



POLITECNICO MILANO 1863

Computer Science and Engineering

A.A. 2016/2017

Software Engineering 2 Project:

“PowerEnJoy”

Requirements **A**nalysis and **S**pecification **D**ocument

November 13, 2016

Prof.Luca Mottola

Matteo Michele Piazzolla Matr. 878554

Andrea Millimaggi Matr. 876062

Revision History

Revision	Date	Author(s)	Description
1.0	13/11/16	Piazzolla Millimaggi	First document issue

Contents

1	Introduction	6
1.1	Purpose	6
1.2	Scope	6
1.3	Definitions, Acronyms, Abbreviations	6
1.3.1	Definitions	6
1.3.2	Acronyms	6
1.3.3	Abbreviations	7
1.4	Reference documents	7
1.5	Document Overview	7
1.6	Stakeholders	7
1.7	Actors	7
1.8	Goals	8
2	Overall Description	9
2.1	Product perspective	9
2.1.1	System interfaces	9
2.1.2	User interfaces	9
2.1.3	Hardware interfaces	9
2.1.4	Software interfaces	9
2.2	User characteristics	10
2.3	Constraints	10
2.4	Assumptions	10
2.5	Regulatory policies	10
2.5.1	Possible Future Implementations	11
3	Specific Requirements	12
3.1	Functional Requirement	12
3.1.1	[G1]Allow guests to register to "PowerEnJoy" service through both website or mobile application.	12
3.1.2	[G2]Allow registered users to login and logout.	12
3.1.3	[G3]Allow authenticated users to modify their personal details and payment informations.	13
3.1.4	[G4]Allow authenticated users to view a list or a map of available cars, safe areas and special parking areas in a configurable radius of distance from user position or specified address.	13
3.1.5	[G5]Allow authenticated users to request for pick up an available car for a limited time and unlock it when the User is in proximity.	13
3.1.6	[G6]Allow authenticated users to drive the unlocked car inside or outside the border of the city.	13
3.1.7	[G7]Allow user to be constantly informed about the service through a screen on the car.	14

3.1.8	[G8] Allow the User to get a modified fee in base of pick up/parking position of the car, number of passengers and status of the battery.	14
3.1.9	[G9] Allow the user to conclude his travel and stop being charged after he turns off and exits the car.	14
3.1.10	[G10] Allow the user to put the car in charge when he leaves it in a special parking area.	15
3.2	Non-Functional Requirement	15
3.2.1	Reliability	15
3.2.2	Security	15
3.2.3	Usability	15
3.3	Use Case	18
3.3.1	Use Case Diagram	18
3.3.2	User Registration	19
3.3.3	User Login	20
3.3.4	Modification of personal details and payment information	21
3.3.5	Search of available cars	22
3.3.6	Request and Unlock	23
3.3.7	Driving	24
4	Sequence Diagrams	25
5	Scenarios	30
5.1	Scenario 1	30
5.2	Scenario 2	30
5.3	Scenario 3	30
5.4	Scenario 4	30
6	Activity Diagram	31
6.1	PowerEnJoy Trip	31
7	Class Diagram	32
7.0.1	PowerEnJoy server	32
8	Alloy	33
9	Appendix	38
9.1	Used software	38
9.2	Time effort	38

List of Figures

1	PowerEnJoy website homepage	16
2	PowerEnJoy Mobile Application map view	17
3	General Use Case Diagram	18
4	User Registration Sequence Diagram (Use Case 3.3.2)	25
5	User Login Sequence Diagram (Use Case 3.3.3)	26
6	Personal information update Sequence Diagram (Use Case 3.3.4)	27
7	Common trip case Sequence Diagram (Use Cases 3.3.6,3.3.7)	28
8	Common trip case Sequence Diagram (Use Cases 3.3.6,3.3.7)	29
9	PowerEnJoy normal flow	31
10	PowerEnJoy Server Class Diagram	32
11	World Tree generated by Alloy Analyzer 4.2	37

1 Introduction

1.1 Purpose

This document is the RASD (Requirement Analysis and Specification Document). The goal of this document is to describe the functional and non-functional requirements of the system in order to show the constraints on the software that the developer has to take into account. This document can have a legal value and can be used as a contractual basis between the customers and the developers.

1.2 Scope

The system the has to be developed is called *PowerEnJoy*. It is a digital system for the management of a car-sharing service that only employs electric cars. In particular the aim is to develop a mobile application that allows the user to pick up and use electric cars in the areas reached by the service.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- User: is a general person who uses “PowerEnJoy”.
- Registered User: a user that have completed the registration process.
- Guest: a visitor, not authenticated user.
- Car reservation: allocation of a car for a defined amount of time.
- Safe area: area in which the User can find cars, pick them up and park them.
- Special parking area: area in which the user can park the car and also charge it plugging it into the power grid.
- City Area: area delimited from the boundaries of the city. It contains Safe and Special parking areas.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- DB: Database.
- API: Application Programming Interface.
- GPS: Global Positioning System,

1.3.3 Abbreviations

- [Gn]: n-goal.
- [R-n]: n-requirement.

1.4 Reference documents

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- IEEE Std 1016 tm -2009 Standard for Information Technology-System Design-Software Design Descriptions.
- ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering

1.5 Document Overview

The structure of this document follows the IEEE standard for the editing of a RASD document. In particular the structure is:

- Introduction: it provides a general description of the document and the system that has to be developed
- Overall Description: it provides a general description of all the things that affects the work on the project and the elaboration of the requirements. So it gives information about hardware and software choices and availability, the main functions of the product, the description of the people at whom the product is addressed, the constraints that will limit the work of the developer, the assumptions done before the development and the dependencies that the use of the product will have from hardware and software entities
- Specific Requirements: it provides a detailed description of the requirements of the product

1.6 Stakeholders

The primary stakeholder is any person who need to use some type of transportation, whether it is personal vehicle or public transportation like buses and railway trains to move inside the city.

1.7 Actors

The entities identified in the problem description are the following:

- GUEST: is a visitor, someone that isn't signed-up yet. He can visualize "PowerEnJoy" Web site and download the mobile app but he cannot access to any service.

- USER: is a client signed-up, via Web or mobile app. He accesses "Passenger Area" and, after being successfully logged in, he can see his personal section, modify his personal information. He can also access to all services, view a map of available cars, make car reservations, modify or cancel them.

1.8 Goals

- [G1] Allow guests to register to "PowerEnJoy" service through both website or mobile application.
- [G2] Allow registered users to login and logout.
- [G3] Allow authenticated users to modify their personal details and payment informations.
- [G4] Allow authenticated users to view a list or a map of available cars, safe areas and special parking areas in a configurable radius of distance from user position or specified address.
- [G5] Allow authenticated users to request for pick up an available car for a limited time and unlock it when the User is in proximity.
- [G6] Allow authenticated users to drive the unlocked car inside or outside the border of the city.
- [G7] Allow user to be constantly informed about the service through a screen on the car.
- [G8] Allow the User to get a modified fee in base of pick up/parking position of the car, number of passengers and status of the battery.
- [G9] Allow the user to conclude his travel and stop being charged after he turns off and exits the car.
- [G10] Allow the user to put the car in charge when he leaves it in a special parking area.

