Politecnico di Milano

Software engineering 2

PowerEnjoy
RASD v1.1

Alessandro Caprarelli	874206
Roberta Iero	873513
Giorgio De Luca	875598

December 29, 2016

Contents

Chapter 1

Introduction

1.1 Purpose

This document represents the Requirement Analysis and Specification Document *RASD*. Its aim is to provide a first overview of the system that we want to develop and its main functionalities, in terms of functional requirements, nonfunctional requirements and constraints. We will integrate these specifications with several diagrams in order to highlight the constraints of our system and its boundaries.

This document is addressed to the developers and programmers, who have to implement it, and to the customers to let them have a first view of the system and the possibility to give us feedbacks according to their needs and opinions.

1.2 Actual system

The company is starting its business right now, so we assume that there is no system providing the requested service. All the functionalities have to be developed from the beginning. To develop some functionalities of PowerEnjoy we will rely on external services.

1.3 Scope

The aim of the project PowerEnjoy is to provide an automated service of car sharing. After a registration, a client can hire a car near him/her through the web or mobile

application and he/she can enjoy of all the extra services offered. The exact position of the client is determined by the GPS signal of the client's device or it's allowed to manually insert a specific address. So the system displays all the available cars in the client's close area. Then the client could make a reservation of a car, after which the system notifies the client with a message of confirmation with the car identifier. If the reservation procedure successfully ends the chosen car won't be available anymore for other clients. Moreover a client cannot hire more than one car at the same time. After the reservation the client has at most one hour to reach the car, when this time expires the system gives a penalty to the client and the car, previously hired, is available again for other clients. The system allows the client to cancel his/her reservation. When the client reaches the car, he/she can tell the system that is nearby through a specific button in the application and he/she starts to pay as soon as the engine ignites. During the travel the system supervises the current charge of the car and notifies it to the client through a screen located in the car. The system stops charging the amount of money that the client has to pay when he/she communicates through the application his/her decision to stop the rent. When the car is parked in a safe area and the client exits, the system locks the car automatically and starts the procedure of payment. The client is notified with the result of this procedure through an SMS, including the final fare.

1.4 Actors

- **Guest:** a person that is not already registered in the system or that has to log in. He/she can only display the home page, with the description of the provided services, and the registration page.
- **Registered Client:** a person who has valid access credentials to log in the system (username and password). Once logged in, he/she can request and hire an available car near him/her, cancel a reservation and he/she have access to all the customer area of the application.
- **System Administrators:** a certified user who, after login, has the responsibility to manage administration processes. He is also in charge of updating data about business logic.

1.5 Goals

G1) The Registered Client can hire a car through web/mobile application.

G2) The Registered Client can hire a car through an SMS.

G3) The Guest Client can register himself/herself into the system as Registered Client.

G4) Registered Clients can login into the system.

G5) The Logged Client can manage his/her sensible data.

G6) The Registered Client can manage his/her requests of hiring.

G7) Ensure the Registered Client the possibility to receive a discount on his/her last ride.

G8) Ensure a uniform distribution of cars in the city.

G9) Guarantee to have always a minimum number of cars in the system with enough charge to be hired.

G10) Allow the system administrator to update and check data in the database of the system.

G11) Guarantee a correct interoperability of the system with external services.

G12) Ensure that a Registered Client's bad behaviour is punished with the application of some penalties.

1.6 Definition, acronyms, abbrevations

1.6.1 Definition

- **Guest client:** a person that is not already registered in the system or that has to log in.
- **Registered client:** a person who has valid access credentials to log in the system.
- **System administrator:** privileged user, in charge of managing administration processes and of updating business logic.
- **Reservation:** it is the action performed by a registered client that allow him/her to reserve an available car for maximum one hour.
- **Journey time = travel time:** time elapsed since the user starts the engine to the user parks the car and terminates the journey.

- **Available car:** a car that is not reserved by any user and has enough charge to be rented.
- **Unavailable car:** a car that is already reserved or damaged, so impossible to reserve.
- **Gps navigation:** it is the navigation system that is included in the car on board system. It could be used by the user to find direction to the final destination.
- **Final destination:** address where the user wants to go.
- **Safe area:** the region where is permitted to park and leave a car once the rent is terminated.
- **Power grid station:** the area where it's allowed users to park the cars, leaving them attached to the power grid.

1.6.2 Acronyms

- **RASD:** Requirements Analysis and Specification Document
- **API:** Application Programming Interface
- **UI:** User Interface

1.6.3 Abbrevetations

- **[Gn]:** n-goal
- **[Rn]:** n-functional requirement
- **[An]:** n-assumption

1.7 Identify stakeholders

Our main stakeholder is the owner of PowerEnjoy that wants a state-of-the-art application in order to let users to easily locate, reserve, use and pay a fleet of shared electric cars spread across the city.

1.8 Reference documents

- PowerEnjoy specification document (assignment).
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

1.9 Document overview

The document is substantially divided in 4 sections:

- Introduction: it gives an high-level description of the software explaining the main purposes, goals and context;
- Overall description: it provides an overview of the main characteristics of the system to develop, pointing out in particular constraints and assumptions.
- Specific Requirements: this part contains all the system requirements, typical scenarios and different kind of diagrams useful to understand easily the functionalities.
- Alloy: the last part of the document contains the description of our model in Alloy, a possible result obtained using this software and the consequent generated world.

Chapter 2

Overall description

2.1 Prodcut perspective

The PowerEnjoy application will consist of a mobile and a web application for clients, that offer the same functionalities, and an internal interface for the system administrator.

The system also relies on existing external services and their interactions will be described later in this document.

2.2 User characteristic

There are two main classes of users in our system: Clients and System Administrators.

The System Administrators are in charge of managing administration processes and of updating business logic. These users can analyze data in order to produce reports and statistics about the quality of service.

A Client is a person that wants to use this application is a person that wants to move across the city using a car without the responsibility of owning it.

2.3 Constraints

2.3.1 Regulatory policies

The system has to manage sensible data such as email addresses and phone numbers to communicate with users, GPS tracking to locate the clients and credit cards information for payments. Registered Clients must agree with the privacy legislation in force in the country in which the software is used.

2.3.2 Hardware policies

For the mobile application the system requests that the client's device has:

- active internet connection;
- enough space in the device memory;
- active GPS, if the client doesn't want to put manually the address of her/his position.

For the Web Application the system requires that the browser support at least websockets component. As a reference we use this tool: http://caniuse.com/#feat=websockets.

2.3.3 Interfaces to other applications

- Google maps: to provide to the user information about the position of the cars and how to reach them.
- Driving licences database: to check, during the registration step, whether a driving licence is valid or not.
- Payment interface service: to manage financial transactions related to PowerEnJoy service.
- SMS gateway provider: in order to send notifications to clients.
- GPS system: to locate clients.

2.3.4 Parallel operations

The server can handle parallel operations and requests related to different clients.

2.4 Assumptions and dependencies

2.4.1 Assumptions

A1) The Clients can leave the rented cars only in one of the safe areas.[see Safe Area in Definitions]

A2) There isn't an old system providing the same services.

A3) Only registered clients can use the services of PowerEnjoy.

A4) There's no car having the same plate.

A5) Each car of PowerEnjoy is conformed to have at most 5 passengers, driver included.

A6) The client that does a reservation can take with him/her in the car at most 4 passengers.

A7) The car sharing service is operating 24 hours per day, 7 days a week, 365 days per year.

A8) The position of each car is known using the GPS signal. Each car is provided of an autonomous GPS system reachable from our service.

A9) In the cars there are sensors that provides the system information about the current number of passengers.

A10) The cars can only be parked in a safe area.

A11) A client can maintain his/her reservation of the car during a stop, parking the car temporary in a not safe area. This is allowed only if the client proclaim the intention to do that through the application. During this stop the client continues to pay with some reduction of the standard hiring cost.

A12) The external services are not affected by downtimes or unavailability.

A13) Users own skills to interact correctly with the system.

A14) Users own at least one device to use the functionalities of the system.

A15) External services provide to our system consistent and valid information.

2.4.2 Dependencies

2.5 Future possible implementations

- Extend the system in order to manage the rent of new kind of vehicle.
- Extend the system to let users filter cars by selecting their desired characteristics(such as number of possible passengers, dimension of the car, minimum desired charge).

Chapter 3

Specific requirements

3.1 External Interface Requirements

The system to be developed has to store information about cars, accounts, all data related to accounting and administration sectors and also data produced by system itself to manage the rental service. The external services called by the system directly interact with it. External Interface Required for PowerEnjoy:

- A database;
- A SMS gateway.
- A Push Service;

