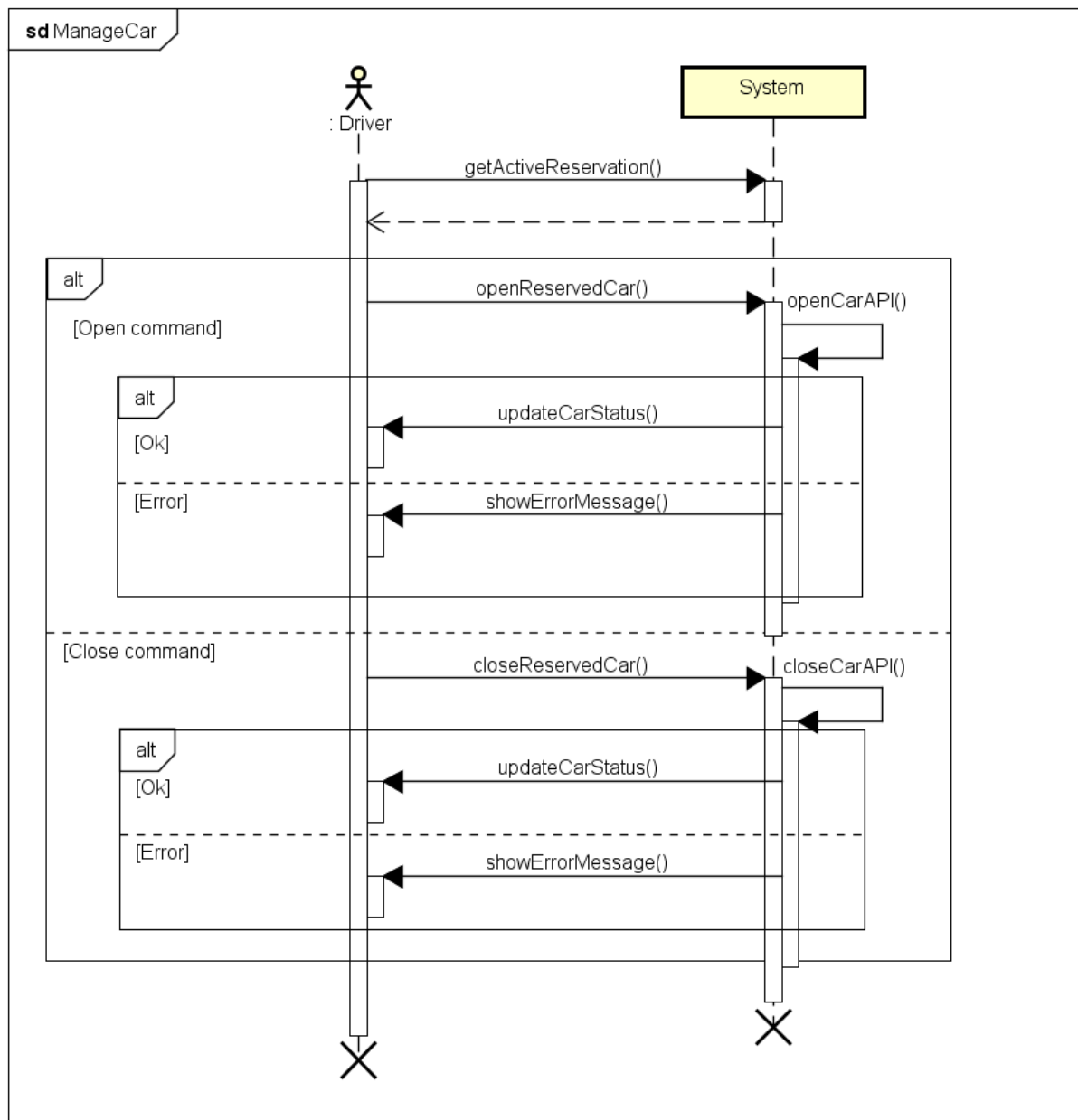
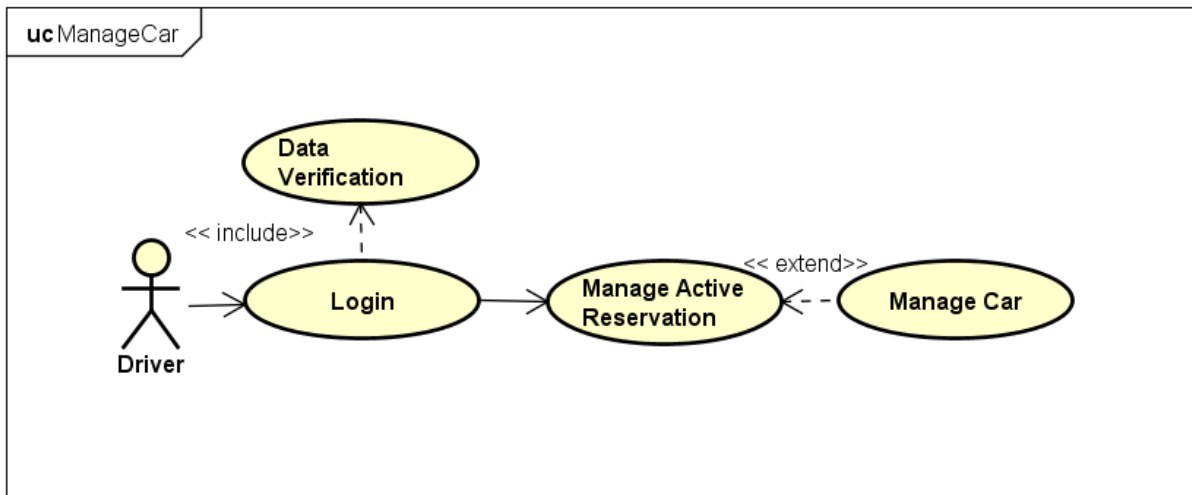
Actor	Registered User
Goal	[G10]
Precondition	The Registered User must be already logged in. The Registered User must provide an address for the desired start location.
Event Flow	1. The Registered User select from the menu "Search Cars Near the Address".
Postcondition	The application retrieve all available cars near the address provided and show them in a list.
Exceptions	1. No cars available.
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.





