

Politecnico di Milano

A.A. 2016-2017

Software Engeenering 2: PowerEnJoy

**R**equirement **A**nalysis and **S**pecification **D**ocument

Contents

[1. Introduction 4](#_Toc466570072)

[1.1. Purpose 4](#_Toc466570073)

[1.2. Scope 4](#_Toc466570074)

[1.3. Glossary: 5](#_Toc466570075)

[1.3.1 Definitions 5](#_Toc466570076)

[1.3.2. Acronyms 5](#_Toc466570077)

[1.3. User characteristics 5](#_Toc466570078)

[1.4. Assumptions 5](#_Toc466570079)

[1.4.2. General Assumptions 5](#_Toc466570080)

[1.4.2. Invoicing Assumptions 6](#_Toc466570081)

[1.6.3. Domain Assumptions 6](#_Toc466570082)

[1.7. Constraints 6](#_Toc466570083)

[1.8. Reference documents 7](#_Toc466570084)

[1.9. Overview 7](#_Toc466570085)

[2. Specific Requirements 7](#_Toc466570086)

[2.1. Functional Requirements 7](#_Toc466570087)

[2.1.1. [G1] Users must be able to register to the system by providing their credentials and payment information. 8](#_Toc466570088)

[2.1.2. [G2] Registered users must be able to find the locations of available cars within a certain distance from their current location or from a specified address. 8](#_Toc466570089)

[2.1.3. [G3] Users must be able to reserve cars. 9](#_Toc466570090)

[2.1.4. [G4] A user must be able to enter in the car reserved by him when nearby. 9](#_Toc466570091)

[2.1.5. [G5] The user is able to know the current charge of the ride. 9](#_Toc466570092)

[2.1.6. [G6] The system is able to calculate the final cost of the rental. 10](#_Toc466570093)

[2.1.7. [G7] The system is able to know when a ride ends. 10](#_Toc466570094)

[2.1.8. [G8] The system is able to charge the monthly invoice to the user’s payment method 10](#_Toc466570095)

[2.2. Non-Functional Requirements 11](#_Toc466570096)

[2.2.1 User Interfaces: Login 11](#_Toc466570097)

[2.2.2 User Interfaces: Registration 12](#_Toc466570098)

[2.1.4. User Interfaces: Search & Reserve 13](#_Toc466570099)

[2.1.5 User Interfaces: map and cars 13](#_Toc466570100)

[2.1.6 User Interfaces: mobile application login and settings 14](#_Toc466570101)

[2.1.7 User Interfaces: mobile application, rental stage 15](#_Toc466570102)

[2.1.8 User Interfaces: mobile application, end of rental 16](#_Toc466570103)

[2.1.8. User Interfaces: mobile application, unlock the car 17](#_Toc466570104)

[2.1.9. Usability 17](#_Toc466570105)

[2.1.10. Privacy requirements 18](#_Toc466570106)

[3. Software Design – UML 18](#_Toc466570107)

[3.1. Actors definition 18](#_Toc466570108)

[3.2. Possible scenarios 18](#_Toc466570109)

[3.7. Domain Model 19](#_Toc466570110)

[3.7.1. Domain Class Diagram 19](#_Toc466570111)

[3.7.2. Main dynamics of the system 20](#_Toc466570112)

[3.7.3. Main sequences of the system 20](#_Toc466570113)

[3.1.4. State Chart Diagram 22](#_Toc466570114)

[4. Model Coherence Analysis 24](#_Toc466570115)

[4.1. Alloy Code 24](#_Toc466570116)

[4.2. Generated World 24](#_Toc466570117)

# Introduction

## Purpose

This document represents the Requirement Analysis and Speciﬁcation Document (RASD). RASD enables to analyse and formalize the real need of customers and to show the constraints and the limit of the software. Documentation spots functional and non-functional requirements in order to define the system, the domain of the problem and their interactions. Moreover, it defines a baseline for project planning and estimation and may also be legally binding. Requirements take care about the customer’s needs or stakeholders’ wishes that could change over the time. Programmers and developers base their work on this document.

## Scope

The aim of the project is to create a new online car-sharing system using only electric cars. Users register themselves online providing their credentials (name, surname, email and others) and selecting a method of payment. After that they receive back a personal password that can be used to access to the system together with the associated email. Logged user is able to look for cars either around his position or a certain address inserted, selecting from different ranges of distance. Then he can choose and reserve one car from the list proposed by the system, which must be reached within an hour. If the user does not unlock the car in one hour from the reservation, he will be charged a fee of 1 EUR. When the user is inside a radius of 20 meters from the reserved car he could ask to the system to unlock the car. Then the system will check the distance between the car and the user, who must be the owner of the reservation.