2 Overall Description

2.1 Product perspective

The system that we are going to develop is made of several parts.

- A Mobile Application usable from every Tablet or Smartphone that has access to an internet connection. The application will be available on Android and iOS.
- A Website where a user can find information about the service.
- A Server backend to manage the service.
- An onboard device installed on the car to manage unlocking and real-time GPS position,

2.1.1 System interfaces

The system will depend on a device installed onboard every 'PowerEnjoy car. That onboard device will communicate with the user and via internet to the server.

2.1.2 User interfaces

The available interfaces will be:

- the mobile application
- the website

The website will be implemented with a responsive design to adapt to all most common screen aspect and resolution with clear and minimal UI to favorite accessibility. The mobile application will be graphically similar to the website with a few views. The use of the website or the mobile application won't be necessary during driving.

2.1.3 Hardware interfaces

The main hardware interaction concerns the geolocalization feature, and in particular the GPS hardware on the smartphone device. The application need an internet connection so a smart device with a 3G or LTE connectivity is required.

2.1.4 Software interfaces

To provide the best use of the service through the website an HTML5 compliant browser is required and an up-to-date version of Android or iOS operating system for the mobile application.

2.2 User characteristics

Everyone who has a car license will be able to register to the car-sharing service and drive one of the electric car available. Because of the variety of people that will use the service, the mobile application must be simple and user-friendly, so that anyone can use it even if it has a little knowledge about the use of a mobile device.

2.3 Constraints

The mobile implementation of the system need to be lightweight to allow compatibility for every range of smartphone. The application must communicate with a small amount of data via 3G or LTE and must be small in terms of size to allow a User to download it without a Wi-Fi connection. The website must be reactive and usable with the major version of the most common web browser.

2.4 Assumptions

1. Cars can be parked in every area of the city where the parking is allowed, either it is free or chargeable. The company takes care of the payment for chargeable parkings.
2. Every car can be uniquely recognized by his plate number.
3. The position of the users are well-known thanks to GPS.
4. The user is the one and only who drives the cars he reserves.
5. Any user who reserves a car has enough money on his credit card to pay the travel.
6. Every car is periodically checked to ensure proper operation.
7. The travel of a user is considered finished when the the engine is off and after a minute from the moment in which he closes the car doors.
8. PowerEnJoy offers various models of cars with different engine sizes

2.5 Regulatory policies

It's user responsibility to ensure that the use of the system complies with the local laws and policies. If the user register to the service must allow for the permission to acquire, store and process personal data and web cookies. The system must offer to the user the possibility to delete the account and all the personal data.

2.5.1 Possible Future Implementations

Depending on the success that the developed system will have, it will be possible to extend the car-sharing to Scooters and Mini-Cars, in order to reach also younger customers.

Another future improvement could be the possibility for the users share a trip with strangers and split the charge as other services like BlaBlaCar.

3 Specific Requirements

3.1 Functional Requirement

With these requirements, we define features and functions with which the user will interact directly.

3.1.1 [G1]Allow guests to register to "PowerEnJoy" service through both website or mobile application.

- [R-1] The system shall allow registration only if the patent license number is provided.
- [R-2] The system shall provide a home page in which a guest user must be able to know what the service is.
- [R-3] The system shall demand the User to read and accept all term of use of the service.
- [R-4] The system shall provide a form in which a Guest can insert his personal data for the registration to the service.
- [R-5] The system shall provide a form in which a Guest can insert his payment information.

3.1.2 [G2]Allow registered users to login and logout.

- [R-6] The system shall show a sign-up page with a login form and a logout button if the User is already logged in.
- [R-7] The system shall validate any input in both client and server side.
- [R-8] The system shall prevent anyone from logging more than once at a time.
- [R-9] The system shall show an error message in case of wrong credential.
- [R-10] The system shall provide a password recovery procedure.
- [R-11] The system shall prevent brute force attack limiting the number of try per IP address.
- [R-12] The system shall redirect the User to the homepage as a guest after the logout.

3.1.3 [G3]Allow authenticated users to modify their personal details and payment informations.

- [R-13] The system shall show a an account page with all the information that the User sent during account registration.
- [R-14] The system shall ask for the password before saving the edited details.
- [R-15] The system shall send and email to notify the changes.

3.1.4 [G4]Allow authenticated users to view a list or a map of available cars, safe areas and special parking areas in a configurable radius of distance from user position or specified address.

- [R-16] The system shall show on a web page or a view in the application the position of the available car.
- [R-17] The system shall allow the User to set a search radius.
- [R-18] The system shall ask the User for the GPS position or to provide an address.

3.1.5 [G5]Allow authenticated users to request for pick up an available car for a limited time and unlock it when the User is in proximity.

- [R-19] The system shall not allow the User to request multiple cars at the same time.
- [R-20] The system shall check if the payment information are valid and show an error if not.
- [R-21] The system shall automatically cancel a car reservation and make the car available again if the User don't unlock the car in one hour and charge with a 1 euro fee.
- [R-22] The system shall check for the User position and if the distance from the car is less than 50 meters allow to unlock the car.
- [R-23] The system shall allow the user to be informed about the autonomy, the battery level, the model, the engine size of the available cars

3.1.6 [G6]Allow authenticated users to drive the unlocked car inside or outside the border of the city.

- [R-24] The system shall inform the User if she is driving outside the city area.
- [R-25] In case of driving outside the city area the system shall charge a 100% more of the current charge.

3.1.7 [G7] Allow user to be constantly informed about the service through a screen on the car.

- **[R-26]** The system shall inform the User of the amount that will be charged.
- **[R-27]** The system shall inform the User of the position of charging stations nearby the current car position.
- **[R-28]** The system shall show on a map the special parking area and the safe area nearby the current car position.
- **[R-29]** The system shall inform the user if she's driving more than 3 km far from the nearest power grid station

3.1.8 [G8] Allow the User to get a modified fee in base of pick up/parking position of the car, number of passengers and status of the battery.

- **[R-30]** The system shall detect the number of passengers on the car
- **[R-31]** The system shall apply a discount of 10% on the last ride if it detects the presence of two other passengers on the car in addition to the user.
- **[R-32]** The system shall check the level of the battery when a car is left.
- **[R-33]** The system shall apply a discount of 20% on the last ride if it detects that the car was left with no more than 50% of the battery empty.
- **[R-34]** The system shall check if the user plugs the car into the power grid when he left the car in a special parking area.
- **[R-35]** The system shall apply a discount of 20% on the last ride if it detects that the user takes care of plugging the car into the power grid when he leaves the car in a special parking area.
- **[R-36]** The system shall calculate the distance from the parking position to the nearest power grid station and if that distance is more than 3 km, shall charge the 30% on the last ride.