#### 3.4.1.6 Registered User manage a reserved car using *PowerEnjoy*

Actor	Registered User
Goal	[G11]
Precondition	The Registered User must be already logged in. The Registered User must have made a car reservation. The car reservation is still active.
Event Flow	<ol style="list-style-type: none"><li>1. The Registered User select from the menu "Manage Active Reservation".</li><li>2. The Registered User click the "Open"/"Close" button.</li></ol>
Postcondition	The application unlock/lock the reserved car.
Exceptions	<ol style="list-style-type: none"><li>1. Unable to open/close the car.</li><li>2. The user is not close enough to the car</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.

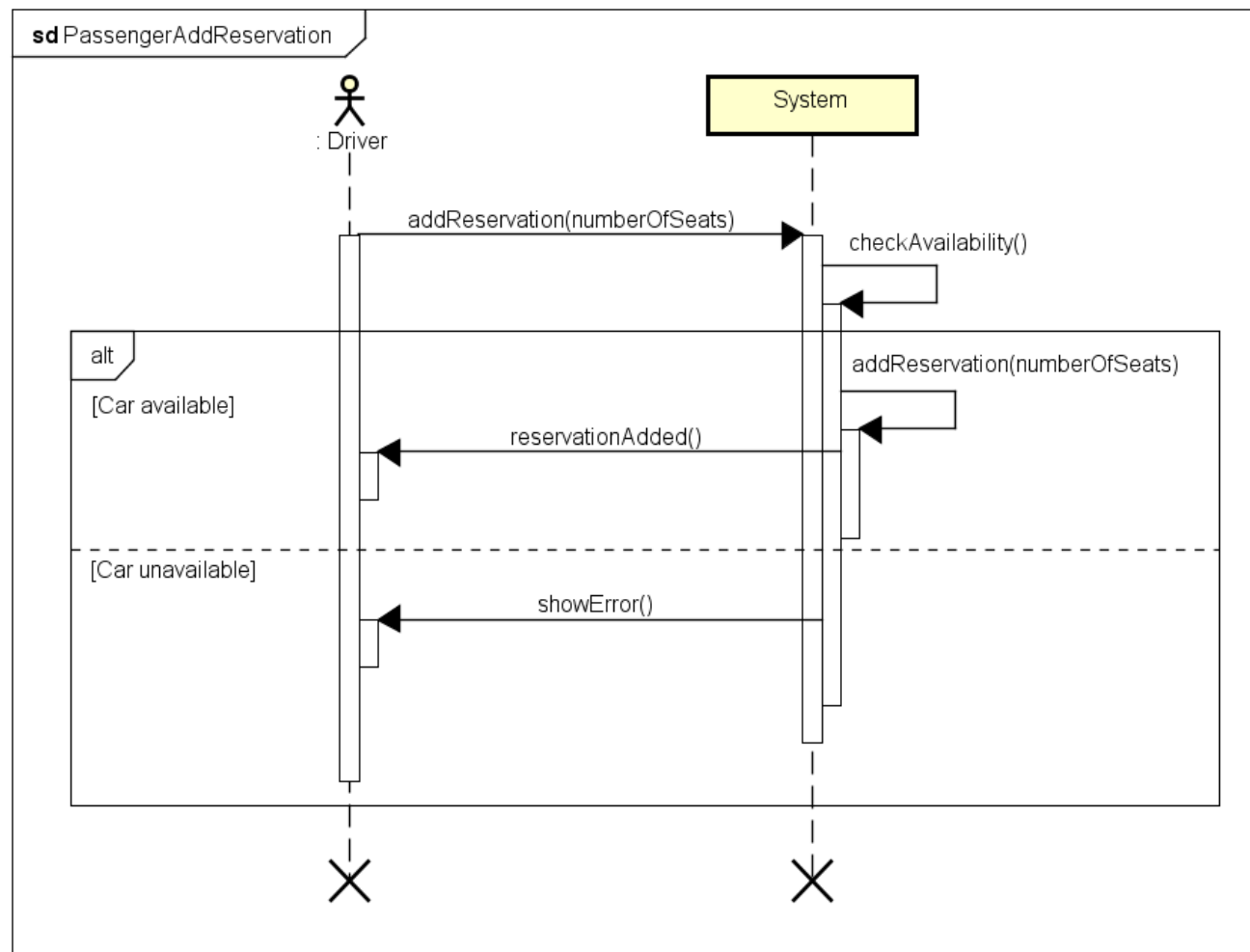
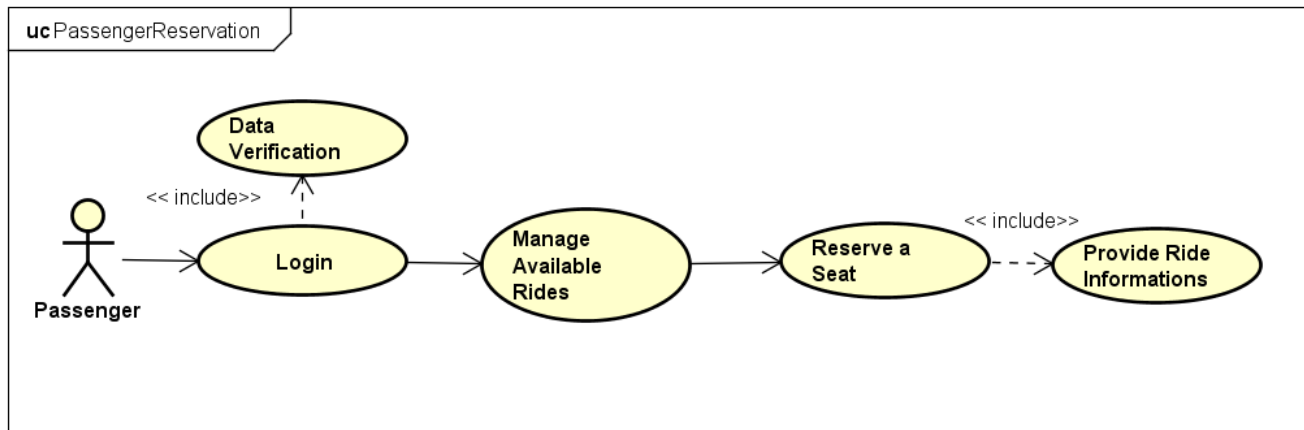