As soon as the engines ignites due to a push of the start/stop button, the system starts charging the user for a fee of 0,25 EUR/min

All the electric cars are equipped with a GPS navigation device, where appears the percentage of the battery and the updated amount he is paying.

Every time the user stops the car, pushing the start/stop button, on the GPS display appears a button which terminates the ride and recap the total cost. If the user selects to finish the ride, after he will have exit, the system will lock the car and set it as available again.

Users can park anywhere inside the safe areas, which may contain a special parking area with a power grid station.

A map with all the safe areas is available on the website.

In order to facilitate the parking, safe areas are whole cities (such as the metropolitan area of Milan), so users spend less time to find a suitable place.

## Glossary:

User: he/she is the client of the service; He/she is able to rent a car in order to travel around the city. He is associated with :

-Name

-Surname

-Other personal information

-Method of payment

-Number of driving licence and expiration date

-Password

Method of payment: is inserted by the user during the registration phase but can be updated over the time. Only one method is active at once and payment are concluded using services offered by the different companies holding the credit card. An invoice containing all the charges collected is generated monthly.

Car: sometimes referred as vehicle is the means of transport rented by users. It contains a set of sensors that analyse the number of passengers presents on the car, control the charge of the battery  and detect when a door is closed. Moreover, it includes a module that transmit this information to the system using the Internet.

Available car: car that is not in use at the moment by any user, has at least 20% of charge and is not reserved by anyone.

Reservation: made by a user that want to use a car. Has a duration of one hour maximum and is associated with a unique car. Once the user ask to unlock the car the car becomes associated to the user until he decides to end the ride.

Charge: amount of money that users have to pay due to the use of the service. It is immediately calculated by the system after a ride but money is transferred only at the end of the month.

Penalty: fee derived from a bad behaving of the use such as a damage on the car or a fine for exceeding speed limits. The fee will be notified to the user and included in the monthly invoice.

GPS navigation device: system that equip each car and that is able to calculate the exact position of the car and display to the user the route to follow. Its display is also used to show the current fee of the ride and the status of the battery.

Start/stop button: device present in all the electric cars that allow the engine to ignite when the car is unlocked. It also allow the engine to stop when the user wants to get off the car.

Power grid station: special safe areas that contain plugs that allow cars to be recharged. They are provided with sensors that detect the number of free spots and communicate the number to the system. They are also called special parking areas.

Safe Area: Space included in boundaries that determine where users can park a car. It covers entire metropolitan cities in order to facilitate users to find a park and they may also contain power grid stations. Users cannot terminate a ride while outside from a safe area.

Ride/Rental: it last from when the user pick up the car until when the system stop charging the user. It includes a possible set of temporary stops and the total path travelled by the user.

Distance: is the amount of meter in a straight line between two points.

Park: is when a user leave the car and want to end the rental. At this point the system stop charging the user.

Stop: is when a user leave the car but wants to resume the ride in the future. The car will be locked by the system that, however, will continue to charge the user for the ride.

[Gn]: n-goal.

[Rn]: n-functional requirement.

[Dn]: n-domain assumption.

### 1.3.1 Definitions

### Acronyms

## User characteristics

## Assumptions

In this paragraph, are described some assumptions so as to define an unambiguous system. These assumptions will not be violated or changed during all the documentations.

### General Assumptions

These assumptions will not be violated or changed during all the documentations. They concerning the global aspects of the system.

* + - * Each user is not able to make nested rental of different cars during the same range of time.
* A single monthly invoice, which summarize all rentals and fees, is sent to the user in order to get the payment.
* Each car of PowerEnJoy has an interactive display inside the car.
* The employees of PowerEnJoy have the access to all the cars of the system.
* the employees are able to modify the status of each car if it is necessary.
* Plugging the car inside a special safe area for recharging the car is free of charge.
* Plugging the car not in a safe area for recharging the car is not free of charge.
* The user, who terminated successfully the procedure of rental online, are not able to cancel the rental.

### 1.4.2. Invoicing Assumptions

At end of each rental, the user will get a summary email containing all the information related the last rental.

The monthly invoice will show the debits during the month, specifying the use and any additional costs (for example, Penalties). In particular, after having carefully examined the case and ascertained the user's involvement, the system eventually will notify the user via email of the amount of penalty. The invoice will be issued in electronic format and can be downloaded from the user’s profile. Following this notification, the amount will be charged to the credit card or pre-paid credit card registered by the user.