3.1.9 [G9] Allow the user to conclude his travel and stop being charged after he turns off and exits the car.

- **[R-37]** The system shall detect the car turning off and the closure of the doors and stop charging the user after this two actions are accomplished

3.1.10 [G10] Allow the user to put the car in charge when he leaves it in a special parking area.

- **[R-38]** The system shall detect that the car is plugged into the power grid and start charging the battery

3.2 Non-Functional Requirement

- PowerEnJoy has to be available 24h/7d.
- Both mobile application and website must be reactive and usable.

3.2.1 Reliability

The service has to guarantee an 24h/7d availability. Components of the project code will be tested after the implementation phase to ensure that they are functional. All the critical software bugs found must be patched in at least 48h. If a Back-End API change occurs, it must be guaranteed a support for clients that implement older API. All the system data must be constantly backed up to assure the data recovery in case of fault.

3.2.2 Security

The system must ensure that all data is protected from unauthorized access. Password should be saved on a DB hashed and salted. Every input from the user and every request must be sanitized

3.2.3 Usability

- The User must be able to use the system with mouse, keyboard and touchscreen.
- The User must be able to choose from several language.
- At the first login the system must provide a simple and skippable 3-views guided tutorial of the service.

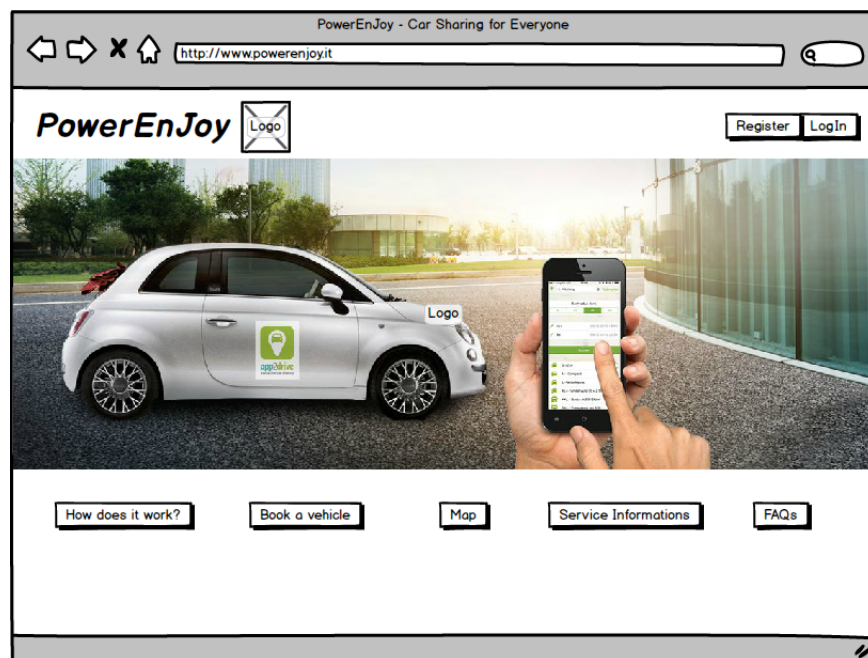


Figure 1: PowerEnJoy website homepage

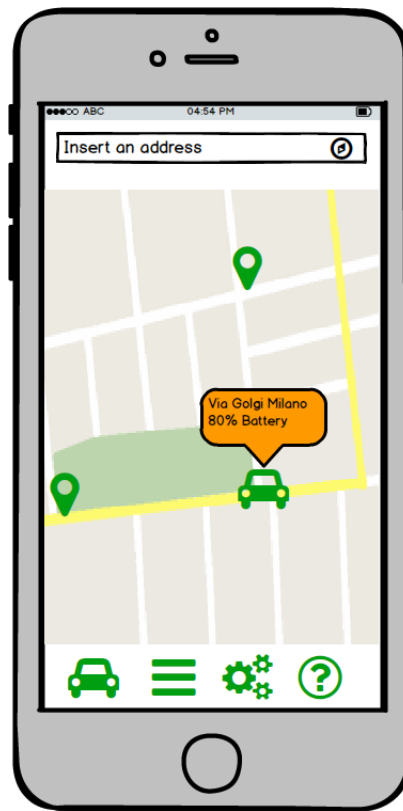


Figure 2: PowerEnJoy Mobile Application map view

3.3 Use Case

3.3.1 Use Case Diagram

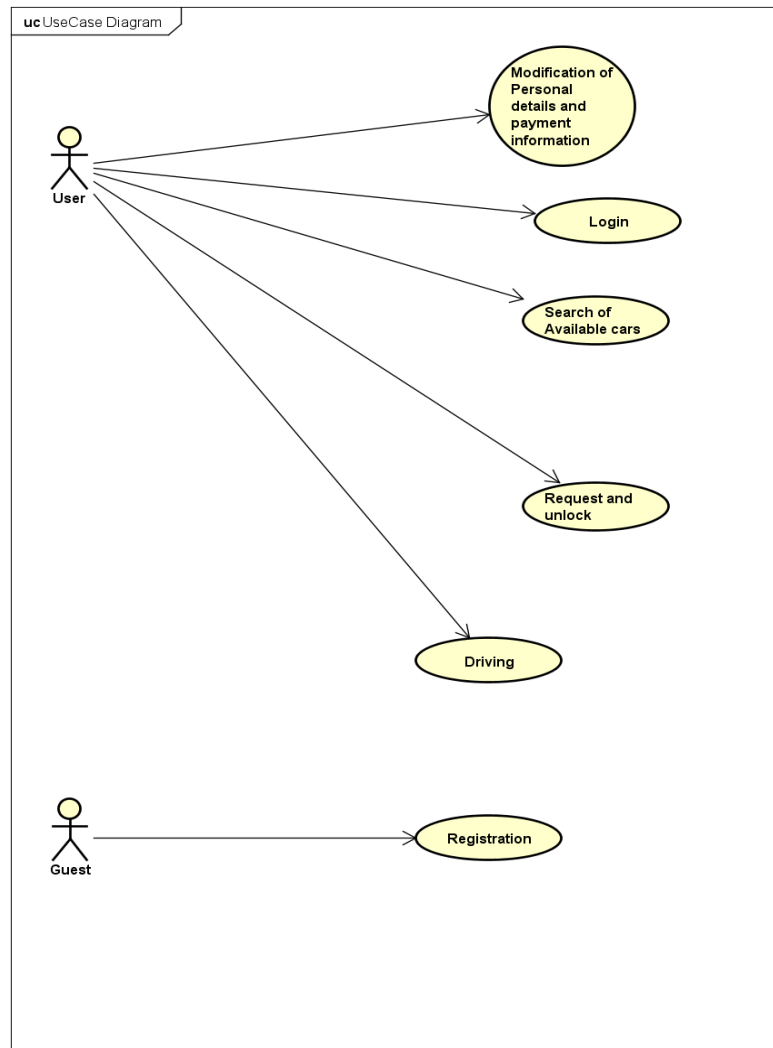


Figure 3: General Use Case Diagram

3.3.2 User Registration

Use Case Name	User Registration
Actors	Guest
Goals	[G1]Allow guests to register to "PowerEnJoy" service through both website or mobile application.
Preconditions	Guest hasn't registered to the webapp yet.
Postconditions	Guest is signed-up and promoted to "PowerEnJoy" User.
Normal Flow	<ol style="list-style-type: none">1. Guest accesses to the webapp or the mobile application2. Guest clicks on "Sign Up" button3. Guest fill Sign-up form fields, entering<ul style="list-style-type: none">-Name-Surname-Email address-Password-Telephone Number-Driving License number4. Guest accepts the term of use5. Guest clicks on "Confirm" button
Alternative Flows	-
Exceptions	<ol style="list-style-type: none">1. At least one field is empty when the guest has confirmed. The system shows again the registration form, showing in addition a message advising the guest that she missed a field.2. At least one field is invalid when the guest has confirmed. The system shows again the registration form, indicating the invalid field.3. Entered email is already associated to another account. The system shows again the registration form, showing in addition a message advising the guest that the email is already associated to another account.

3.3.3 User Login

Use Case Name	User Login
Actors	User
Goals	[G2]Allow registered users to login and logout.
Preconditions	User has signed-up as PowerEnJoy User.
Postconditions	User is redirected to personal area.
Normal Flow	<ol style="list-style-type: none">1. User accesses to the webapp or the mobile application2. User clicks on “Log-in” button3. User fills username and password fields of Log-in form4. User clicks on “Submit” button
Alternative Flows	-
Exceptions	<ol style="list-style-type: none">1. Invalid username and/or password. User is redirected by the system again to the page of Login form.2. At least one field is empty. User is redirected by the system again to the page of Login form.

3.3.4 Modification of personal details and payment information

Use Case Name	Modification of personal details and payment information
Actors	User
Goals	[G3]Allow authenticated users to modify their personal details and payment informations.
Preconditions	User is logged in
Postconditions	User is redirected to the modifications area where he can modify his personal information
Normal Flow	<ol style="list-style-type: none">1. User clicks on “Personal Details” button in his personal area. A page with the personal data of the user is shown2. User clicks on “Modify Data” button. A form for modifications is shown3. User fills fields he wants to modify4. User clicks on “Confirm” button5. User is redirected to his personal area
Alternative Flows	-
Exceptions	<ol style="list-style-type: none">1. At least one field is empty2. At least one field is invalid3. Entered email is already associated to another account

3.3.5 Search of available cars

Use Case Name	Search of available cars
Actors	User