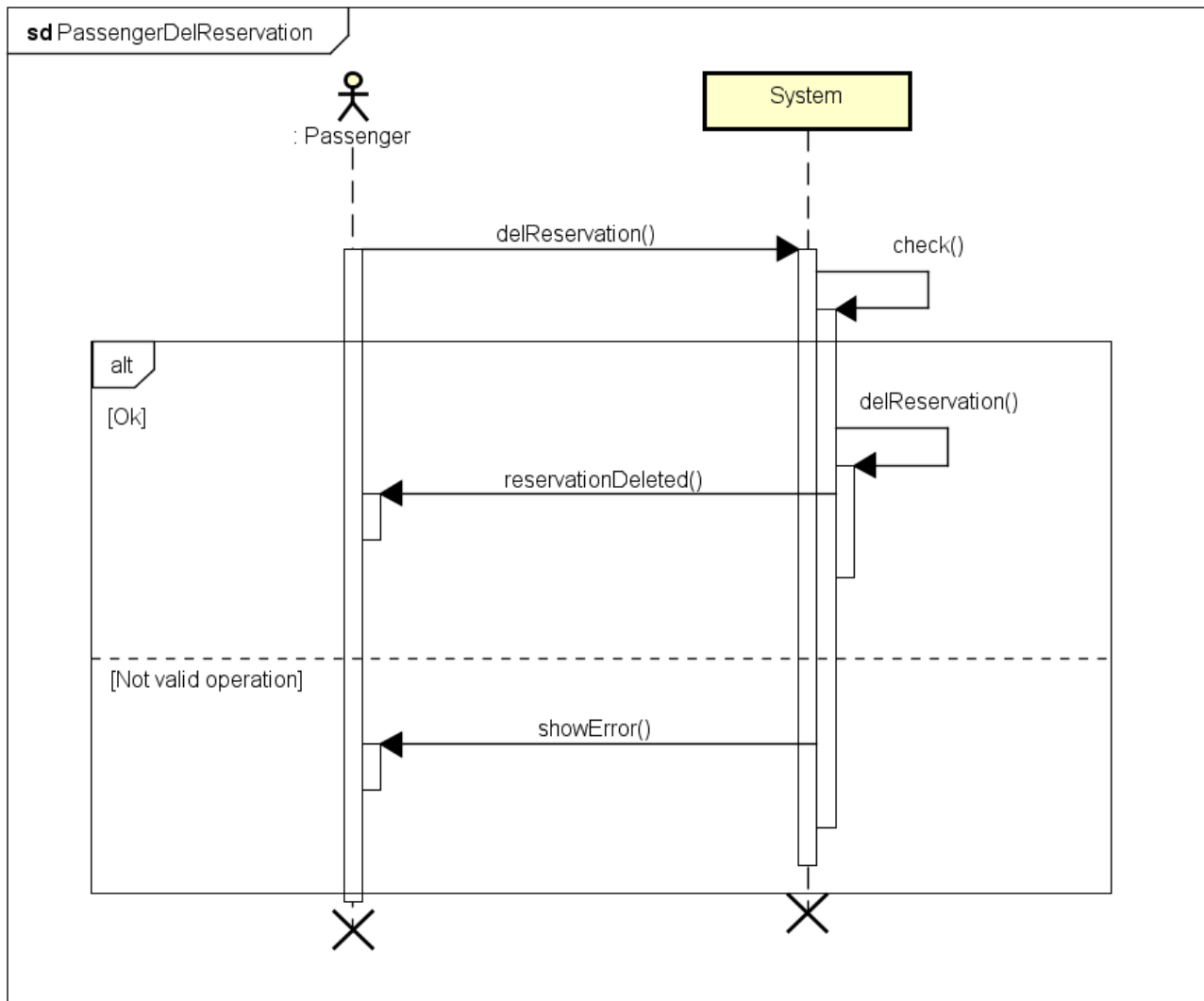
#### 3.4.1.7 Passenger reserves a number of seats of a shared ride using *PowerEnjoy*

Actor	Passenger
Goal	[G12]
Precondition	The Passenger must be already logged in.
Event Flow	<ol style="list-style-type: none"><li>1. The Passenger select from the menu "Join a Ride".</li><li>2. The Passenger chooses at which shared ride he/she wish to join from a list.</li><li>3. The Passenger must provide the number of the seats he/she wants to reserve.</li></ol>
Postcondition	The application reserves seats of the selected ride.
Exceptions	<ol style="list-style-type: none"><li>1. Impossible to reserve seats.</li><li>2. Wrong number of seats.</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.

#### 3.4.1.8 Passenger delete a reservation of a shared ride using *PowerEnjoy*

Actor	Passenger
Goal	[G12]
Precondition	The Passenger must be already logged in.
Event Flow	<ol style="list-style-type: none"><li>1. The Passenger selects from the menu "Manage Active Reservation".</li><li>2. The Passenger selects the unique active reservation.</li><li>3. The Passenger click on the "Delete Reservation" button.</li><li>4. The Passenger at the message "Are you sure?" clicks on the "Yes" button.</li></ol>
Postcondition	The application delete the seats reservation.
Exceptions	<ol style="list-style-type: none"><li>1. Reservation is exceeded.</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.