These are example of penalties:

* administrative sanctions (e.g. due to traffic violations);
* service call-out (e.g. non-routine cleaning if you leave the vehicle too dirty; or if you leave it with the lights on);
* other (e.g. failure to report an accident, riding abroad);

### Domain Assumptions

## Constraints

Regulatory Policy:

-Privacy of all registered people must be granted. All the data will remain stored in the database of the system and will not be shared to third parts.

-Each user registered to the system has to provide a valid driving licence.

-Each user has to be associated with at least one valid credit card.

Hardware Limitation:

-Each user has to have a device able to know its position and send it to the system when unlocking the car.

-Each device has to be connected to the internet when interacting with the application.

-Each car has to maintain an internet connection in order to allow the system to lock and unlock it correctly.

-GPS presents on the devices used by user should be able to provide a decent position that enable the system to verify the distance between the user and the car-

-GPS presents on car should be able to place them in safe areas correctly.

-If the device of a user do not support the mobile application, the user can still use the application through the browser.

Interface with other application:

-The application take advantage of the services provided by Google Maps in order to calculate the route to arrive to the final destination and place correctly the safe areas on the map.

-The application will interface to a dbms in order to store information about the users and the records of the rides

-The application will use services provided by credit cards companies to send the monthly invoice to users.

## Reference documents

## Overview

## Specific Requirements

## Functional Requirements

Starting from the domain properties (

### 2.1.1. [G1] Users must be able to register to the system by providing their credentials and payment information.

* [R1] Visitor must not be already registered to perform registration process.
* [R2] Username must be unique.
* [R3] An online form allows users to submit their credentials.
* [R4] The user can access to the system with a password.
* [R5] The system sends a password to the user after mail confirmation.
* [R6] The system sends a mail after the registration in order to verify the user’s email address.
* [R7] The system before sending the confirmation mail must check the payment methods.
* [R8] The system provides a captcha inside the registration page in order to avoid bot attacks.
* [D1] User email address must be valid.
* [D2] Method of payment used for registration must be correct.

### 2.1.2. [G2] Registered users must be able to find the locations of available cars within a certain distance from their current location or from a specified address.

* [R1] The system has the position of all cars.
* [R2] The system knows the level of charge of all cars.
* [R3] The system is able to acquire the user’s position, if needed.
* [R4] The system provides a form where the user can insert a specified address and the radius of the search.
* [R5] The set of safe areas is associated with geographical coordinates.
* [D1] Each car has a GPS connected with the system.
* [D2] The system has a geographic map of the area.

### 2.1.3. [G3] Users must be able to reserve cars.

* [R1] The system provides a form where the user can select one car for his rental.
* [R2] The system has a database in order to store all the rentals.
* [R3] The user is able to select only one car at once.
* [R4] Each car can be rented at once only by one person at maximum.
* [R5] The user’s last monthly invoice must be paid before the reservation.
* [D1] The user can rent only one car in the same range of time.
* [D2] All economic transactions end correctly
* [D3] It will be never possible that if a user takes more than one hour to pick-up his car, the system will not charge a fee of 1 EUR.
* [D4] The user can rent only one car for each rental.

### 2.1.4. [G4] A user must be able to enter in the car reserved by him when nearby.

* [R1] The system can have the remote control of lock and unlock of all cars.
* [R2] The system must check the position of the user before unlocking the car.
* [R3] The application can send information from the user to the system.
* [

R4] The user needs to provide his real and accurate position to the system.

* [D1] The user is connected to internet.
* [D2] The user has the access to the application.

### 2.1.5. [G5] The user is able to know the current charge of the ride.

* [R1] Each car has an interactive display.
* [R2] The car is able to notify the system when the engine ignites.
* [R3] The car must be unlocked before that the engine ignites.
* [R4] The system is able to calculate dynamically the amount of charge.
* [D1] Each car provides a start/stop button.

### 2.1.6. [G6] The system is able to calculate the final cost of the rental.

* [R1] The system is able to calculate discounts
* [R2] The system is able to know the number of passengers inside the car.
* [R3] The system is able to know the number of cars inside each special safe area.

### 2.1.7. [G7] The system is able to know when a ride ends.

* [R1] The car is able to understand if a user leaves the car.
* [R2] The system is able to understand if the car is parked in a safe area.
* [R3] The car is able to communicate with the system.
* [R4] The system is able to lock the car automatically after it stops charging the user.
* [D1] There is always a connection between the car and the system.