Goals	[G4]Allow authenticated users to view a list or a map of available cars, safe areas and special parking areas in a configurable radius of distance from user position or specified address.
Preconditions	User is logged in
Postconditions	User is redirected to a page with the available cars
Normal Flow	<ol style="list-style-type: none">1. User clicks on “Search a car” button in his personal area. A form is shown2. User select a distance from the “Distance from your position” field3. User clicks on “Confirm” button4. User is redirected to a page showing the available cars
Alternative Flows	<ul style="list-style-type: none">• Different flow from step 2:<ol style="list-style-type: none">2 User write an address in the “Address” field3 User clicks on “Confirm” button4 User is redirected to a page showing the available cars
Exceptions	<ol style="list-style-type: none">1. There aren't available cars within the distance set2. There aren't available cars near the address set

3.3.6 Request and Unlock

Use Case Name	Request and Unlock
Actors	User
Goals	[G5]Allow authenticated users to request for pick up an available car for a limited time and unlock it when the User is in proximity.
Preconditions	User is logged in and has searched a car (ref. 3.3.5)
Postconditions	The car requested is reserved for the user
Normal Flow	<ol style="list-style-type: none">1. User selects a car from the list.2. A page with information about the car (battery level, model, engine size, autonomy in kilometers) is shown3. User clicks the “Request” Button4. The car is reserved for the user5. User reaches the car and is less than 3 meters far from the car6. User unlocks the car clicking on the “Unlock” button in the application
Alternative Flows	<ul style="list-style-type: none">• Steps 1 is different:<ol style="list-style-type: none">1 User select a car on the map.• Different flow from step 5:<ol style="list-style-type: none">5 An hour passes without the User unlocking the car6 The system charges the User with an extra fee of 1 euro7 The system cancels the request and makes the requested car available again
Exceptions	<ol style="list-style-type: none">1. The car selected has been requested by another user before the User has clicked on the “Request” button.

3.3.7 Driving

Use Case Name	Driving
Actors	User
Goals	[G6] Allow authenticated users to drive the unlocked car inside or outside the border of the city. [G8] Allow the User to get a modified fee in base of pick up/parking position of the car, number of passengers and status of the battery. [G9] Allow the user to conclude his travel and stop being charged after he turns off and exits the car.
Preconditions	User is logged in and has searched, requested and unlocked a car
Postconditions	-
Normal Flow	<ol style="list-style-type: none"> 1. User enters the car, eventually also other passengers enter the car 2. User starts driving 3. The system detects the number of passengers 4. The screen shows a map with the position of the car, the safe areas and special parking areas near the car position 5. The screen shows the money to be charged up to the current moment 6. The system charges the standard fee every minute. 7. User parks and exit the car and, if any, the other passengers exit the car 8. The system locks the car 9. The system detects the level of the battery 10. The system detects if the car has been plugged in a power grid 11. The system detects the position of the left car 12. The system applies a discount on the ride of the: <ul style="list-style-type: none"> • 10%, if it has detected the presence of at least two other passengers in addition to the user during the travel • 20%, if it has detected that the car was left with no more than the 50% of the battery empty 13. The system charges the 30% more on the last ride if it detects that the car has been left more than 3 km far from the nearest power grid station <p>The system takes the money from the User, using his selected payment method</p>
Alternative Flows	<ul style="list-style-type: none"> • Before the last step, the system detects the presence of at least two other passengers in addition to the user during the travel and so applies a 10% discount on the ride • Before the last step, the system detects that the car was left with no more than the 50% of the battery empty and so applies a 10% discount on the ride • At step 7 the User parks the car in a special parking area and plugs the car into the power grid. Before the last step the the system detects that the car has been plugged into the power grid and so applies a 20% discount on the ride • While driving the User exits the city area. So: <ol style="list-style-type: none"> 1. An acoustic signal is emitted by the screen 2. A message is shown on the screen, warning the User that he has exited the city area 3. The system charges the 100% more than the standard fee every minute until the user re-enters in the city area
Exceptions	-