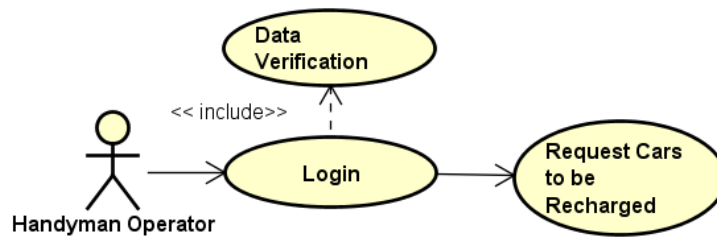




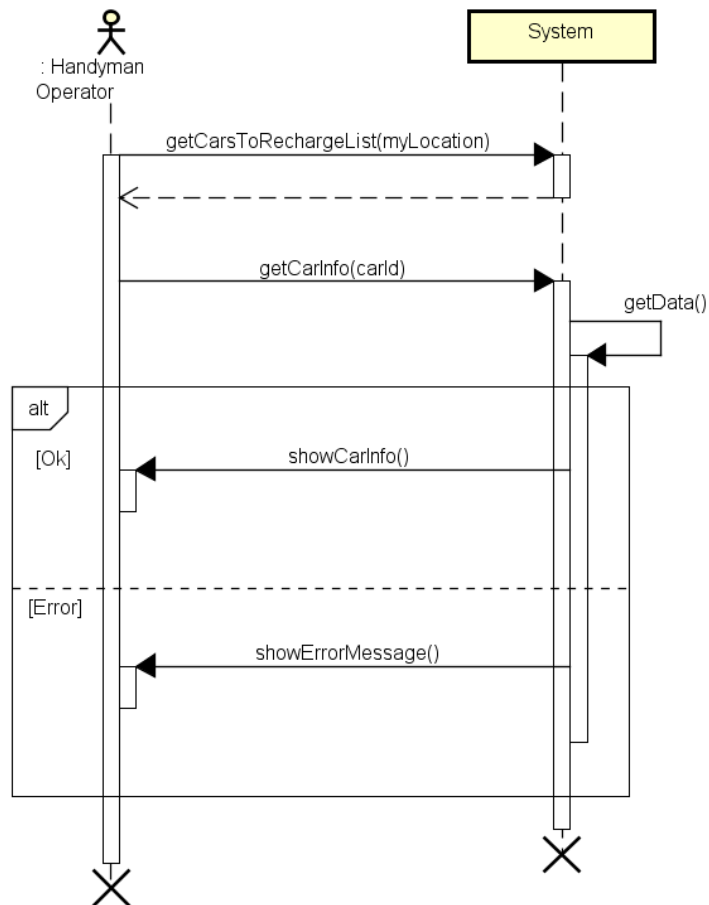
**3.4.1.9 Handyman operator gets which and where are cars that have to be attach to the charging system using *PowerEnJoy***

Actor	Handyman operator
Goal	[G13]
Precondition	The Handyman operator must be already logged in.
Event Flow	<ol style="list-style-type: none"><li>1. The Handyman operator selects from the menu "Get Cars to Recharge".</li><li>2. The Handyman operator gets a car list order by distance from him/her.</li><li>3. The Handyman operator click on the "More Info" button.</li></ol>
Postcondition	The application show which and where are available cars that have to be attach to the charging system.
Exceptions	<ol style="list-style-type: none"><li>1. No cars to be recharged.</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.