### 2.1.8. [G8] The system is able to charge the monthly invoice to the user’s payment method

* [R1] The system must know all the rentals of each user.
* [R2] The system is able to summarize all the costs related to the user in one rental.

## Non-Functional Requirements

### 2.2.1 User Interfaces: Login

The mock-up above shows the home page; it appears as soon as the user reach the website. The home page allows the users not only to login inside the website but also to register a new profile. A small description of the features of this car-sharing service is located on the right of the page.

*Figure 1*

### 2.2.2 User Interfaces: Registration

As mentioned previously the user can access to the registration page from the home page. This page is composed of a set of forms, the user needs to fulfil all the forms in the page in order to register his profile.

*Figure 2*

### 2.1.4. User Interfaces: Search & Reserve

After the registration (or login) the user can search and reserve a car in this web-page. It is possible to find a car in two different ways, from their current position (with the user’s GPS) or from a specified address. The user can also define the radius in which he is able to find cars.

*Figure 3*

### 2.1.5 User Interfaces: map and cars

The system sends to the user’s client the information which he was looking for. The map contains all the information that the user needs, inside the user can see all the cars inside the radius of the point he has defined but also his position. The labels inside the map spot the position of cars. On the right of the page, you can see a small legend in which the user can see the mean of colours of labels.

*Figure 3*

### 2.1.6 User Interfaces: mobile application login and settings

These two mock-ups represent the login page and the settings panel for mobile application. User can access to the features of the application only after the login.

*Figure 4 Figure 5*

### 2.1.7 User Interfaces: mobile application, rental stage

After login the user can rent a car. In the picture on the right the user is able only to rent a car, but after the application unlock the other functionalities (picture on the left).

The functionalities of application are:

* Reserve a car: the user can rent a car
* Car’s position: find the position rented car on the map
* Unlock the car: iff the user is nearby a car he is able to ask for unlocking the car.
* End the rental: iff the user is out of the car and the car is parked in a safe area he is able to end the rental of a car.

*Figure 6 Figure 7*

### 2.1.8 User Interfaces: mobile application, end of rental

At the end of the rental, the user receives a notification on his application. The notification contains all the information related to the last ride. The user can know each detail of his rental like an invoice (time, distance and expanse).

*Figure 8*

### 2.1.8. User Interfaces: mobile application, unlock the car

This error will appear iff the user tries to unlock the car but he is not nearby the car.

*Figure 9*

### 2.1.9. Usability

The choice of extending safe areas to entire cities facilitates users while looking for a park, in fact, in this way they can use all the parking opportunities offered by the city. Moreover, the minimalism of the interface of the application allows user to speed up the processes of renting a car and lock/unlock it. The application will be released in the three main app market (Windows Store, App store, Google Play) and allow users to manage multiple accounts on the same phone. In order to avoid people stuck in the cars with the battery of their phone at 0% we provide USB power plugs within the cars so that users can recharge their phone and end the ride through the application.

### 2.1.10. Privacy requirements

The system will protect users’ personal data by storing them in safe serves. In particular, Login credentials are encrypted by using hash functions.

All the information related to rides will be private and won’t be shared.

# Software Design – UML

## Actors definition

## Possible scenarios

Scenario 2

Mark is an Australian tourist that is visiting Milan, with his family, for the well-known fashion week. Unfortunately, the train from the airport arrives in the Milan central station at 1 a.m. and therefore he has no idea about how reach the city centre. He would like to avoid the high fees of a taxi and at the same time respect the environment; that is why he chose PowerEnJoy. After logging in on the web site he looks for the nearest car and gets the one parked outside from the station and discovers the GPS navigator service present on the car. This engaging feature allows Mark to find the closest path to the centre and at the same time the closest power grid station where parking. Mark parks the car at the special safe area and connect the car to the power plug. The system notice that and consequently apply the 30% discount on the ride and moreover an additional 10% discount is added due to the presence of more than two passengers detected by the sensors.

Scenario 3

Massimiliano is arriving by train to Rome and decides to plan his visit by booking a car on the PowerEnJoy system. However, due to a problem of an engine, the train arrives at the station with a delay of one hour. Therefore, Massimiliano is not able to get the car within one hour from the reservation that expires. The user is charged of a fee of 1 euro and the car becomes available again.