3.1.1 User interfaces

Home mobile app

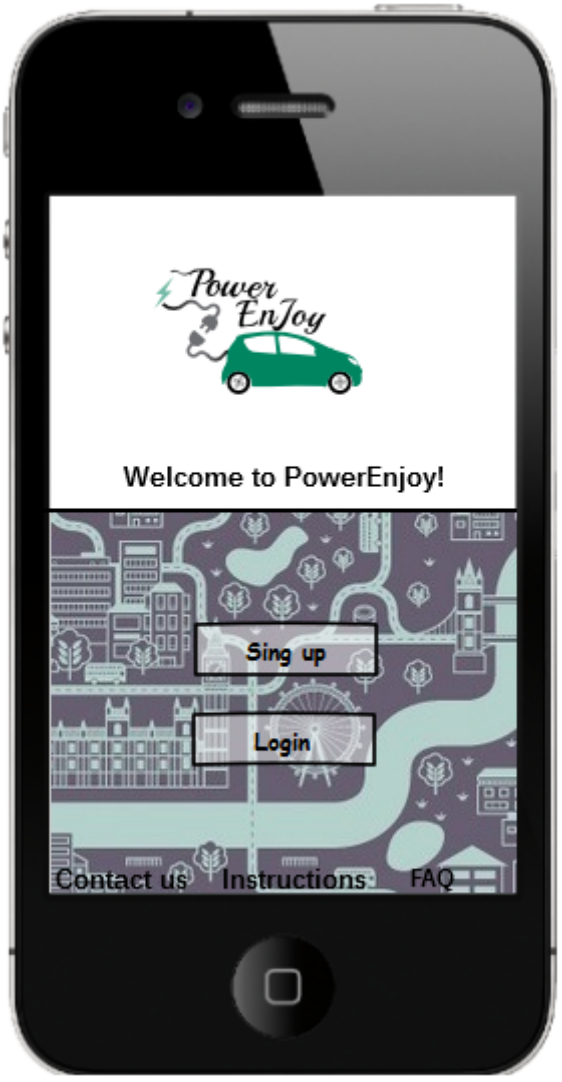

Figure 3.1: Homepage of mobile app

Home web app

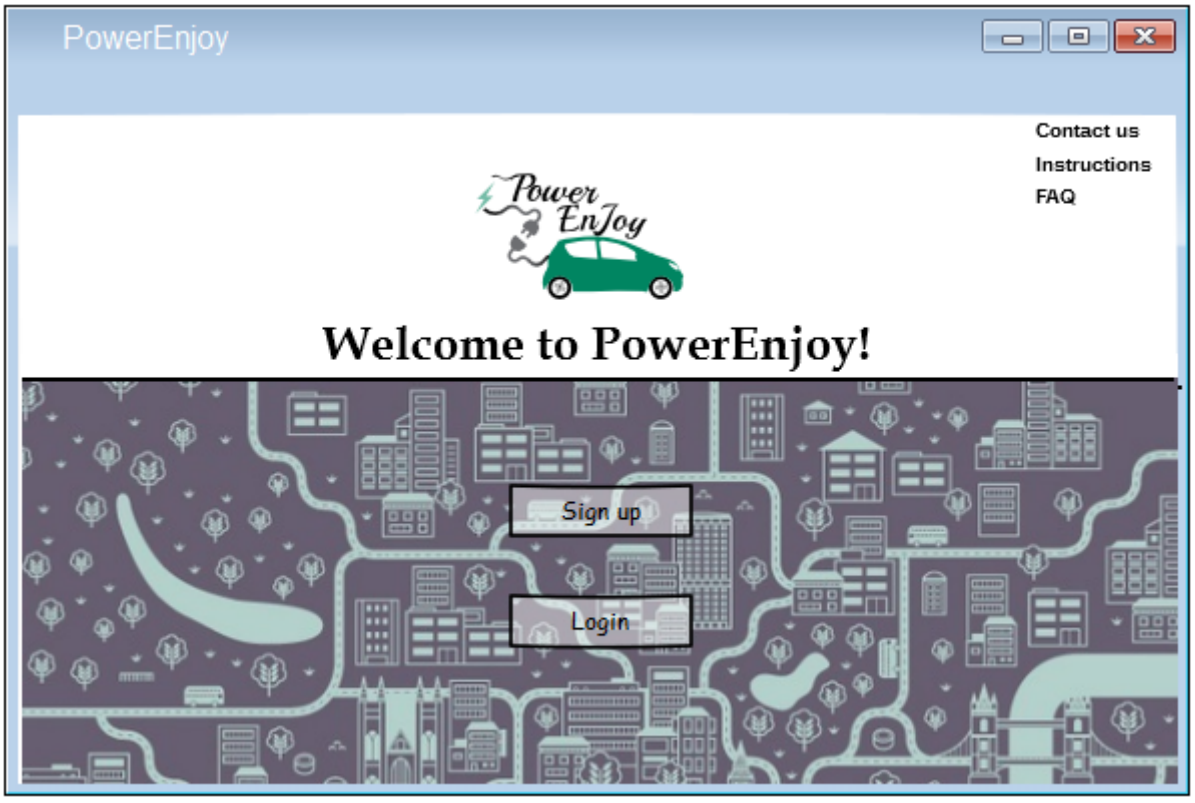

Figure 3.2: Homepage of web app

Reservation mobile app

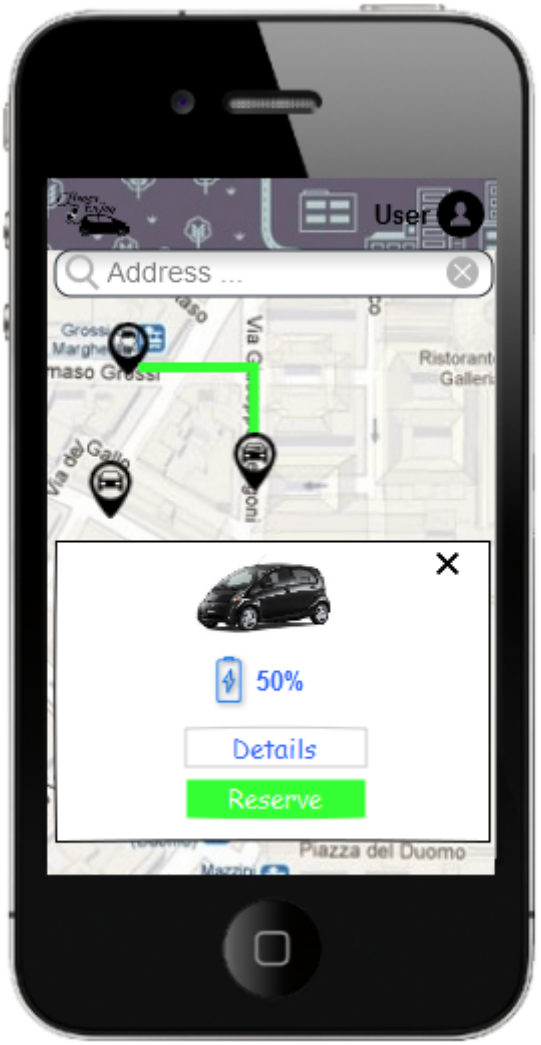

Figure 3.3: Reservation page of mobile app

Reservation web app

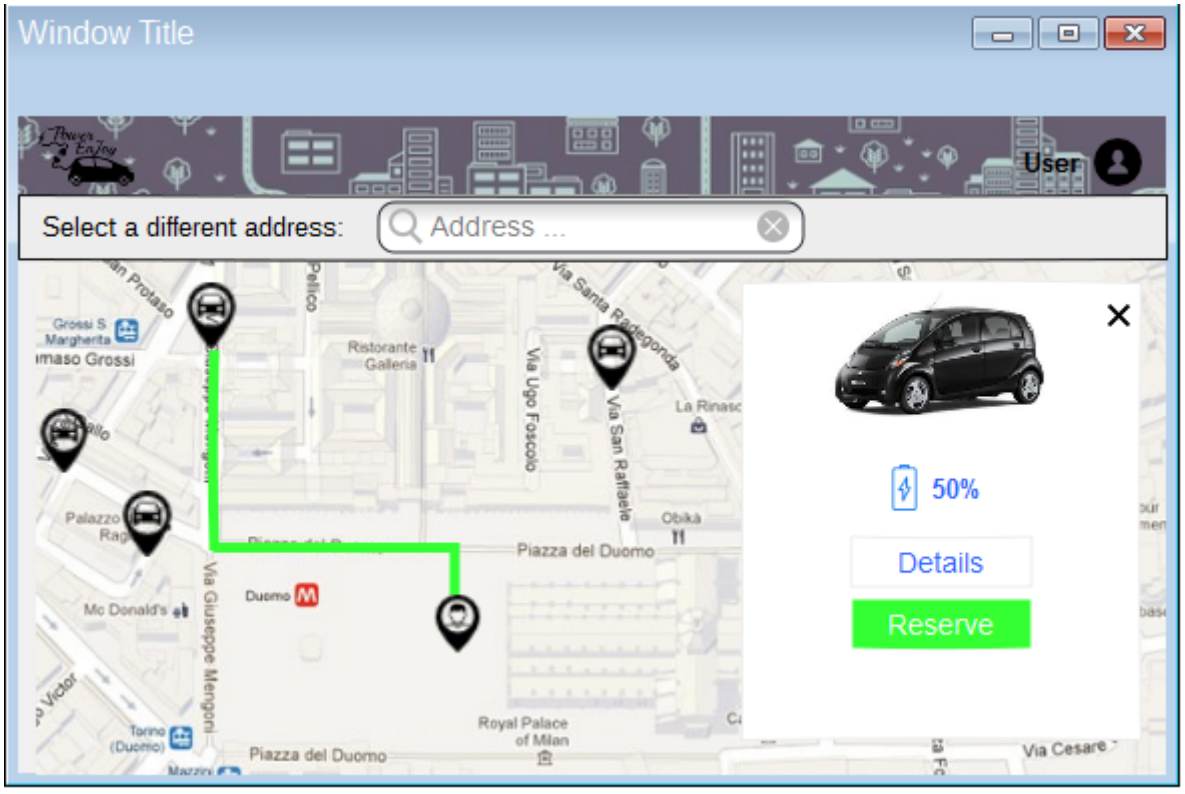

Figure 3.4: Reservation page of web app

Account details mobile app

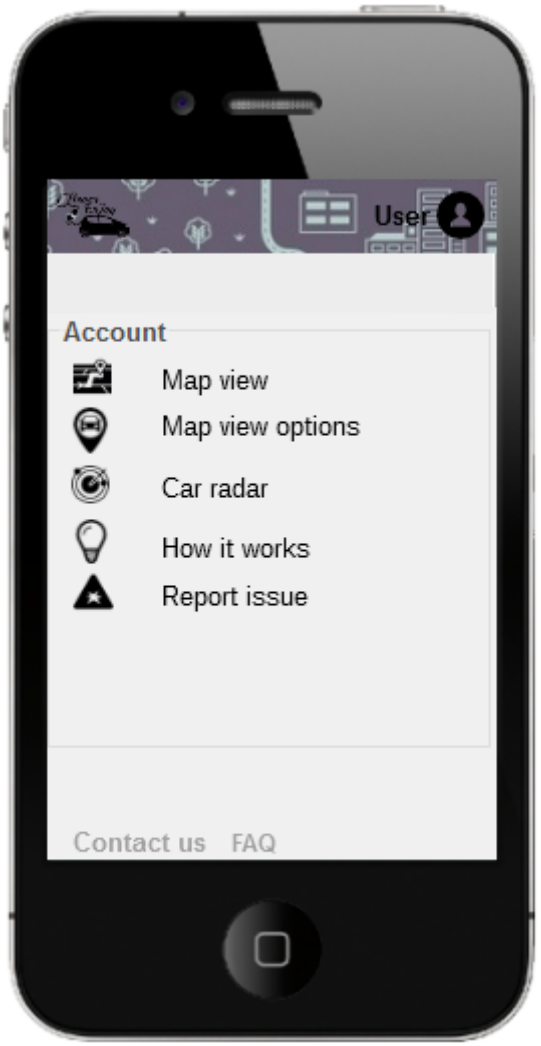

Figure 3.5: Options collapsable panel

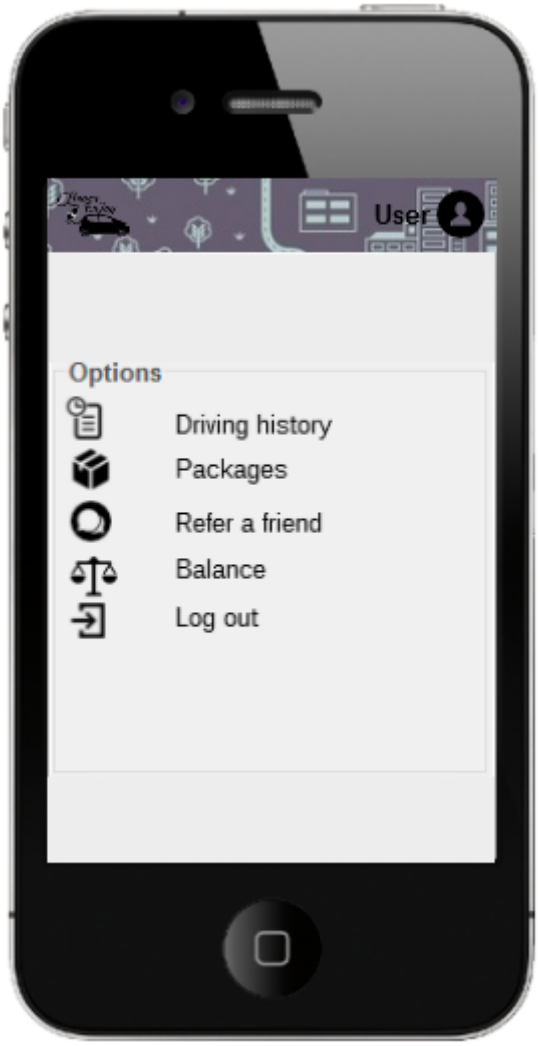

Figure 3.6: Account collapsable panel

Account details web app

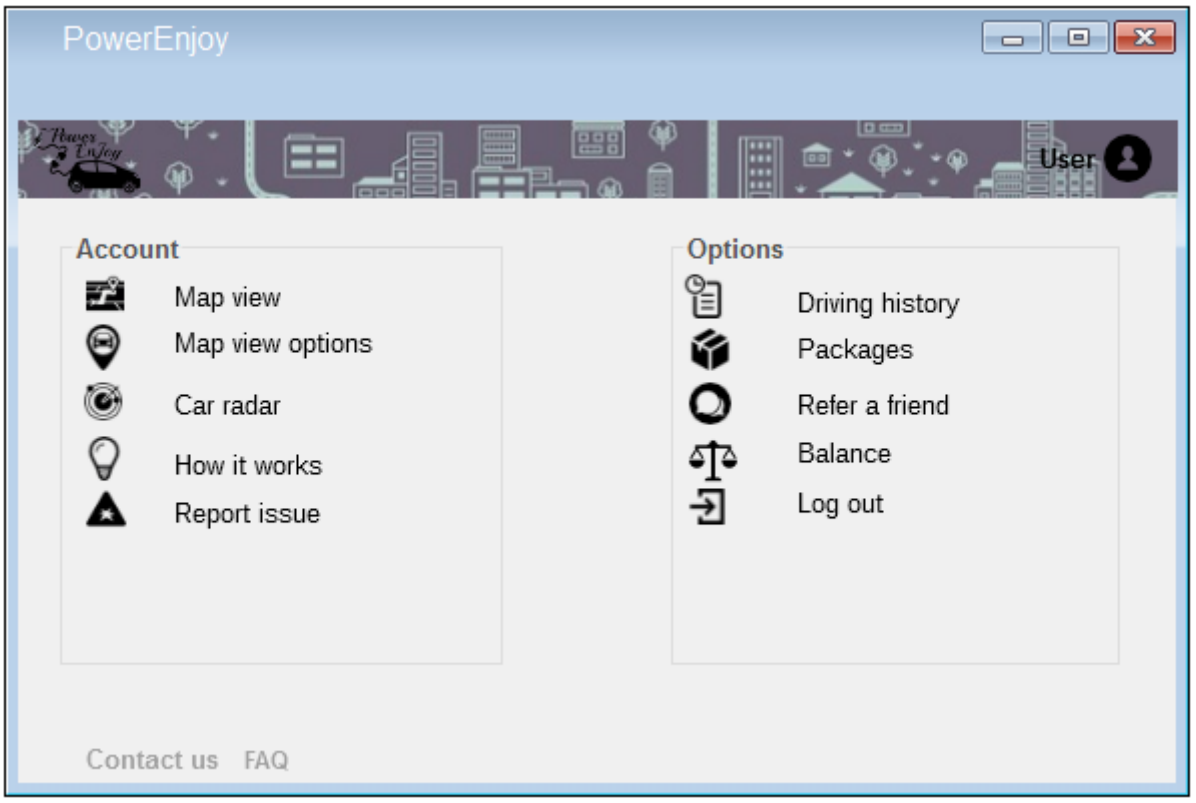

Figure 3.7: Account details and options page of web app

3.1.2 Hardware interfaces

The hardware interfaces of our system are:

- Owned by the clients: a computer or a mobile device in order to complete all the actions concerning registration, login, reservation procedure and also to manage his/her account data.
- Owned by PowerEnJoy: a GPS navigation device for each car, that represents the main interface between the client and the system during the ride period. Its main functionalities are:
  1. Require the client his/her personal pin code;

2. Let the client choose all the options related to his/her ride;
3. Show the driver all the notifications sent by the system;
4. Let the client choose to end his/her rent.

3.1.3 Software interfaces

- DBMS: MySQL
- Proxy server: Nginx
- Application server: Express.js
- Web app UI library: Riot.js
- Websocket library: Socket.io

3.1.4 Comunication interfaces

The system communicates using websocket, that makes possible to have an effective real-time application.

3.2 Functional requirements

3.2.1 [G1] - The Registered Client can hire a car through web/mobile application

R1.1) The system allows the Registered Client to choose the car to hire only between the available ones.

R1.2) The system allows only one reservation per time for each Registered Client.

R1.3) The system allows the Registered Client to cancel his/her request.

R1.4) The system considers the reservation request as valid for at most one hour from the moment in which the request of hiring is accepted. If the Registered Clients doesn't reach the car before this time expires, the system gives him/her a penalty of one euro and mark the car as 'available' again.

R1.5) When a car is reserved the system marks it as 'unavailable'.

R1.6) The system unlock the chosen car only when the Registered Client is nearby and he/she communicates this to the system through the application.

R1.7) The system locks the car when the Registered Client parks it in a safe area or in a power grid station and he/she communicates the system the intention of ending the rent.

R1.8) When the rent is finished the system marks the car as "available" again.

R1.9) The system starts charging the Registered Client for a given amount of money per minute as soon as the engine ignites.

R1.10) The system stops charging the user as soon as the car is parked in a safe area or in a power grid station and the Registered Client exits the car. Then the system starts the payment procedure.

R1.11) The system allows the Registered Client to start driving if and only if he/she insert his/her personal code.

R1.12) The system allows the Registered Client to choose an address in which he/she wants to hire a car or if he/she wants to be located through the GPS signal.

R1.13) The system must supervise the charge of the car during all the rental period, showing the current charge on the GPS navigation device. It also has to show a warning notice if the charge is equal to the minimum possible charge admitted.

R1.14) The system allows the Registered Client to find the location of an available car within a certain distance from his/her current location.

3.2.2 [G2] - The Registered Client can hire a car through an SMS.

R2.1) The system must check that the SMS is sent from a phone number related to a Registered Client account.

R2.2) The system accepts only well formed SMS messages, that follow a precise pattern: "Action + plate".

R2.3) The system must check a correspondence of the plate indicated by the Registered Client with one of the cars in its database labeled as available.

R2.4) The system sends an SMS to communicate the Registered Client the outcome of his/her request.