uc CarsToRecharge



sd CarsToRecharge

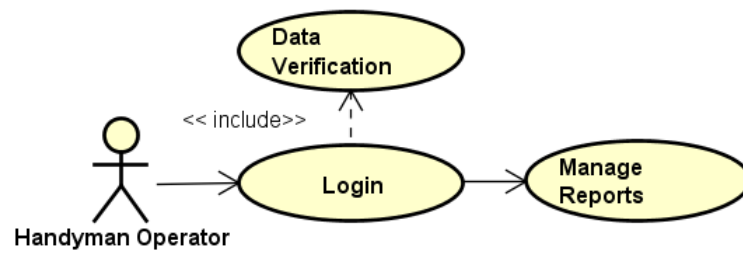


#### 3.4.1.10 Handyman operator sends a report using *PowerEnJoy*

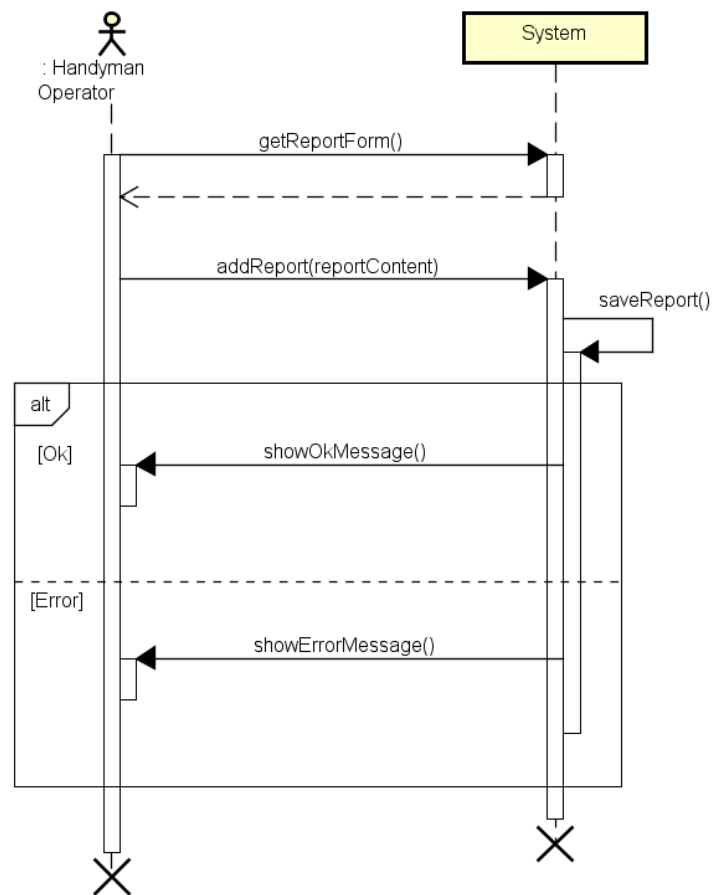
Actor	Handyman operator
Goal	[G14]
Precondition	The Handyman operator must be already logged in.
Event Flow	<ol style="list-style-type: none"><li>1. The Handyman operator selects from the menu "Send a Report".</li><li>2. The Handyman operator fills all required fields.</li><li>3. The Handyman operator click on the "Send Report" button.</li></ol>
Postcondition	The application generate a report that is sent to the old system.
Exceptions	<ol style="list-style-type: none"><li>1. Data not valid or missing.</li></ol>
Exception Handling	All exceptions are handled alerting the Registered User of the problem through an error message.