Scenario 4

Mr.Jones is a rich tourist, who loves cars well designed. During the month of October, he needed to reach several times the Milan Cathedral situated in the centre of Milan. The access to the historical centre of Milan is limited by the Congestion Charge area (Area C) from Monday to Friday. Motorcycles and scooters, electric vehicles are exempted from payments. So Mr.Jones used PowerEnJoy. Mr.Jones is also a cheapskate person, in order to save money he left the car with more than 50% of battery because the system applies a discount of 20% on the last ride.

Scenario 5

Mrs.Robinson’s grandparents lives in Italy. Twice a year she wants to visit his grandparents, but they live far from city centre, so she can’t use subway or buses to reach his grandparents. For this reason, she uses PowerEnJoy. They live 3 km from the nearest power grid station and far from the railway station. So the system charges 30% on her the last ride more to compensate for the cost required to recharge the car on site. The same would have happen if she arrives at their home with less than 20% of battery.

## 3.7. Domain Model

### 3.7.1. Domain Class Diagram

### 3.7.2. Main dynamics of the system

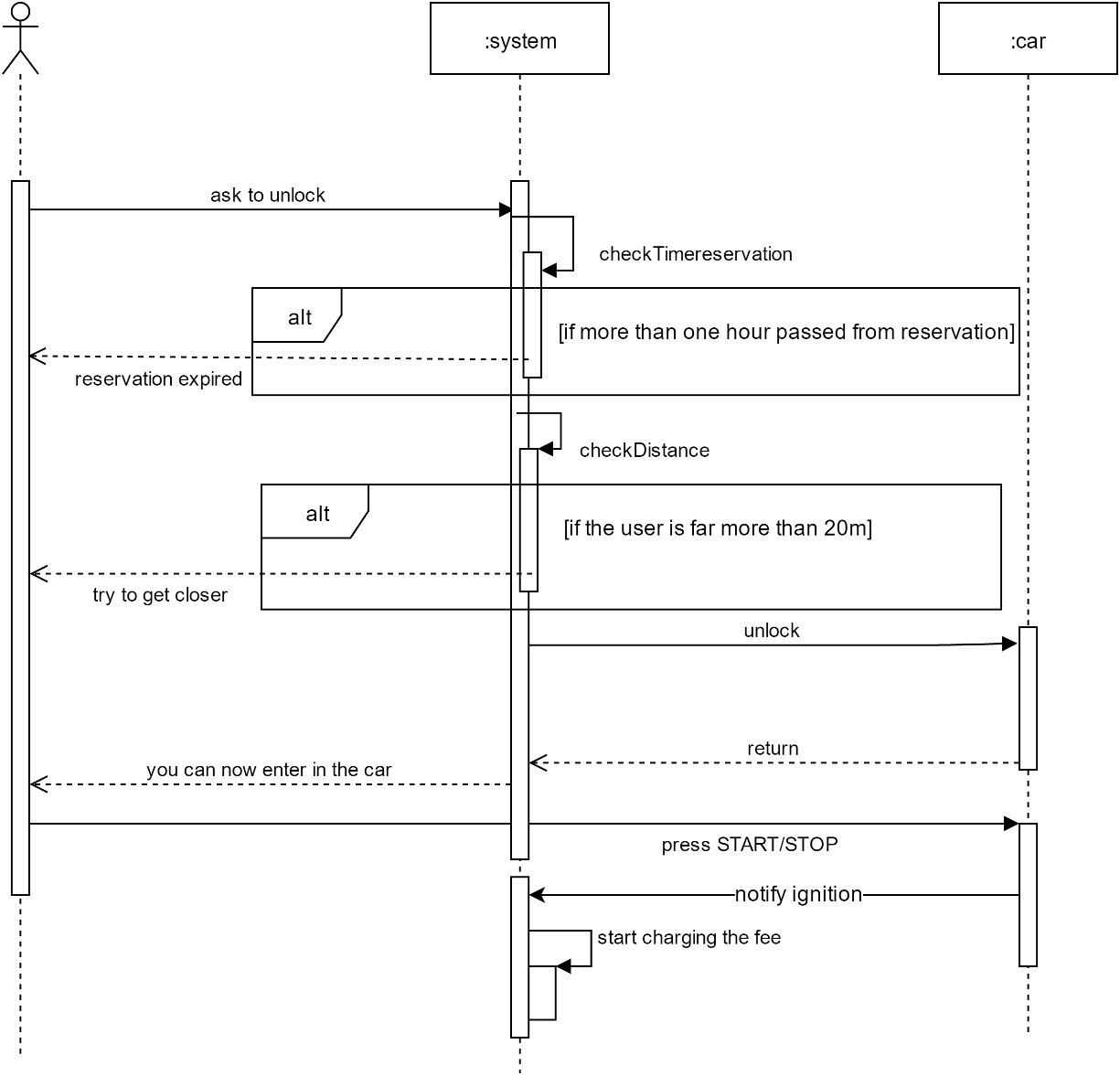
### 3.7.3. Main sequences of the system

*Sign Up Sequence Diagram*

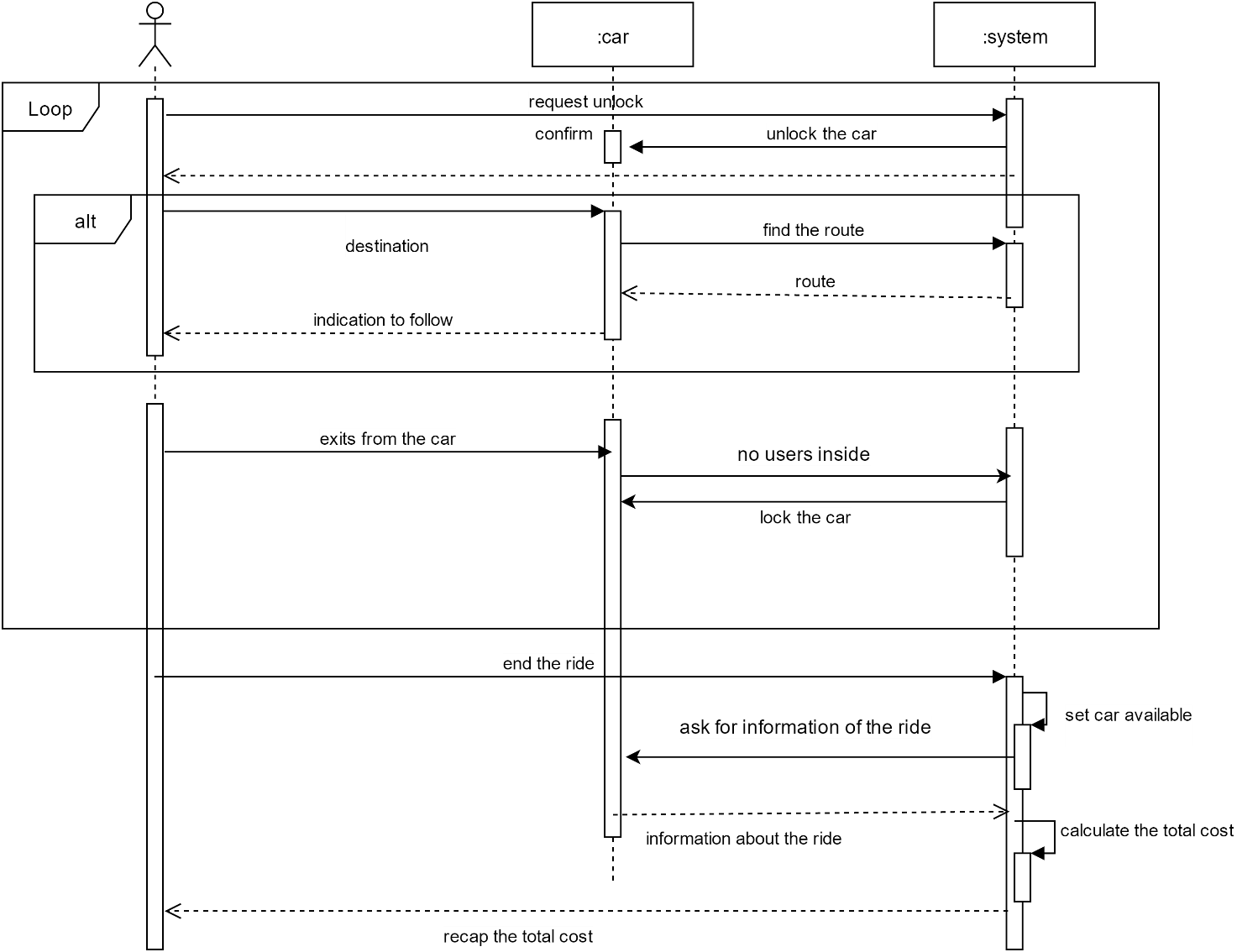
The following Sequence Diagram represents the Sign Up functionality. First of all, the user reaches the web-page. After the fulfilment of all the forms, the user sends the page to the website. The website turns in the data to the system, and he will check the payment method. As we can see there is a loop that holds until the user fill properly all the forms. When the system has successfully confirmed the payment method, the users is able to reach the next web page.