R2.5) If the request has been accepted, the system marks the car as 'unavailable' and reserves it.

R2.6) The system unlocks the car if and only if it receives an 'unlock' SMS from the same phone number that has reserved it.

R2.7) The system deletes the request of reservation if and only if it receives a 'delete' SMS from the same phone number that has done the reservation. The system and mark the car as available again.

R2.8) The system considers the reservation request as valid for at most one hour from the moment in which the request of hiring is accepted. If the Registered Client doesn't send an 'unlock' SMS before this time expires, the system gives him/her a penalty of one euro and mark the car as 'available' again.

R2.9) When the rent finishes the system sends the Registered Client an SMS with a recap of the overall cost of the rent.

R2.10) The system locks the car when the Registered Client parks it in a safe area or in a power grid station and he/she communicates the system the intention of ending the rent.

R2.11) The system marks the car 'available' again once it is locked.

R2.12) The system allows the Registered Client to start driving if and only if he/she insert his/her personal code.

R2.13) The system starts charging the Registered Client for a given amount of money per minute as soon as the engine ignites.

R2.14) The system stops charging the user as soon as the car is parked in a safe area or in a power grid station and the Registered Client exits the car. Then the system start the payment procedure.

R2.15) The system must supervise the charge of the car during all the rental period, showing the current charge on the GPS navigation device. It also has to show a warning notice if the charge is equal to the minimum possible charge admitted.

3.2.3 [G3] - The Guest Client can register himself/herself into the system as Registered Client.

R3.1) The system requests an e-mail and a telephone number during the registration.

R3.2) The system rejects an e-mail and a telephone number already used by another Registered Client.

R3.3) Once the registration procedure is successfully completed, the system sends a confirmation email to the new Registered Client.

R3.4) The system checks if e-mails or telephone numbers are well formed.

R3.5) The system considers a user as a Registered Client if and only if he/she opens the activation link in the confirmation e-mail.

R3.6) The system considers a phone number valid if and only if the user inserts the code sent by the system through an SMS.

R3.7) The registration procedure requires the Guest Client to indicate his/her driving licence.

R3.8) The system requires the Registered Client to specify one valid payment method.

R3.9) The system must send back to the Registered Client's email a password that he/she can use to access the system.

3.2.4 [G4] - Registered Clients can login into the system.

R4.1) The system allows only Registered Clients, with valid access credentials to login in the application.

R4.2) The system allows Registered Clients that have forgotten their password to recover it, sending an email with a temporary password to the e-mail address indicated during the registration procedure.

R4.3) The system requires Registered Clients to enter valid username (email address) and password to login.

3.2.5 [G5] - The Logged Client can manage his/her sensible data.

R5.1) The system allows the Logged Client to change his/her payment method.

R5.2) The system allows the Logged Client to change his/her current password.

R5.3) The system allows Logged Client to change his/her own email address. The system considers valid the new email address if and only if the Logged Client open the activation link in the confirmation e-mail sent by the system at the end of the procedure.

R5.4) The system allows Logged Client to change his/her phone number. The activation of the new phone number is completed only when the Logged Client insert the code sent by the system through an SMS to the new phone number.

R5.5) The Registered Client can manage his/her requests of hiring.

R5.6) The system allows Registered Client to cancel his/her request of hiring a car.

R5.7) The system allows Registered Client, once logged into the system, to display his/her history of hiring request.

3.2.6 [G6] - Ensure the Registered Client the possibility to receive a discount on his/her last ride.