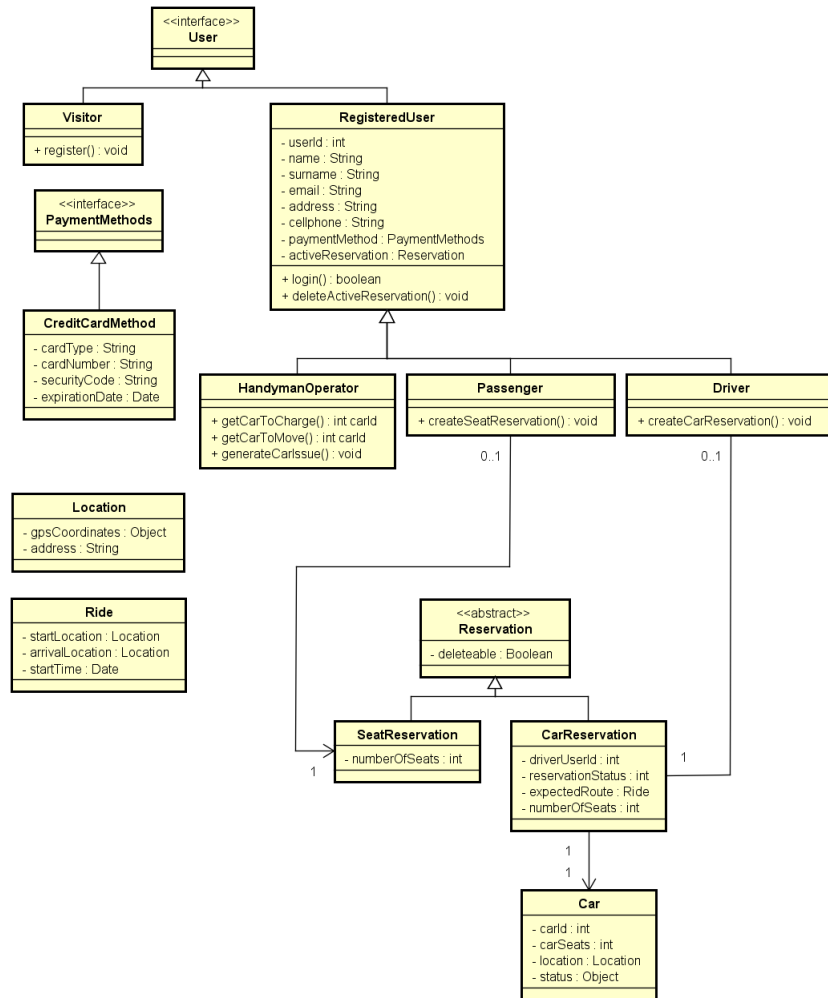
uc GenerateReport



sd GenerateReport

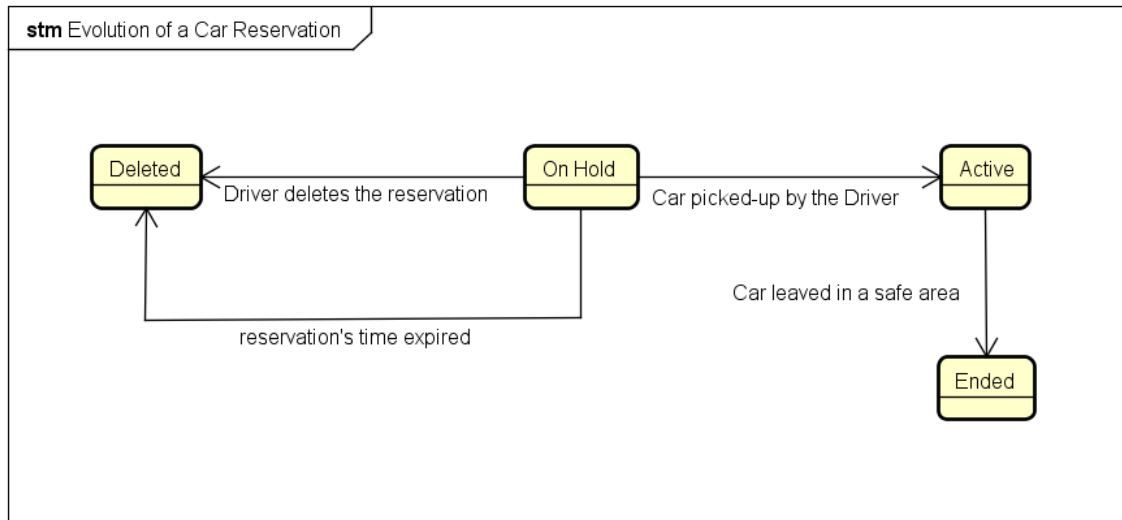


### 3.4.2 Class Diagram

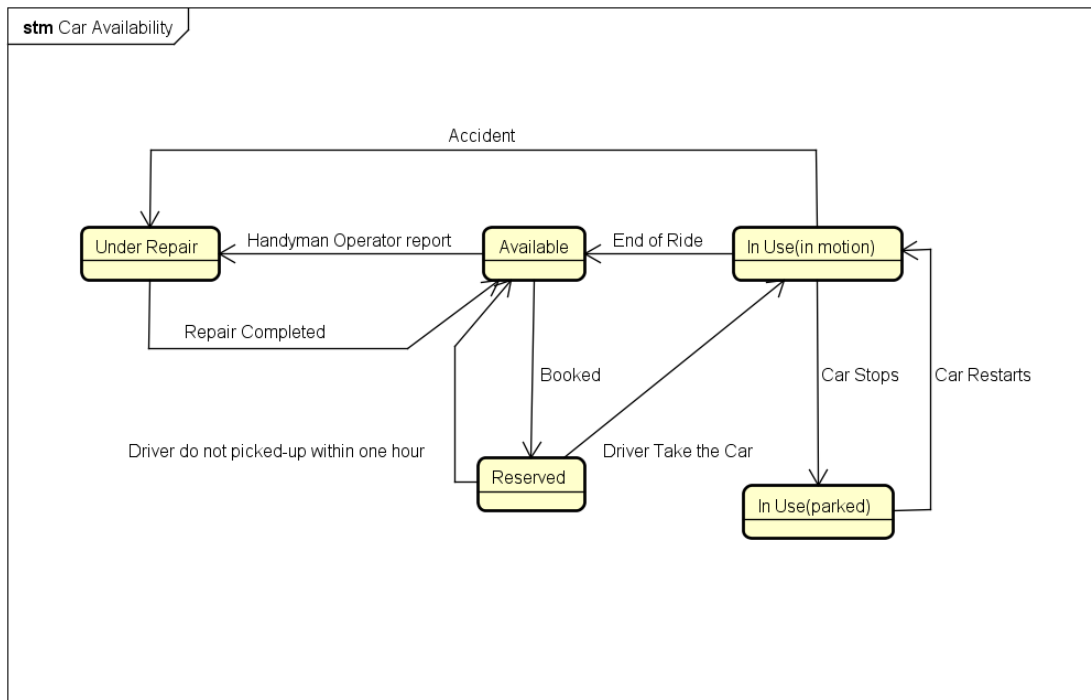


### 3.4.3 State Machine Diagrams

**3.4.3.1 Evolution of a Car Reservation** The following finite state machine shows the possible states of a car reservation and events that enable the change of state.



**3.4.3.2 Car Availability** The following finite state machine shows the possible states of Availability of a car and events that enable the change of state.



## **3.5 Non Functional Requirements**

### **3.5.1 Performance Requirements**

To ensure an high level of usability of the service it is important to calibrate the server resources. The objective is to avoid a deadlock due to the overload in situations of heavy loads.

### **3.5.2 Availability & Reliability**

The system must guarantee an availability of 99%. The application will be hosted by multiple redundant servers to prevent service failures due to hardware faults. All servers will be equipped with an autonomous power supply backup system.