C:\Users\defi9\Downloads\Untitled Diagram (6).png

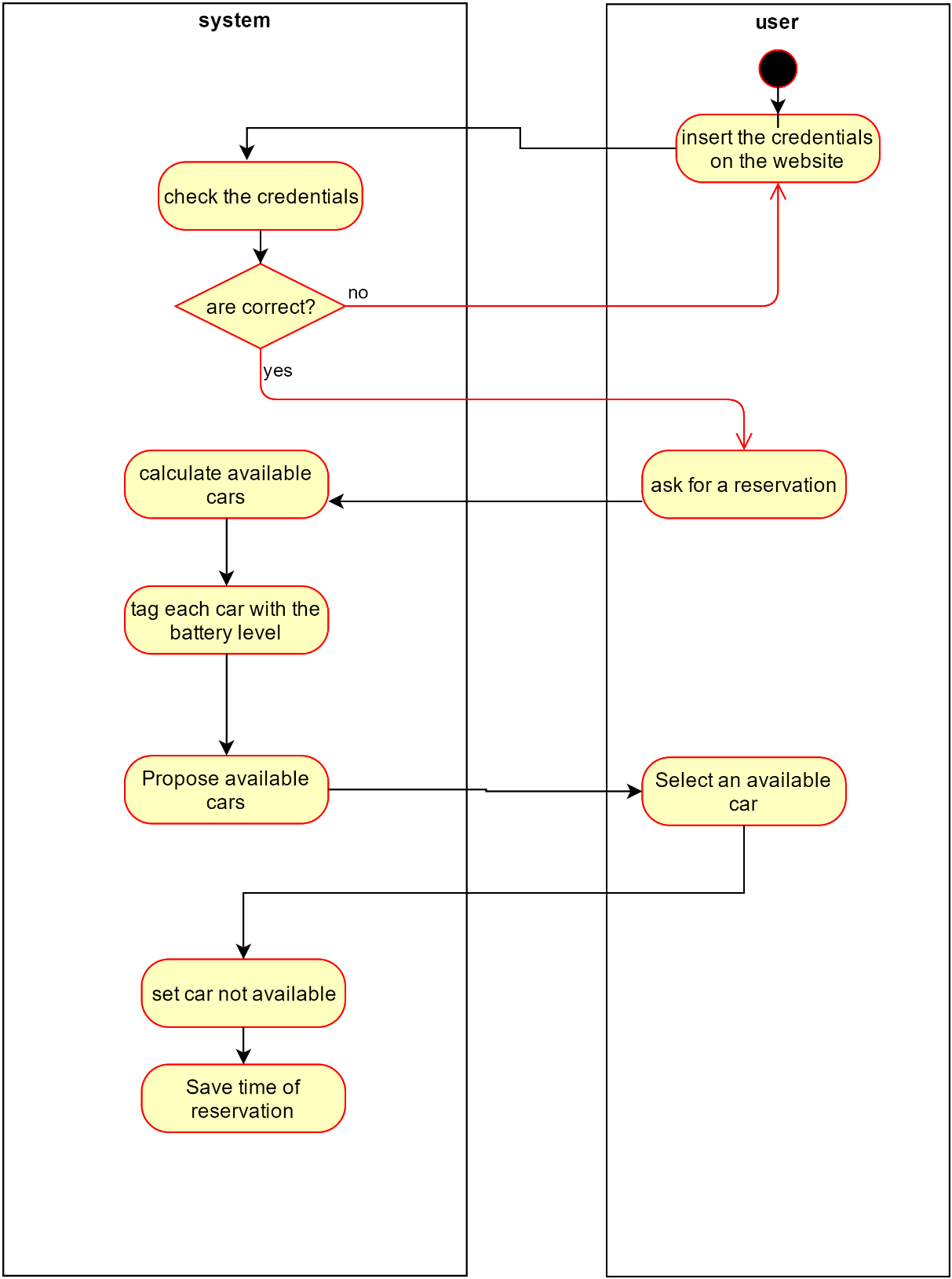
Once the user has reached the car ask to the system, through the app, to unlock the car. The system check the distance between the user(position provided by the gps on his device) and the car itself. If the distance is under 20m and the request arrives within one hour from the reservation the car is unlocked and the user can get on. By simply pressing the start/stop button present on the car the engine ignites and the car send to the system the actual time in order to calculate the charge of the ride.