4 Sequence Diagrams

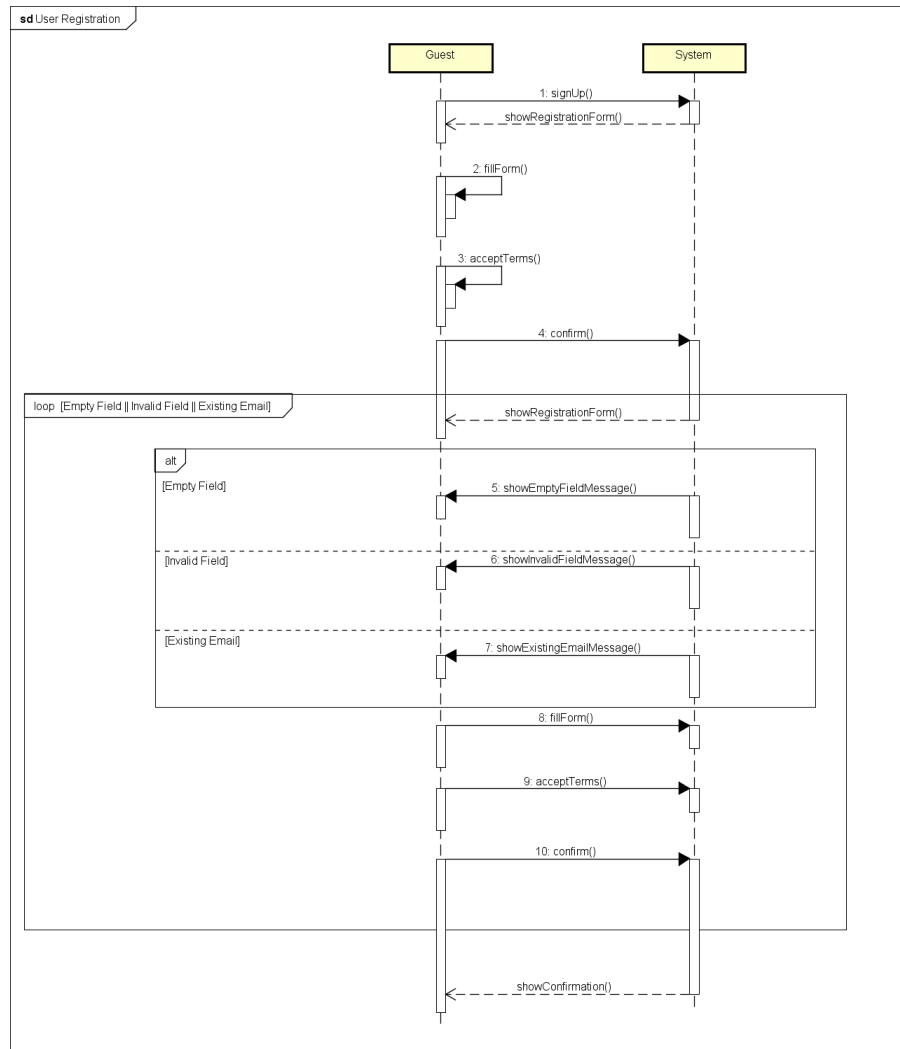
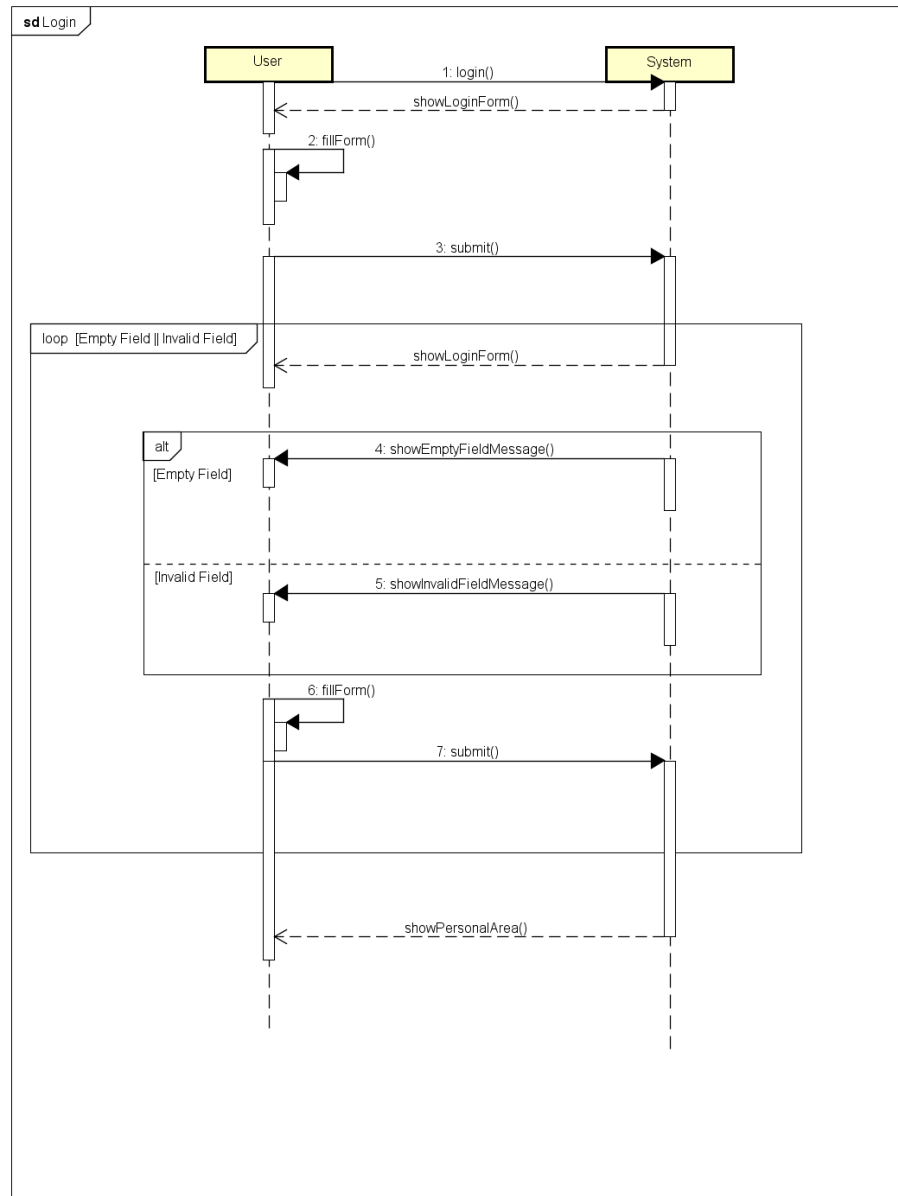


Figure 4: User Registration Sequence Diagram (Use Case 3.3.2)