R6.1) During the entire rent period the system must check if some conditions happen in order to apply a discount on the final rent cost are satisfied.

R6.2) The system allows the Registered Client to enable the 'save money' option at the beginning of his/her ride through the GPS navigation device.

R6.3) If the system detects the Registered Client took at least two other passengers onto the car, the system applies a discount of 10% on the last ride.

R6.4) If a car is left with more than 50% of charge, the system applies a discount of 20

R6.5) If a car is left in a power grid station and the Registered Client takes care of plugging the car into the power grid, the system applies a discount of 30% on the last ride.

3.2.7 [G7] - Ensure a uniform distribution of cars in the city.

R7.1) The system, considering the actual distribution of the cars in the area, must always highlight through the GPS navigation device the area in which the client may park the car

3.2.8 [G8] - Guarantee to have always a minimum number of cars in the system with enough charge to be hired.

R8.1) The system must report a warning if it detects that the number of cars in the system with enough charge to be hires is less than the minimum possible.

3.2.9 [G9] - Allow the system administrator to update and check data in the database of the system.

R9.1) The system must disallow not certified system administrators to perform such operations.

R9.2) The system must guarantee integrity and consistency of data after eventual updates performed on them.

R9.3) The system allows the system administrators to add new Safe Area in the set of Safe Area predefined by the management system.

3.2.10 [G10] - Guarantee a correct interoperability of the system with external services.

R10.1) The system must always be able to perform requests to external services, when they're needed.

3.2.11 [G11] - Ensure that a Registered Client's bad behaviour is punished with the application of some penalties.

R11.1) If a car is left at more than 3 KM from the nearest power grid station, the system charges 30% more on the last ride.

R11.2) If a car is left with more than 80% of the battery empty, the system charges 30% more on the last ride.

3.3 Scenario

3.3.1 Scenario 1

Frank and his flatmates have nothing to eat for lunch and they need to go to the supermarket, that is not close to Frank's house. The thought to do the same long road two times is not an awesome picture in guys' minds. Fortunately yesterday Frank saw an advertisement about the car sharing service of PowerEnJoy, so he downloads the mobile application and registers himself to use that service. Frank activates the

GPS signal tracking to let the app locates his position and the app shows him a car near his house. So they make a reservation and exit from home to reach the car. Once arrived, Frank open again the app to communicate to PowerEnJoy he's nearby: the system unlocks the doors and Frank and his friends are able to jump in. The car starts the engine only after the insertion of the personal account PIN, provided during the registration: Frank enters it and the car becomes able to be turned on. After reaching the supermarket, Frank and his friends don't want to leave the reservation of the car, because they know there is the possibility to find no cars nearby later: so they decide to leave the rent in "Pit Stop" mode. After the shopping they come back home, reactivating the "Go" mode. While driving Frank sees in the screen of the GPS navigation device the evaluation of the amount of money he is going to pay. So he decides to choose the option save money to obtain a discount: he inserts the address of his house and the system indicates him where to leave the car in order to have a discount. So he reaches the station and leaves the car in a specific space, where is also possible to plug the car to the power grid. He terminates his rent choosing the option on the GPS navigation device "To the Box". After that he has 30 seconds to plug the car to the power grid, to achieve the discount on that ride.

3.3.2 Scenario 2

Mrs. Hilly is a dynamic old lady. Her grandson talked to her about PowerEnJoy and its innovative system to share cars in an ecological way with a cheap fare. This service seems suitable for Hilly because she doesn't have an internet connection but is really skilled to use SMS, the first way to communicate with her friends. So her nephew helps her to complete the registration. Hilly today has a medical examination and the hospital is too far to reach only with her legs: so she starts walking on the road, hopeful to find a PowerEnJoy car to rent. After few meters, she gets close to a PowerEnJoy car and starts to follow the instructions given by her grandson: she enters the number of the service as addressee and as text the caption "Rent" followed by the plate of the car. After few seconds she receives an SMS that confirms her reservation. So she sends another SMS with the caption "Open" and waits until the doors of the car are unlocked by the system. Then she enters her personal code through the GPS navigation device and drives to the hospital.

3.3.3 Scenario 3

Last week Jhonny bought a train ticket to come back home. It's 14:25 and the departure time is fixed at 15:00: Jhonny is late and he thinks the best way to arrive

in time at the train station is to rent a car of PowerEnJoy. Fortunately there's one near him, so he opens the mobile application and reserves it. Once entered in the car and arrived at the station, he realizes there's no park available near it and the only way to leave it is to be double-parked. So he leaves the car and takes his train. After a short time, some traffic officers sees the car in double-park: so they jump out the car and fine a ticket. Meanwhile Lucy is arriving at destination (the same train station) and she has a heavy handbag to bring at home, so decides to hire a PowerEnJoy. She arrives at the car and immediately sees the ticket on the windscreen of the car. To avoid all possible problems Lucy, before starting her ride, inserts in the GPS navigation device the presence of a ticket and then starts her trip to home.

3.3.4 Scenario 4

Milan has to face a big problem constantly present in daily life: pollution. For this reason the administrative offices of the city have decided to announce the alternation of cars traffic, basing it on the numbers of the plates of the cars. Today there's the opportunity to travel only for the owners of cars having an even plate number. Unfortunately Shaun owns only a car with odd plate number, so he can't use his car. A lot people have found a solution taking the public transports but Shaun can't do that, because he has to go in a zone of the city unreachable by them. Thus he thinks it's a perfect situation to try PowerEnjoy service. He opens the mobile application and, as he also presumed, the nearest car is about 3 kilometers far from him, but he has an important appointment and he can't be late for that. So he starts walking to the car, following the instructions given by the app that estimates 40 minutes to reach the position of the car. It's been 30 minutes when Shaun is about 2.3 kilometers far from the car and the app shows him a warning reciting "Ehy Shaun, you have 30 minutes to reach the car. After that lapse of time your request won't be considered still valid and you will have to pay a fee of 1 EUR". Shaun doesn't care about the message because he knows he is nearby. Once arrived, he sends the request to open the doors of the car and jump in.

3.3.5 Scenario 5

Sandy wants to go to a concert not so far from her house but she doesn't want to arrive walking, because of the low security of the road that she has to pass through. The doors to enter in the concert field are going to be opened soon and she wants to hire a PowerEnJoy car: she opens her computer browser and goes on www.powerenjoy.it and inserts the address of her house and chooses a car among the

available ones. Sandy is a fan of PowerEnJoy and she wants to see the history of her previous requests. Thus open the profile page and click on History: the site shows her all the requests advanced in the past and also the last one. It's late and Sandy has to go, so starts moving to reach car. Once exited the house, it starts raining: unfortunately the concert is hold in an open space and if she go, she will get sick. So she decides to come back home and cancel the request through the History panel of the site.

3.3.6 Scenario 6

After a birthday party, Paul has to come back home but it's too late so there aren't public transportations anymore. Paul wants to hire a car with PowerEnJoy but he is worried about how much he is going to pay. So he decides to take advantage of all the possibility to obtain a discount, since he already used PowerEnJoy a lot of times. First of all he finds three friends, that lives near him, to join him in the car. So they hire a car, Paul inserts the destination address in order to choose the option "save money". As soon as the engine ignites the sensors of the system situated in the car detect that there are three more people with the driver, so it will apply another discount to the final total amount. When they arrive at the destination, Paul is happy because he has arrived home in a brief time paying a little amount of money.

3.3.7 Scenario 7

James usually goes to work walking. Since the day is rainy he decides to hire a car of PowerEnjoy. When he finally enters the car, he finds out that the car has been left in bad conditions by the previous user. So before starting his ride, James reports the system all the present issues of the cars through the GPS navigation device, in order to not be responsible of them and to help PowerEnjoy to take measures. Then James reaches his destination without problems.