### **3.5.3 Security**

All the communications between servers and clients must be protected by strong encryption using the SSL protocol. Users' passwords must not be stored in plain text in the database. The application must generate a snapshot of the last 10 days system's state.

### **3.5.4 Maintainability**

When finished the software will not require any particular update except for security fix and the maintenance for APIs offered by the system. The code should be well documented to help developers to understand and edit it quickly.

### 3.6 Alloy

```
open util/boolean

// Model definition

sig Str {} //String

sig Flo {} //Float

sig Date {}

sig PaymentMethod {}

abstract sig RegisteredUser {
  ID: one Int,
  password: one Str,
  name: one Str,
  surname: one Str,
  email: one Str,
  address: one Str,
  cellphone: one Str,
  savedPaymentMethod: one PaymentMethod,
  activeReservation: lone Reservation
}
{ID > 0}

sig Driver extends RegisteredUser {}

sig Passenger extends RegisteredUser {}

sig HandyMandOperator extends RegisteredUser {}

abstract sig Reservation {
  reservationId: one Int,
  car: one Car
}
{reservationId > 0}

sig CarReservation extends Reservation {
  isShared: one Bool,
  numberOfSeatsReserved: one Int,
  seatsReservation: set SeatReservation,
}
{ numberOfSeatsReserved > 0}

sig SeatReservation extends Reservation {
  numberOfSeatsReserved: one Int
}
{ numberOfSeatsReserved > 0}

sig Car {
  carPlate: one Str,
  seats: one Int
```

```

}

sig Location {}

// Model constraints
fact noEmptyRegisteredUser {
    all u: RegisteredUser |
        (#u.ID = 1) and (#u . password=1) and (#u.name = 1) and(#u.
            ↪ surname = 1)
        and (#u.email = 1) and (#u.address = 1) and (#u.cellphone = 1)
            ↪ and (#u.savedPaymentMethod ≥ 1)
}

fact uniqueUserID {
    no disj u1, u2: RegisteredUser | (u1.ID = u2.ID)
}

fact uniqueRegisteredUser {
    no disj u1, u2: RegisteredUser | (u1.email = u2.email)
}

fact uniqueUserReservation {
    no disj u1, u2: RegisteredUser | (u1.activeReservation = u2.
        ↪ activeReservation)
}

fact uniqueReservation {
    no disj r1, r2: Reservation | (r1.reservationId = r2.reservationId)
}

fact uniqueCar {
    no disj c1, c2: Car | (c1.carPlate = c2.carPlate)
}

fact noSameCarInCarReservation {
    no disj r1, r2: CarReservation | r1.car = r2.car
}

fact carReservationMustHaveAuthor {
    no r: CarReservation | all d: Driver | r not in d.activeReservation
}

fact seatReservationMustHaveAuthor {
    no r: SeatReservation | all p: Passenger | r not in p.activeReservation
}

fact seatReservationMustHaveCarReservation {
    no r: SeatReservation | all c: CarReservation | r not in c.
        ↪ seatsReservation
}

fact seatReservationMustHaveTheSameFatherReservationCar {

```

```

    all rs:SeatReservation | all rc: CarReservation | (rs in rc.
        ↪ seatsReservation)      (rs.car = rc.car)
}

//check noPassengersOnNonSharedReservation
assert noPassengersOnNonSharedReservation {
    all r: CarReservation | (r.isShared=False)      (#seatsReservation = 0)
}

//Predicates
pred show() {
    #HandyMandOperator = 1
    #Driver = 2
    #Passenger = 2
    #CarReservation = 2
    #car ≥2
}
run show for 10

```





## 4 Appendix

### 4.1 Software and tools used

- TeXstudio v2.11.2 (<http://www.texstudio.org/>) to produce this document.
- Evolus Pencil v2.0.5 (<http://pencil.evolus.vn/>) to generate mockups.
- Astah Professional 7.1.0 (<http://astah.net/>) to create Use Cases Diagrams, Sequence Diagrams, Class Diagrams and State Machine Diagrams.
- Alloy Analyzer 4.2 (<http://alloy.mit.edu/alloy/>) to verify the consistency of the model.

### 4.2 Hours of work

The time spent to redact this document:

- Bresich Matteo: 63 hours.

Days	Hours of work
27/10/16	2h
28/10/16	5h 30'
31/10/16	5h
03/11/16	3h
05/11/16	5h 30'
07/11/16	6h 30'
08/11/16	7h
09/11/16	7h
10/11/16	6h 30'
11/11/16	5h
12/11/16	2h
13/11/16	8h