**Figure 5:** User Login Sequence Diagram (Use Case 3.3.3)

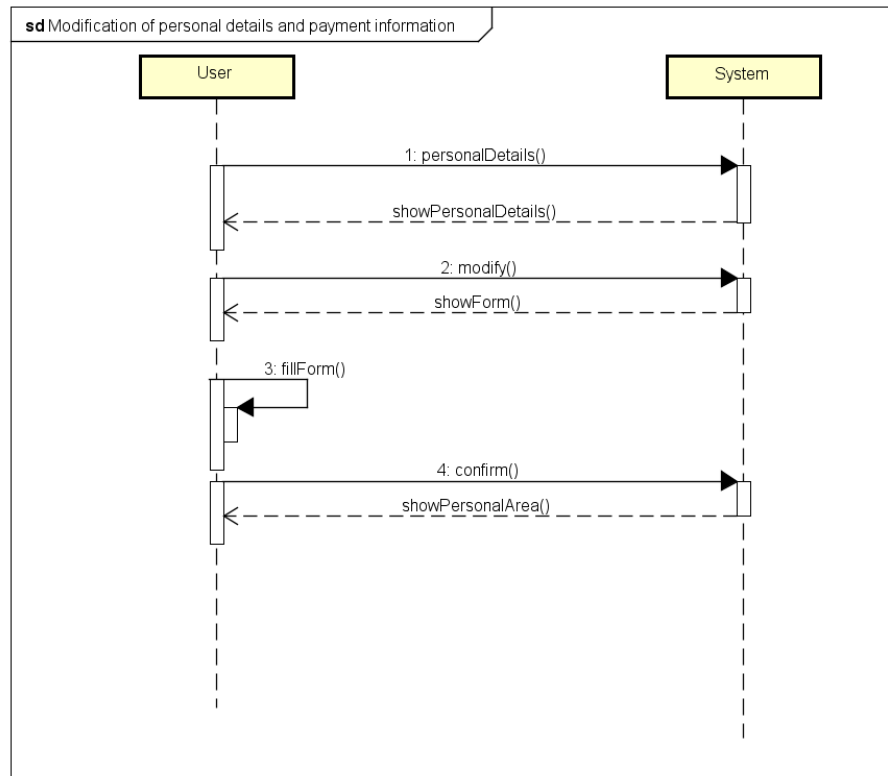


Figure 6: Personal information update Sequence Diagram (Use Case 3.3.4)

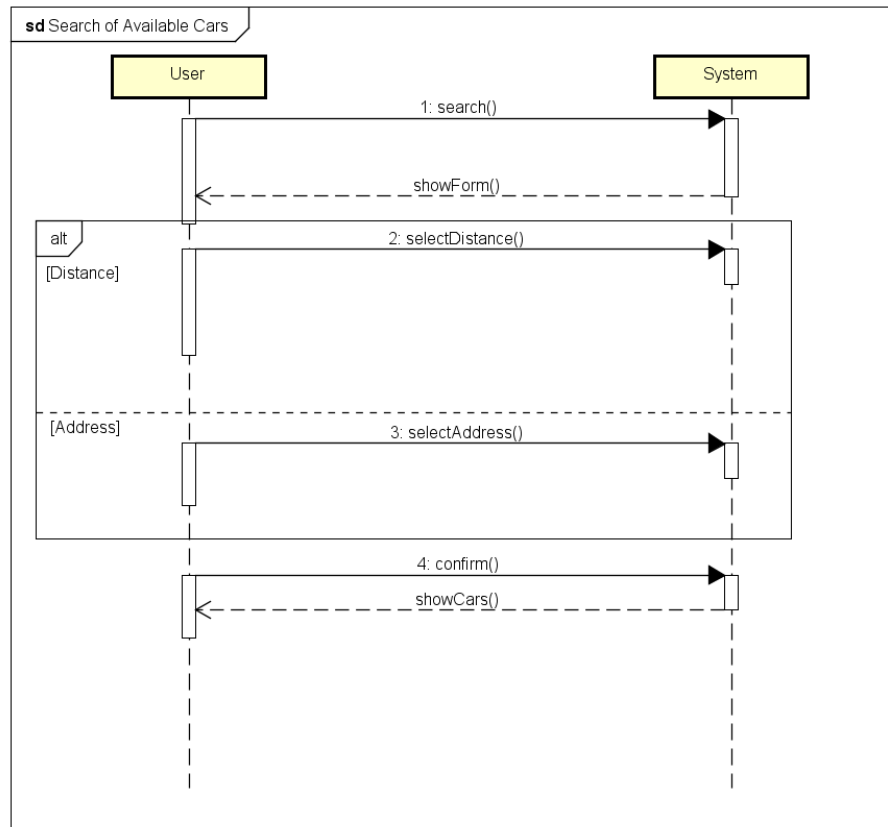


Figure 7: Common trip case Sequence Diagram (Use Cases 3.3.6,3.3.7)

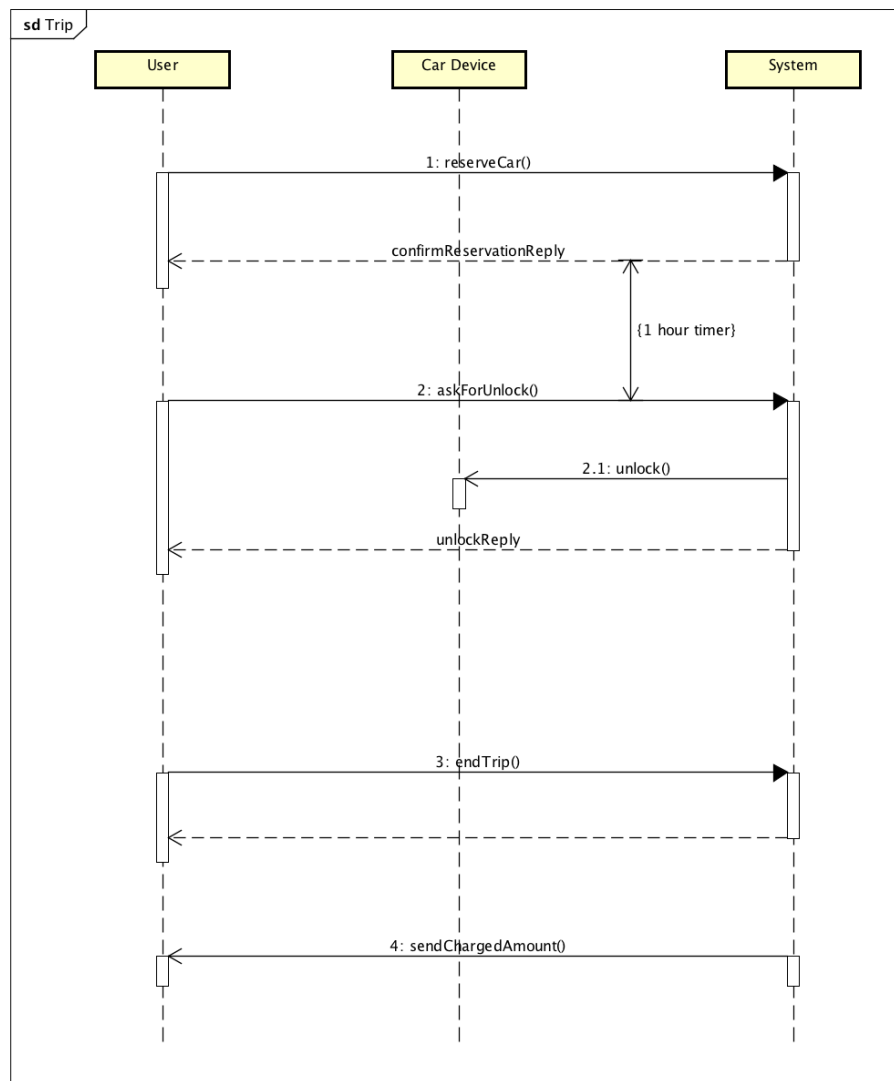


Figure 8: Common trip case Sequence Diagram (Use Cases 3.3.6,3.3.7)

5 Scenarios

5.1 Scenario 1

Andrew is a student. Because in his home city there aren't the courses he wants to attend, he moves to another city. There he rents a flat with other students. During the week he doesn't need any mean of transport and travels only on foot, because his house is very near to the university. In the weekend Andrew wants to have some fun with his friends, but unfortunately the areas with the coolest bars are far from his house and at night there aren't public transports going to those area. Furthermore neither he or his friends have a car, because they are all out of their home towns. So Andrew uses his Power EnJoy application, searches a car near his address, reserves a car, reaches it and drives till his favorite fun area, carrying with him some of his friends.