3.4 UML models

3.4.1 Use cases diagram

General use case

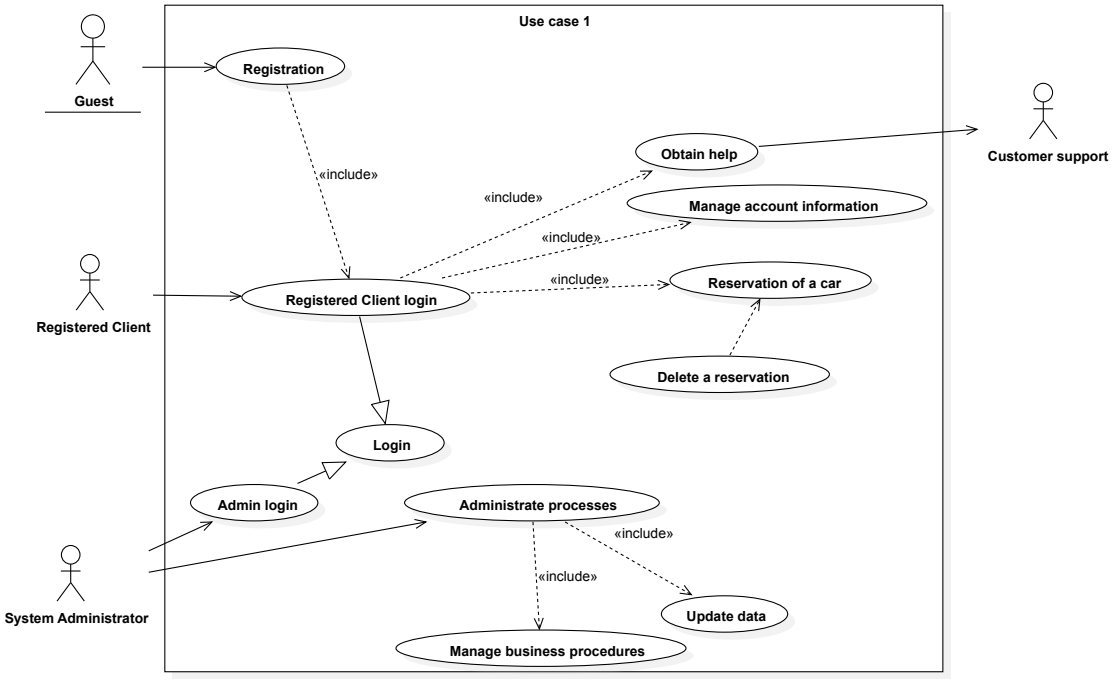

Figure 3.8: General use case

Rent use case

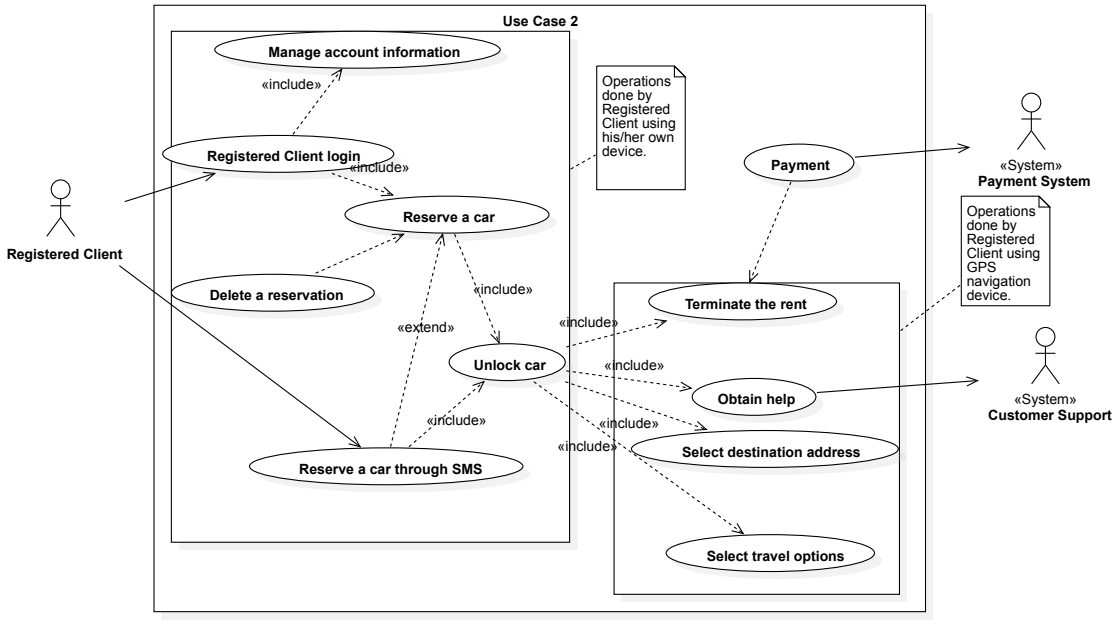

Figure 3.9: Rent use case

3.4.2 Use cases description

Registration

Name	Sign up
Actor	Guest
Entry conditions	–
Flow of events	1. Guest clicks the "Sign up " button and the registration page is shown. 2. He/she fills the form with his/her personal details and presses the "Sign Up" button. 3. The system checks the data submitted by the guest, in particular if all the mandatory fields have been filled, and if there are no problems he/she is registered into the system. In addition the system sends back to the new user an email with an activation link and a password that he/she can use to access the system. 4. The user opens the email and clicks the activation links. The system activates the user.
Exit conditions	The user receives an email to confirm the registration and a password to login
Exceptions	If one value of the form is not valid or missing(concerning the mandatory fields), the registration is not possible and the app shows an alert indicating the errors. If the user is already registered with the information provided, the system forbid him/her to create a new account.

Login

Name	Sign in
Actor	Guest, Registered client
Entry conditions	Guest must already have credentials
Flow of events	1. Guest clicks "Sign in" button and the login page is shown. 2. He/she writes his/her username and password and clicks the "Sign in" button. 3. The system checks the credentials and if they are valid the user is logged. 4. The systems shows the main page to the user.
Exit conditions	Guest is promoted to a Registered client.
Exceptions	If the credentials passed to the system are not valid the app shows an alert indicating the errors. If the Registered Client doesn't remember either his/her username or his/her password the system allows to start the procedure to recover it.

Reservation through web/mobile application

Name	Reserve a car through web/mobile application
Actor	Registered client
Entry conditions	Registered Client must be already logged into the system.
Flow of events	1. The Registered Client look for a car nearby his/her current position or near a specified address. 2. The Registered Client makes a reservation of a car, by a click on the "Reserve" button. 3. The Registered Client reaches the reserved car and clicks on the "I'm nearby" button, to let the system unlock the car doors. 4. The Registered Client enters the car and insert in the GPS navigation device his/her personal pin code.
Exit conditions	The Registered Client terminates his/her rent through the "Terminate Rent" button in the GPS navigation device and automatically pays for the rent through the payment method specified during the registration procedure.
Exceptions	If the Registered Client doesn't reach the reserved car in one hour from the reservation the system applies him/her a penalty, marks the reservation as not valid anymore and the car as available again. If the Registered Client has already an active reservation, the system shows an alert indicating the kind of error.

Reservation through SMS

Name	Reserve a car through SMS
Actor	Registered client
Entry conditions	The Registered Client is nearby the car he/she wants to rent.
Flow of events	1. The Registered Client sends an SMS with the caption "Hire" followed by the plate of the car he/she wants to rent. 2. The Registered Client receives a confirmation SMS and writes another SMS with the caption "Open". 3. The Registered Client waits until the doors are unlocked. 4. The Registered Client enters the car and insert in the GPS navigation device his/her personal pin code.
Exit conditions	The Registered Client terminates his/her rent through the "Terminate Rent" button in the GPS navigation device and automatically pays for the rent through the payment method specified during the registration procedure.
Exceptions	If the Registered Client doesn't send the "Open" SMS within one hour the system applies him/her a penalty, marks the reservation as not valid anymore and the car as available again. If the phone number used for reservation is not associated to any Registered Client in the database, the system sends a warning SMS pointing out the error. If the phone number is associated to a Registered Client but he/she has already an active reservation, the system sends a warning SMS pointing out the error.

System Administrator login

Name	System Administrator login
Actor	System Administrator
Entry conditions	The system administrator is registered in the system
Flow of events	1. The system administrator opens the administration page. 2. The system administrator inserts his/her credentials.
Exit conditions	The system administrator is now capable of doing his/her tasks.
Exceptions	If the credentials provided are not valid, the system shows an alert indicating the error.