The diagram shows the interaction between the entities of our system during a ride, from the moment the car is unlocked to the end of the ride. the user, through the app, ask the system to unlock the car that forward the request to the car. Each user can take advantage, if desired, of the money saving option offered by system. Indeed, the user insert the final destination on the GPS system present on the car that asked to the system the most suitable route to follow in order to maintain a balanced distribution of the cars in the city. After the ride is completed the user exit from the car that is locked. At this point, the user can decide to end the ride by pressing a button on the app that notify the system to set the car available again and calculate the final cost. On the contrary, the user can decide not to end the ride so when is close enough can ask the system to unlock the car again and resume his ride. However, the time spent out of the car by the user is calculated normally.



The diagram illustrates the process of car reservation performed by a user. Firstly, the user logs in the website by providing his credentials and then ask to the system the available cars; If the credentials are correct the system proposes a map containing all the available cars and the user has the possibility to choose one of them. Each car is labelled with the level of the charge and all the car that are rented or with a level of the batter below 20% are not shown. The user select the most suitable car based on the position and at this point, the reservation is completed, the car is set to unavailable and the time of the reservation is stored in order to check that the user collects the car within one hour.



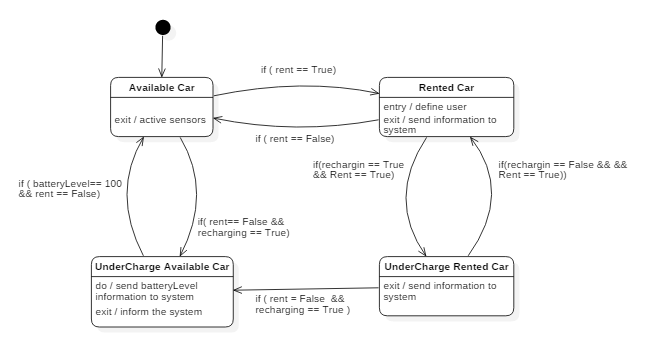
The diagram reveals the flow of activities when a user discover that his car has a problem or an incident has happened.Firstly, through the app he has the chance to know the number of the digital exchange of the system that he can call in case of emergency. An operator will be ready to answer to all the doubts related to the situation and guide the user in solving the issue. As soon as the operator realizes that a problem occurred, he terminates the ride and set the car not available for other users. At this point he decides if a mechanic is needed for the situation and in case he calls him. Finally, the operator sends detailed information to the user about what to do and in case of an incident the rules to follow and where the useful documents(e.g. agreed motor accident statement) can be found.

### C:\Users\Lorenzo Frigerio\AppData\Local\Microsoft\Windows\INetCacheContent.Word\OperatorCall.png

### 3.1.4. State Chart Diagram

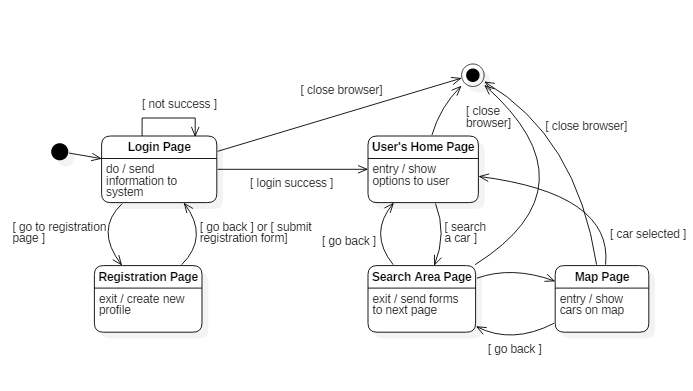
*Car States*

The following diagram would give the information related the possible states in which the car could be. First of all, the car is available and any user is able to rent it. After the rental the car changes its state. In this new state the user can recharge the car or end the rental. During the rental any recharge activities inside a special safe area is free of charge, but outside the special safe area the user needs to pay at their own expense. As we can see from the diagram the car can be recharged also during the available state.



*Browser Application States*

The following diagram would give the information related the possible states of the web application of PowerEnJoy service. The diagram begins when one user which reaches the website. In the website the user can find all the information related the service but also a login form. If the user hasn’t a profile, he can create one in the “Registration page”. After the login several options, related to the service, are shown to the user. The user can select “search a car” from the “User’s Home Page”. In the “Search Area page” the user can decide to use his position or an address in order to search a car. After that a “Map page”, which contains all the car inside a specified radius.



# Model Coherence Analysis

## Alloy Code

## Generated World