5.2 Scenario 2

It's Friday night and Bob have to go out with his girlfriend. Unfortunately his parents are away and have the car with them, so Bob searches a car near his home using Power EnJoy application. He sees on the map that there is only one car near his house and that its autonomy allows him only to reach his girlfriend's house. He looks for a special parking area near his girlfriend's house and finds one. So he decides to go to that special parking area, leave there the car, plug it into the power grid and request again for that car. Then he waits there for his girlfriends, having enough charge when she arrives to go downtown.

5.3 Scenario 3

Jim and Kate are two young tourists. They walked a lot all the day and so now they want to find an easy way to go to their hotel. Jim his registered to the Power EnJoy car-sharing service, so his reserves and picks up a car. They decide to park the car near the hotel, but when they're going to do that, they see a message telling that they are more than 3 km far from the nearest power station. Because Jim knows would have to pay more because of that and he hasn't a lot of money left, he decides to park nearer to a power station.

5.4 Scenario 4

Lucas need to go to do some shopping downtown on Sunday morning but on that date the municipal administration set a cars ban. He walk using Power EnJoy application on his mobile phone to reach the nearest Special parking areas where he has reserved an electric car that allow him to go downtown even with the ban.

6 Activity Diagram

6.1 PowerEnJoy Trip

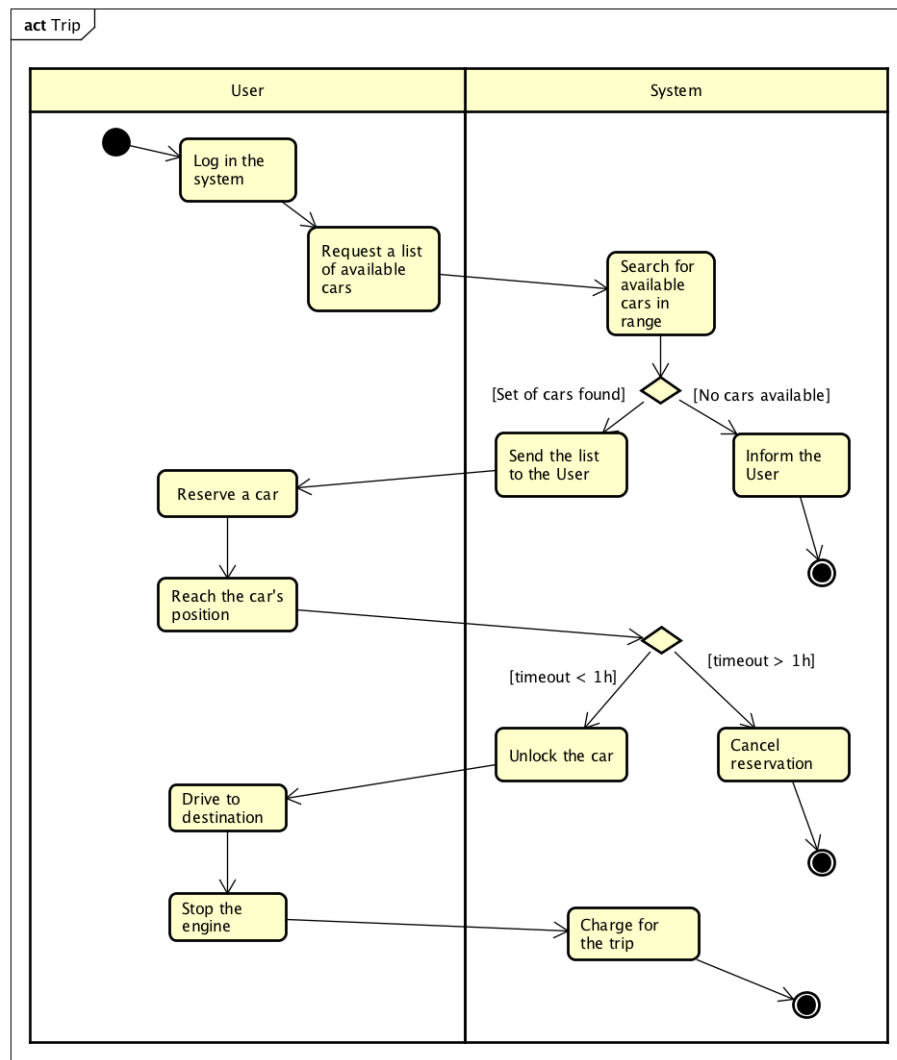
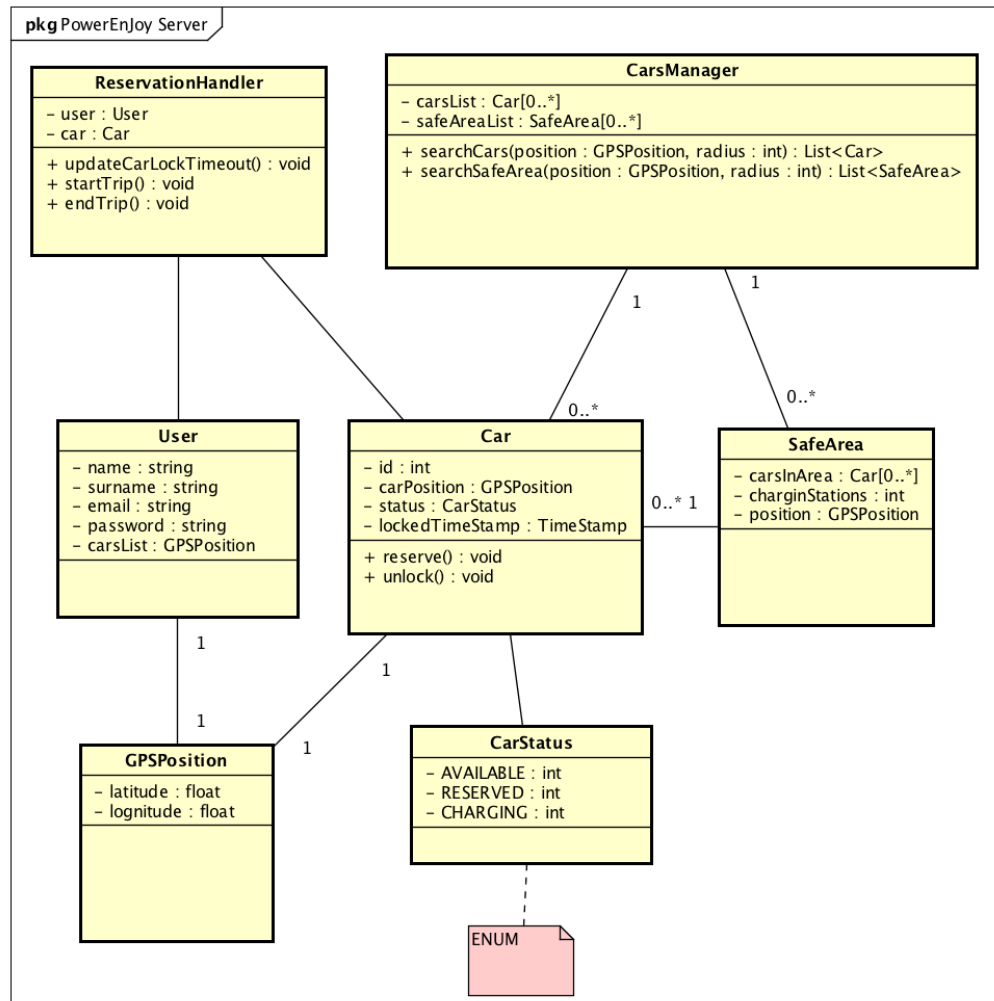