3.4.3 Sequence diagrams

Login

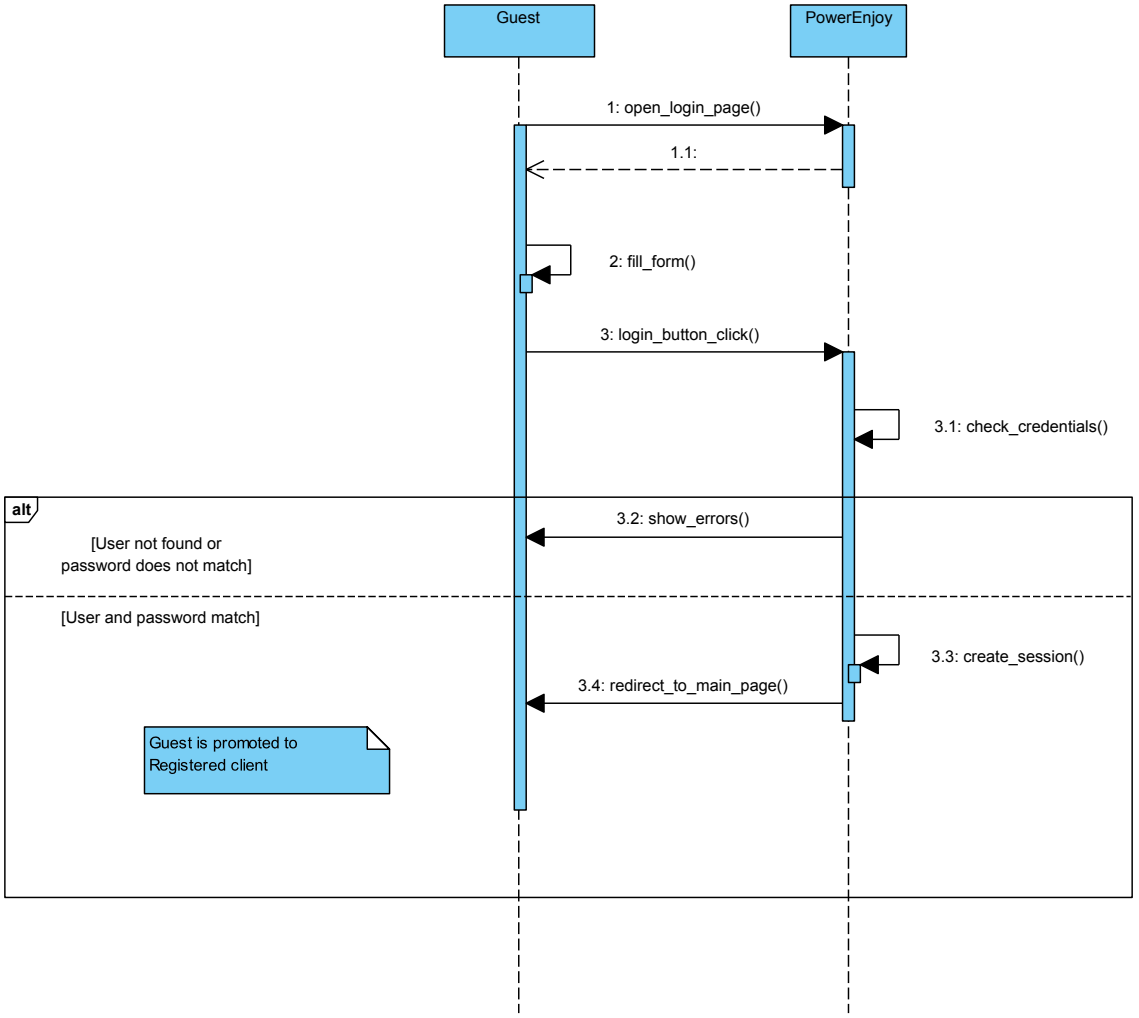

Figure 3.10: Login sequence diagram

Registration

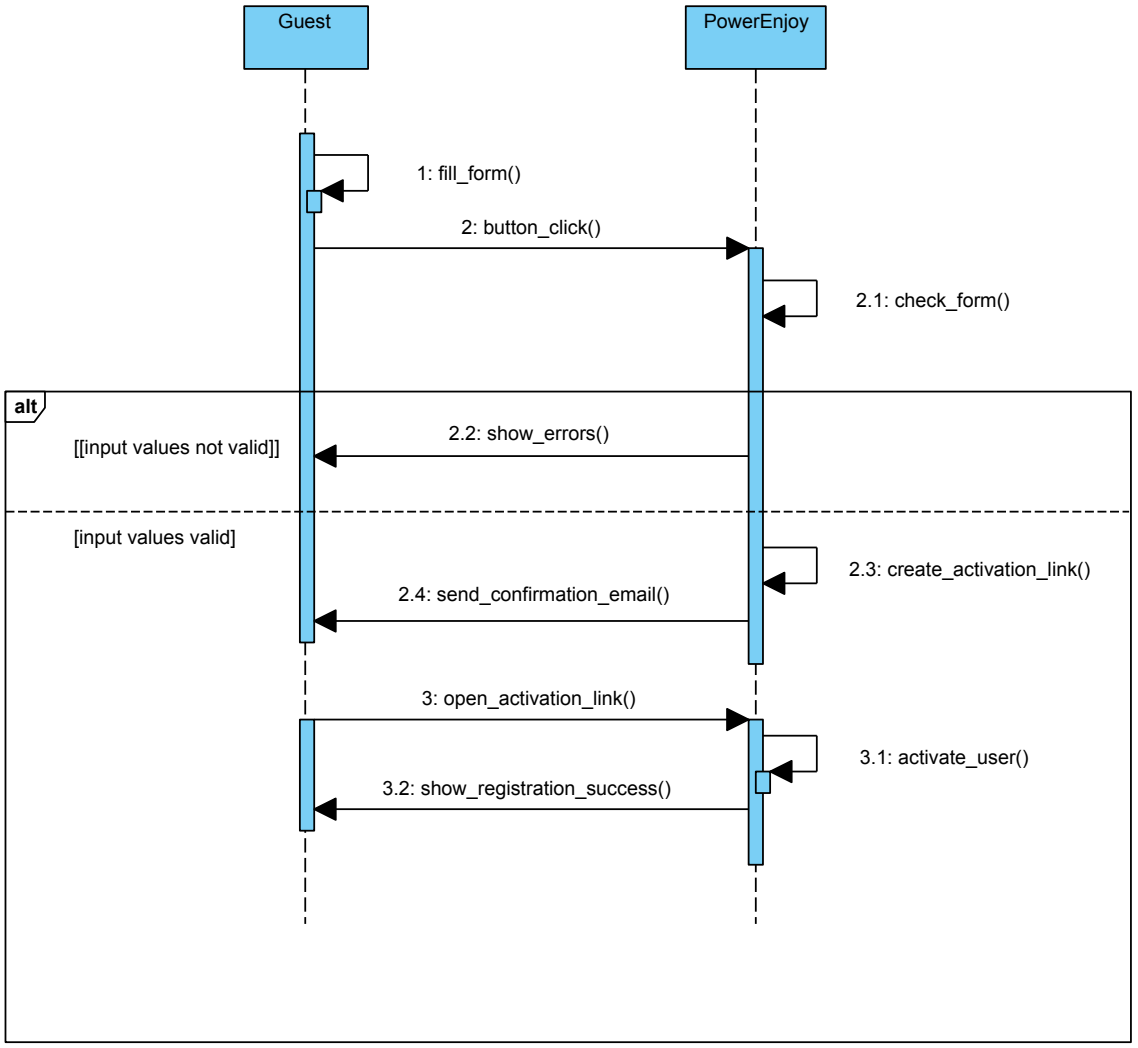

Figure 3.11: Registration sequence diagram

Car reservation through mobile or web application

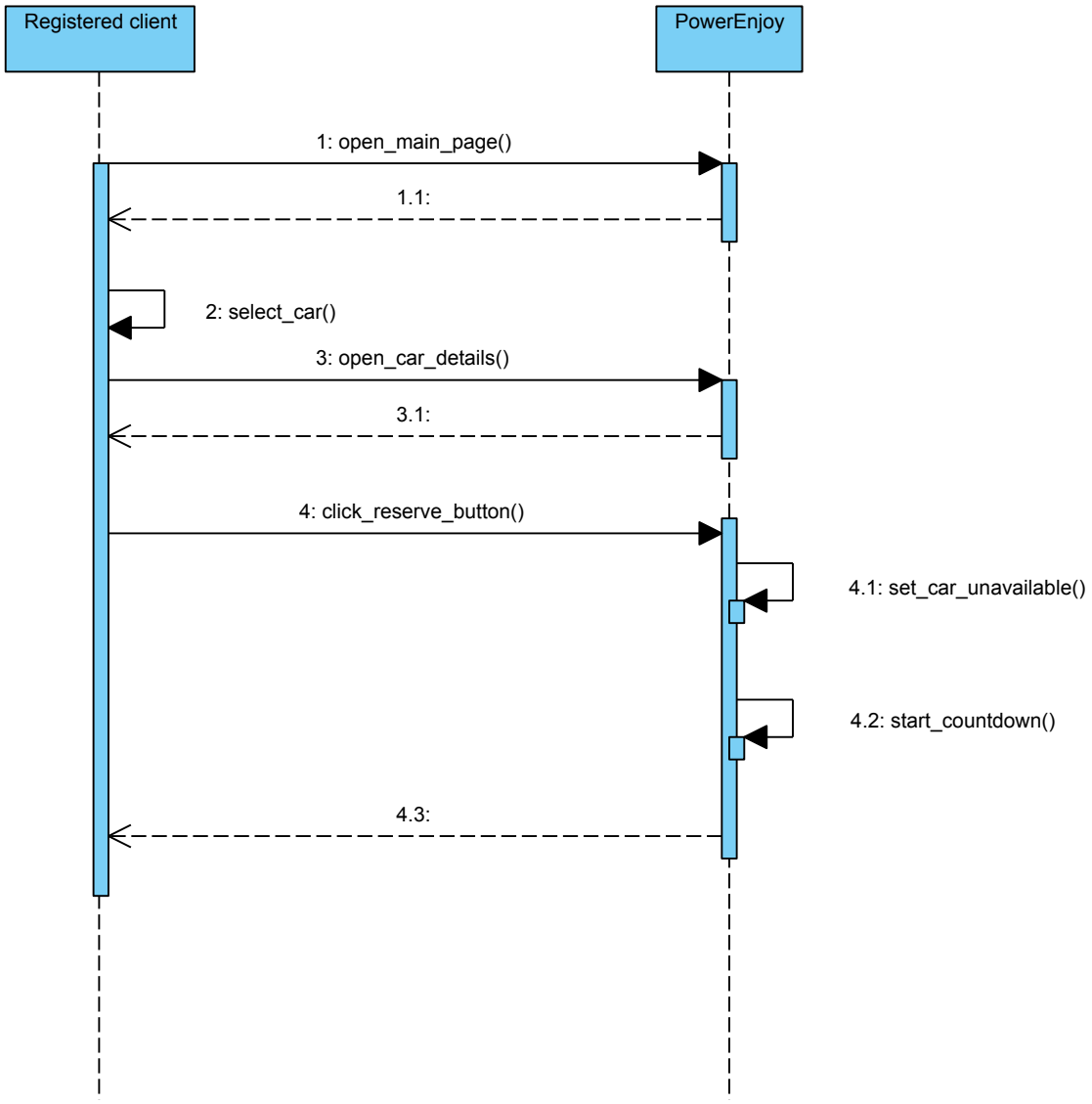

Figure 3.12: Rservation sequence diagram

Car reservation through SMS

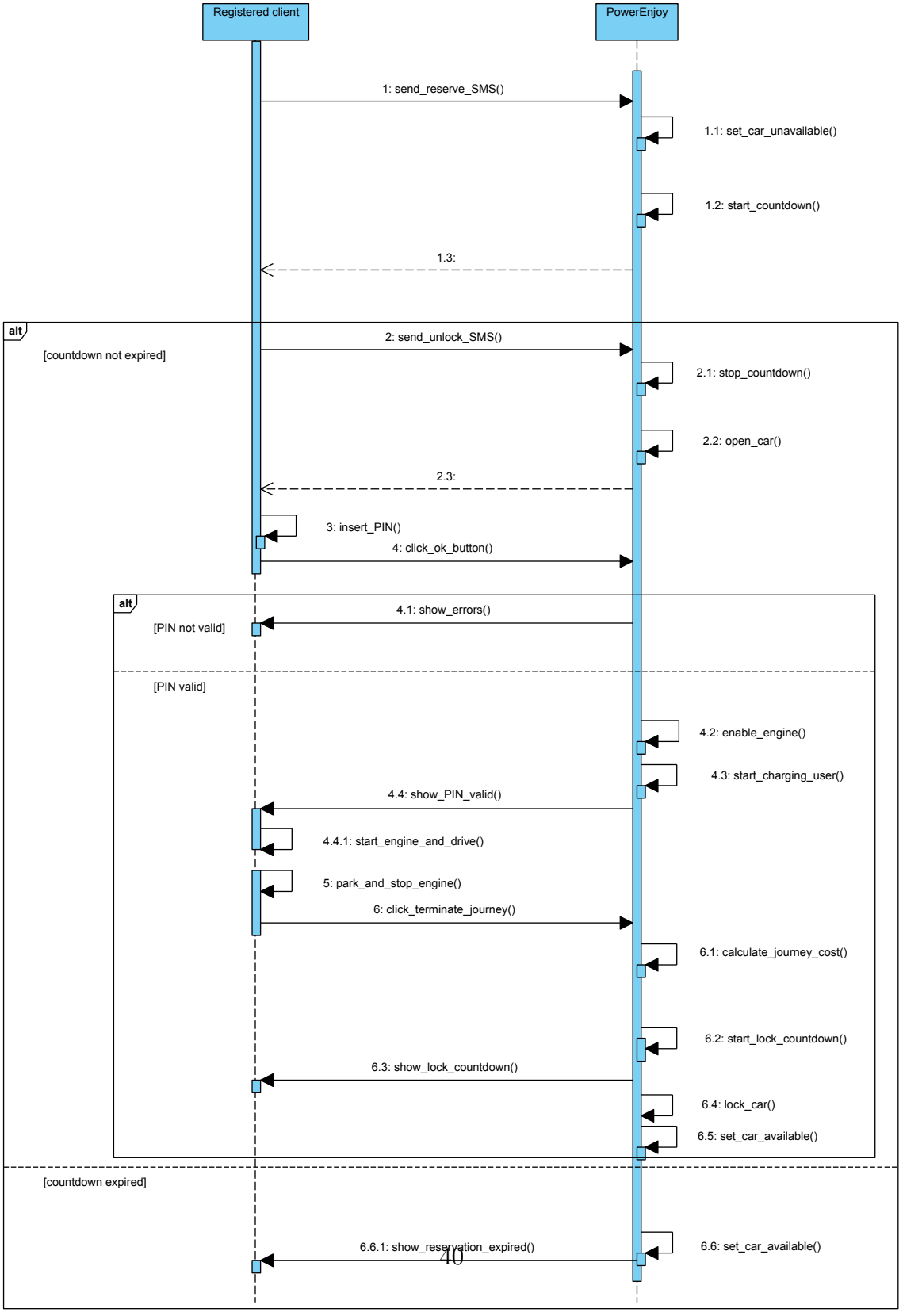

Figure 3.13: SMS reservation sequence diagram

Cancel reservation

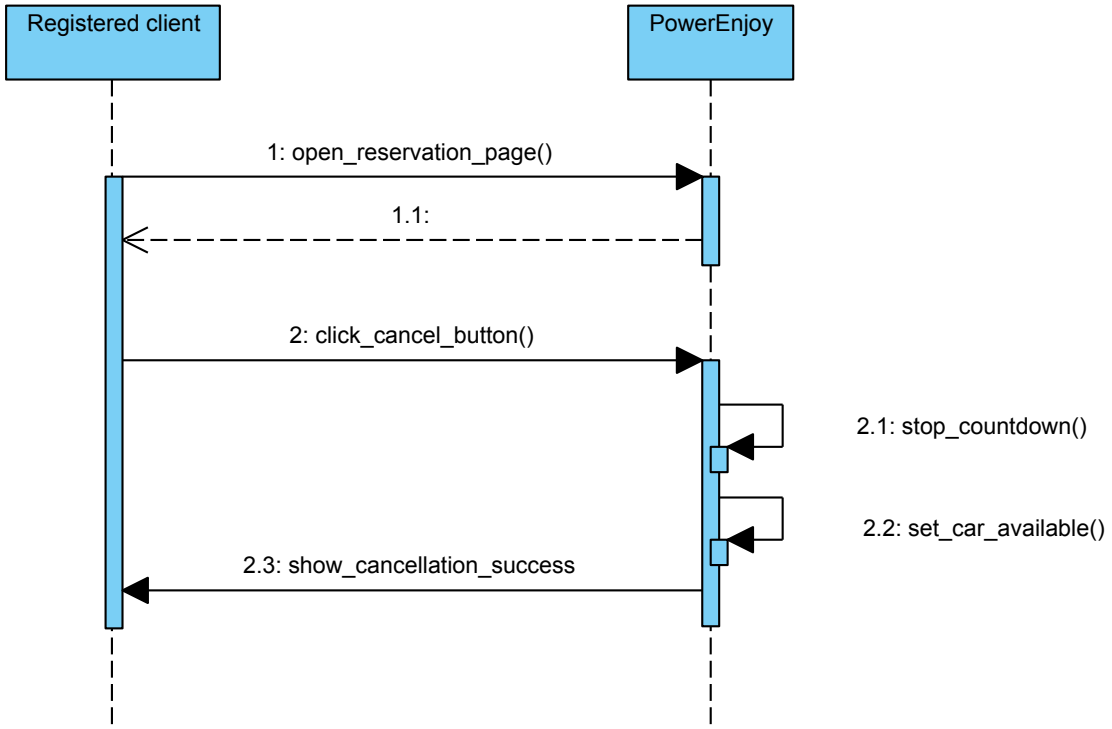

Figure 3.14: Cancel reservation sequence diagram

Car usage

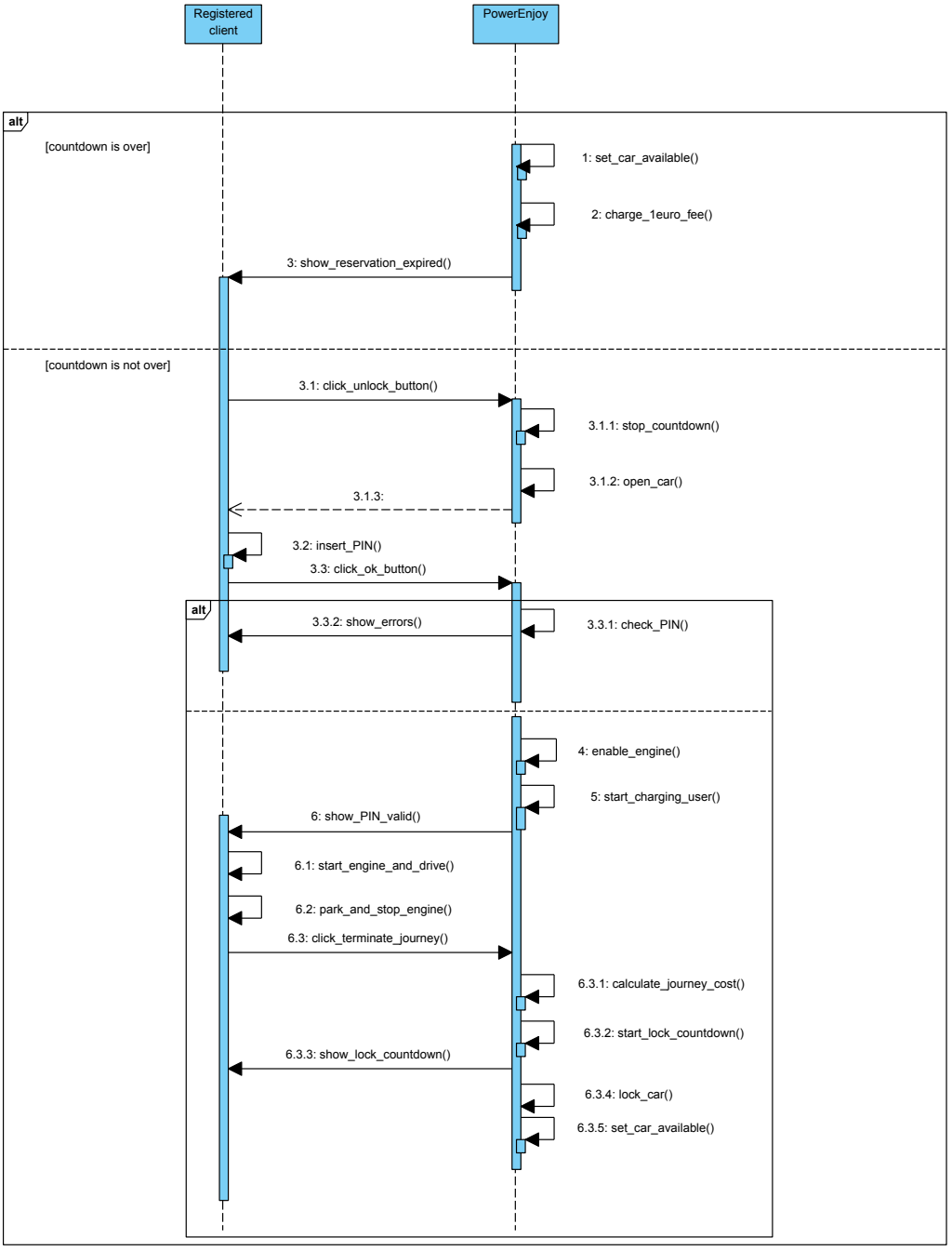

Figure 3.15: Car usage sequence diagram

3.4.4 Class diagrams

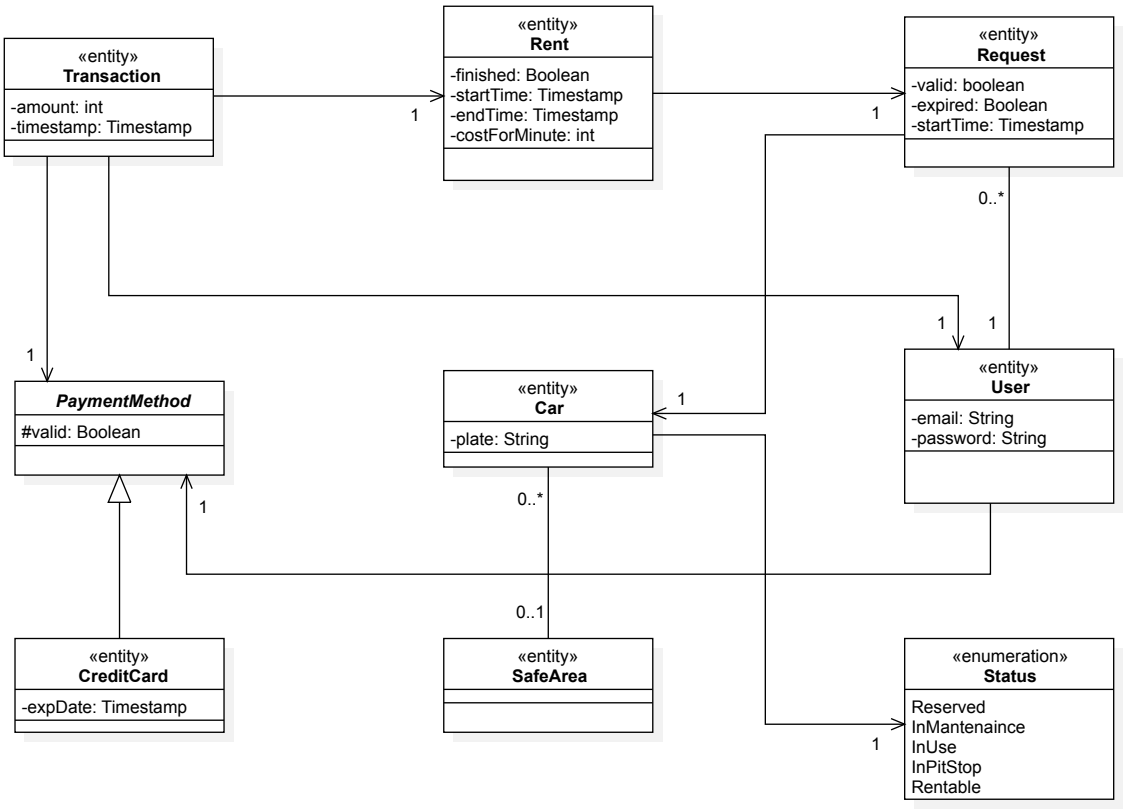

Figure 3.16: Class diagram

3.4.5 State chart diagrams

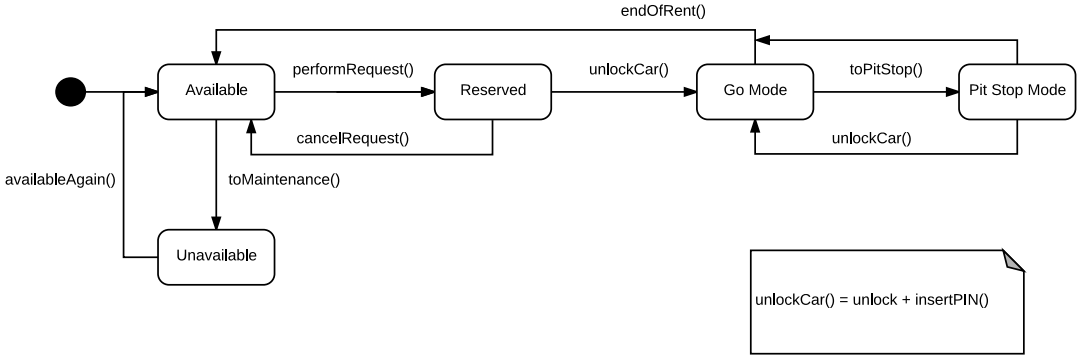

Figure 3.17: State chart diagram

3.5 Non-functionals requirements

3.5.1 Performance requirements

The system must guarantee that most of the operations have to be processed in less than 500 ms, in order to provide a real-time application.
If this target is achieved, the main boundary of the application is the user's internet connection bandwidth.