Figure 9: PowerEnJoy normal flow

7 Class Diagram

7.0.1 PowerEnJoy server



powered by Astah

Figure 10: PowerEnJoy Server Class Diagram

8 Alloy

```
// ---- SIGNATURES ----
open util/integer

sig Strings{}

enum CarStatus{AVAILABLE, RESERVED, CHARGING}

sig User{
    username: Strings,
}

sig GPSPosition{}

sig Car{
    plaqueNumber: one Int,
    seats: one Int,
    position: GPSPosition,
    status: CarStatus
}{
    seats>0
}

sig Request{
    // id: Strings,
    user: User,
    car: Car
}

sig SafeArea{
    cars: set Car,
    chargingStations: Int,
    positions: some GPSPosition
}{ chargingStations >=0}

sig Travel{
    request: Request,
    passengers: Int,
    pickPosition: GPSPosition,
    leftPosition: GPSPosition
}
{passengers >=0}

//FACTS
fact NoDuplicatedUsernames{
```

```
        no u1,u2:User | (u1!=u2) && (u1.username =
            u2.username)
    }

    fact NoTravelWithTooManyPassengers{
        no t:Travel | (t.passengers >
            t.request.car.seats)
    }

    fact NoCarsWithSamePlaqueNumber{
        no c1,c2: Car | (c1!=c2) && (c1.plaqueNumber
            = c2.plaqueNumber)
    }

    fact RequestedCarsReserved{
        no c:Car | c.status=RESERVED && no r:Request
            | r.car=c
    }

    fact DrivenCarsReserved{
        all t:Travel | (t.request.car.status =
            RESERVED)
    }

    fact RequestedCarsInSafeAreas{
        all r:Request | one s:SafeArea|
            r.car.position in s.positions
    }

    fact PickedCarInSafeArea{
        all t:Travel | one s:SafeArea| t.pickPosition
            in s.positions
    }

    fact LeftCarsInSafeArea{
        all t:Travel | one s:SafeArea| t.leftPosition
            in s.positions
    }

    fact TravelCorrespondsToRequest{
        all r:Request | one t:Travel | t.request=r
    }

    fact UserIsTheOnlyDriver{
        no t:Travel, r:Request | (t.request = r) &&
            (t.request.user != r.user)
```

```
}

fact DifferentRequestsDifferentCars{
    no r1,r2:Request | (r1!=r2) && (r1.car =
        r2.car)
}

fact DifferentRequestsDifferentUsers{
    no r1,r2:Request | (r1!=r2) && (r1.user =
        r2.user)
}

fact DifferentTravelsDifferentUsers{
    no t1,t2:Request | (t1!=t2) && (t1.user =
        t2.user)
}

fact DifferentGPSPositionInDifferentSafeAreas{
    no s1,s2:SafeArea, gps:GPSPosition | s1 != s2
        and gps in s1.positions and gps in
        s2.positions
}

fact CarInDifferentSafeArea{
all s1,s2:SafeArea | no c:Car | s1!=s2 and c in
    s1.cars and c in s2.cars
}

fact ChargingOnlyInSafeArea{
all c:Car | c.status=CHARGING => (one s:SafeArea | c
    in s.cars and c.position in s.positions)
}

// ASSERTIONS

assert ReservedCarsNotAvailable2{
all r:Request | (r.car.status != AVAILABLE)
}
check ReservedCarsNotAvailable2

assert DrivenCarsNotAvailableAndNotCharging{
all t:Travel | (t.request.car.status != AVAILABLE) &&
    (t.request.car.status != CHARGING)
}
check DrivenCarsNotAvailableAndNotCharging
```

```
assert ReservedCarMustHaveAReservation {  
  all c:Car | c.status=RESERVED => one r:Request |  
    r.car = c  
}  
check ReservedCarMustHaveAReservation  
  
//SHOW  
  
pred show{  
  #Request = 2  
  #User = 4  
}  
  
run show for 5
```



9 Appendix

9.1 Used software

1. **TeXstudio:** <http://www.texstudio.org/> to redact this document in L^AT_EX format.
2. **Astah:** <http://astah.net> to draw all the UML diagrams presented in this document.
3. **Balsamiq Mockups 3:** <https://balsamiq.com> to draw both mobile and web mockups of the service.
4. **Alloy Analyzer:** <http://alloy.mit.edu/alloy/> to test the world model in compliance with the requirements.

9.2 Time effort

List of estimated hours spent day by day to redact this document by each components of the group:

23/10/16	Matteo Michele Piazzolla 3h Andrea Millimaggi 2h
26/10/16	Matteo Michele Piazzolla 3h Andrea Millimaggi 3h
28/10/16	Matteo Michele Piazzolla 1h Andrea Millimaggi 1h
02/11/16	Matteo Michele Piazzolla 2h Andrea Millimaggi 1h
04/11/16	Matteo Michele Piazzolla 2h Andrea Millimaggi 2h
05/11/16	Matteo Michele Piazzolla 1h Andrea Millimaggi 1h
08/11/16	Matteo Michele Piazzolla 1h Andrea Millimaggi 1h
10/11/16	Matteo Michele Piazzolla 2h Andrea Millimaggi 3h
11/11/16	Matteo Michele Piazzolla 3h Andrea Millimaggi 4h
12/11/16	Matteo Michele Piazzolla 3h Andrea Millimaggi 3h