3.5.2 Design constraints

The system architecture is organized following the MVC pattern, meaning that all the data are stored in the server side, all the business operations are done in the server side (M and C of the acronym) and the goal of the mobile and web apps is only to give an user friendly interface to interact with the system (V of the acronym).
The server side application is written in Javascript using Express.js as a web framework.
The android app is written in Java.
The iOs app is written in Swift.
The windows phone app is written in XAML and C#.
The web application app is written in HTML5, CSS3 and Javascript using Riot.js as a UI library.

3.5.3 Software system attributes

Availability

The system must be accessible anytime, with a downtime ratio of 0.01 %.
In order to achieve this goal the system uses Nginx as a proxy server.
This software manages more instances of the application server in order to have:

- load balancing, in order to let more users use the application concurrently
- fault tolerance, in order to replace application server instances that incurred in fatal errors

Maintanability

The systems is highly maintainable thanks to the modular architecture that the MVC forces to use.
Each module is well documented, as well as the whole application, in order to help future developers to understand how it has been developed and organized.

Portability

The mobile application is developed for all the major mobile OSes.
In addition to this, the web application id design with a responsive layout in order to let users use it from any device with a modern browser.

3.5.4 Security

All the connection between clients' app (mobile and web) and server are under secure websocket (wss).
This protocol is an analog of HTTPS, that uses SSL to avoid man-in-the-middle attack, encrypting all the data exchanged among the two parties.
Furthermore all the sensible data stored in our DB are encrypted, meaning that also the system administrator does not have access to those data.

Chapter 4

Alloy

4.1 Alloy code

The following alloy code is created from the class diagram.

```
1  //#SIGNATURES
2
3
4  sig Boolean {
5    value : one Int
6  }
7  {value=0 or value=1}
8
9  sig DateTime{
10   timestamp: one Int
11 }
12 {timestamp>=0}
13
14 abstract sig PaymentMethod{
15   valid: one Boolean
16 }
17
18 sig CreditCard extends PaymentMethod{
19   expDateID: one DateTime,
20 }
21
22 sig User{
23   pmId: one PaymentMethod,
24 }
25
26 sig Transaction{
27   userID: one User,
```

```
28    rentID: one Rent,
29    paymMethID: one PaymentMethod,
30    amount: one Int
31  }
32  {amount > 0}
33
34  sig SafeArea{
35    carsIn: set Car
36  }
37
38  sig Car{
39    rentable: one Boolean,
40    inMaintenance: one Boolean,
41    inUse: one Boolean,
42    reserved: one Boolean,
43    inPitStop: one Boolean
44  }
45
46  sig Request {
47    carID: one Car,
48    userID: one User,
49    startTime: one DateTime,
50    valid: one Boolean,
51    expired: one Boolean
52  }
53
54  sig Rent {
55    requestID: one Request,
56    costForMinute: one Int,
57    systemTimeStart: one DateTime,
58    systemTimeEnd: one DateTime,
59      finished: one Boolean
60  }
61  {costForMinute > 0}
62
63  //#FACTS
64
65  fact OneUserOnePaymentMethodValid{
66    all u : User | (one pm : PaymentMethod | u.pmId = pm and pm.valid
        .value = 1)
67  }
68
69  fact ValidRequest{
70    all r : Request | r.valid.value = 1 and r.expired.value = 0 iff (
        one u : User | u = r.userID and one c : Car | c = r.carID)
71  }
72
73  fact NoRequestNotValidButExpired{
74    no req : Request | req.valid.value = 0 and req.expired.value = 1
```

```
75 }
76
77 fact OneRentOneValidNotExpiredRequest{
78   all r : Rent | one req : Request | r.requestID = req and req.
       valid.value = 1 and req.expired.value = 0
79 }
80
81 fact NoRentWithMoreThanOneRequest{
82   no disj r1,r2 : Rent | r1.requestID = r2.requestID
83 }
84
85 fact PaymentOnlyAfterEndRent{
86   all r : Rent | r.finished.value = 1 iff (one t1 : Transaction |
       t1.rentID = r and t1.userID = r.requestID.userID)
87 }
88
89 fact NoTransactionsSameRent{
90   no disj t1, t2 : Transaction | t1.rentID = t2.rentID
91 }
92
93 fact RentableCar{
94   all c : Car | c.rentable.value = 1 iff (c.inMaintenance.value = 0
        and  c.inUse.value = 0 and c.reserved.value = 0 and c.
       inPitStop.value = 0)
95 }
96
97 fact InMaintenanceCar{
98   all c : Car | c.inMaintenance.value = 1 iff (c.rentable.value = 0
        and  c.inUse.value = 0 and c.reserved.value = 0 and c.
       inPitStop.value = 0)
99 }
100
101 fact ReservedCar{
102   all c : Car | c.reserved.value = 1 iff (c.rentable.value = 0 and
        c.inUse.value = 0 and c.inMaintenance.value = 0 and c.
       inPitStop.value = 0 and (one r : Request | r.carID = c and r.
       valid.value = 1 and r.expired.value = 0))
103 }
104
105 fact InUseCar{
106   all c : Car | c.inUse.value = 1 iff (c.rentable.value = 0 and  c.
       reserved.value = 0 and c.inMaintenance.value = 0 and (one r :
       Request | r.carID = c and r.valid.value = 1 and r.expired.
       value = 0 and one rent : Rent | rent.requestID = r))
107 }
108
109 fact PitStopOnlyInUseState{
110   all c : Car | c.inPitStop.value = 1 implies c.inUse.value = 1
111 }
```

```
112
113 fact NoMoreThanOneRequestForCar{
114   no disj req1,req2 : Request | req1.carID = req2.carID and req1.
        valid.value = 1 and req1.expired.value = 0 and req2.valid.
        value = 1 and req2.expired.value = 0 and
115     req2.startTime.timestamp >= req1.startTime.timestamp and req2.
          startTime.timestamp - req1.startTime.timestamp =< 1
116 }
117
118 fact ReservedCarHasValidRequest{
119   all req : Request | (req.valid.value = 1 and req.expired.value =
        0) iff (req.carID.reserved.value = 1 or req.carID.inUse.value
        = 1)
120 }
121
122 fact NoAvailableCarOutSafeArea{
123   all c1 : Car | (c1.rentable.value = 1 or  c1.reserved.value = 1)
        iff (some sa : SafeArea | (one c2 : sa.carsIn | c2 = c1))
124 }
125
126 fact NoSafeAreaWithSameCarIn{
127   all c1 : Car |(c1.rentable.value = 1 or c1.reserved.value = 1)
        implies (no disj sa1, sa2 : SafeArea | c1 in sa1.carsIn and c1
         in sa2.carsIn)
128 }
129
130 fact RentNotFinishedCarInUse{
131   all r : Rent | r.finished.value = 0 implies (one c : Car | c = r.
        requestID.carID and c.inUse.value = 1)
132 }
133
134 fact RentFinishedNotCarInUse{
135   all r : Rent | r.finished.value = 1 implies (one c : Car | c = r.
        requestID.carID and c.inUse.value = 0)
136 }
137
138 fact PaymentMethodForATransactionAndARent{
139   all t : Transaction | (one r : Rent | t.rentID = r and r.finished
        .value = 1) and (one pm : PaymentMethod | pm.valid.value = 1
        and t.paymMethID = pm)
140 }
141
142 fact NoDuplicateDateTime{
143   no disj dt1,dt2 : DateTime | dt1.timestamp = dt2.timestamp
144 }
145
146 fact NoDuplicateBoolean{
147   no disj b1,b2 : Boolean | b1.value = b2.value
148 }
```

```
149
150 //#ASSERTS
151
152 assert NoRentWithSameRequest{
153   no disj r1,r2 : Rent | r1.requestID = r2.requestID
154 }
155 check NoRentWithSameRequest
156
157 assert TransactionHasPaymentMethod{
158   all t : Transaction | (one r : Rent | t.rentID = r and r.finished
        .value = 1) and (one pm : PaymentMethod | pm.valid.value = 1
        and t.paymMethID = pm)
159 }
160 check TransactionHasPaymentMethod
161
162
163 assert CarInUseRentNotFinished{
164   all r : Rent | r.finished.value = 0 implies (one c : Car | c = r.
        requestID.carID and c.inUse.value = 1)
165 }
166 check CarInUseRentNotFinished
167
168 assert SafeAreaContainsOnlyCarsReservedOrRentable{
169   all c1 : Car | c1.rentable.value = 1 or  c1.reserved.value = 1
        iff (one sa : SafeArea | (one c2 : sa.carsIn | c2 = c1))
170 }
171 check SafeAreaContainsOnlyCarsReservedOrRentable
172
173 assert OneUserOnePaymentMethod{
174   all u : User | (one pm : PaymentMethod | u.pmId = pm and pm.valid
        .value = 1)
175 }
176 check OneUserOnePaymentMethod
177
178 assert AllRentHasOneValidNotExipredRequest{
179   all r : Rent | one req : Request | r.requestID = req and req.
        valid.value = 1 and req.expired.value = 0
180 }
181 check AllRentHasOneValidNotExipredRequest
182
183 assert PaymentOnlyAfterEndRent{
184   all r : Rent | r.finished.value = 1 iff one t : Transaction | t.
        rentID = r
185 }
186 check PaymentOnlyAfterEndRent
187
188 assert CarReservedOnlybyOneRequest{
189   no disj req1,req2 : Request | req1.carID = req2.carID and req1.
        valid.value = 1 and req1.expired.value = 0 and req2.valid.
```

```
      value = 1 and req2.expired.value = 0 and
190     req2.startTime.timestamp >= req1.startTime.timestamp and req2.
          startTime.timestamp - req1.startTime.timestamp =< 1
191 }
192 check CarReservedOnlybyOneRequest
193
194 assert ReservedCarValidRequest{
195   all req : Request | req.valid.value = 1 and req.expired.value = 0
        implies req.carID.reserved.value = 1 or req.carID.inUse.value
        = 1
196 }
197 check ReservedCarValidRequest
198
199 //#PREDICATES AND SHOWS
200
201 pred show1 (){
202 #Boolean = 2
203 #Rent = 2
204 #Transaction = 1
205 #User = 5
206 #Car = 4
207 #Request = 2
208 #SafeArea = 3
209 #PaymentMethod = 2
210
211 }
212 run show1 for 10
213
214 pred show2 (){
215 #Boolean = 2
216 #Rent = 2
217 #Transaction = 0
218 #User = 6
219 #Car = 7
220 #Request = 3
221 #SafeArea = 2
222 #PaymentMethod = 3
223
224 }
225 run show2 for 10
```

4.2 Alloy generated worlds

In this section are presented the worlds generated by the alloy analysis software using the given show methods.

4.2.1 World 1

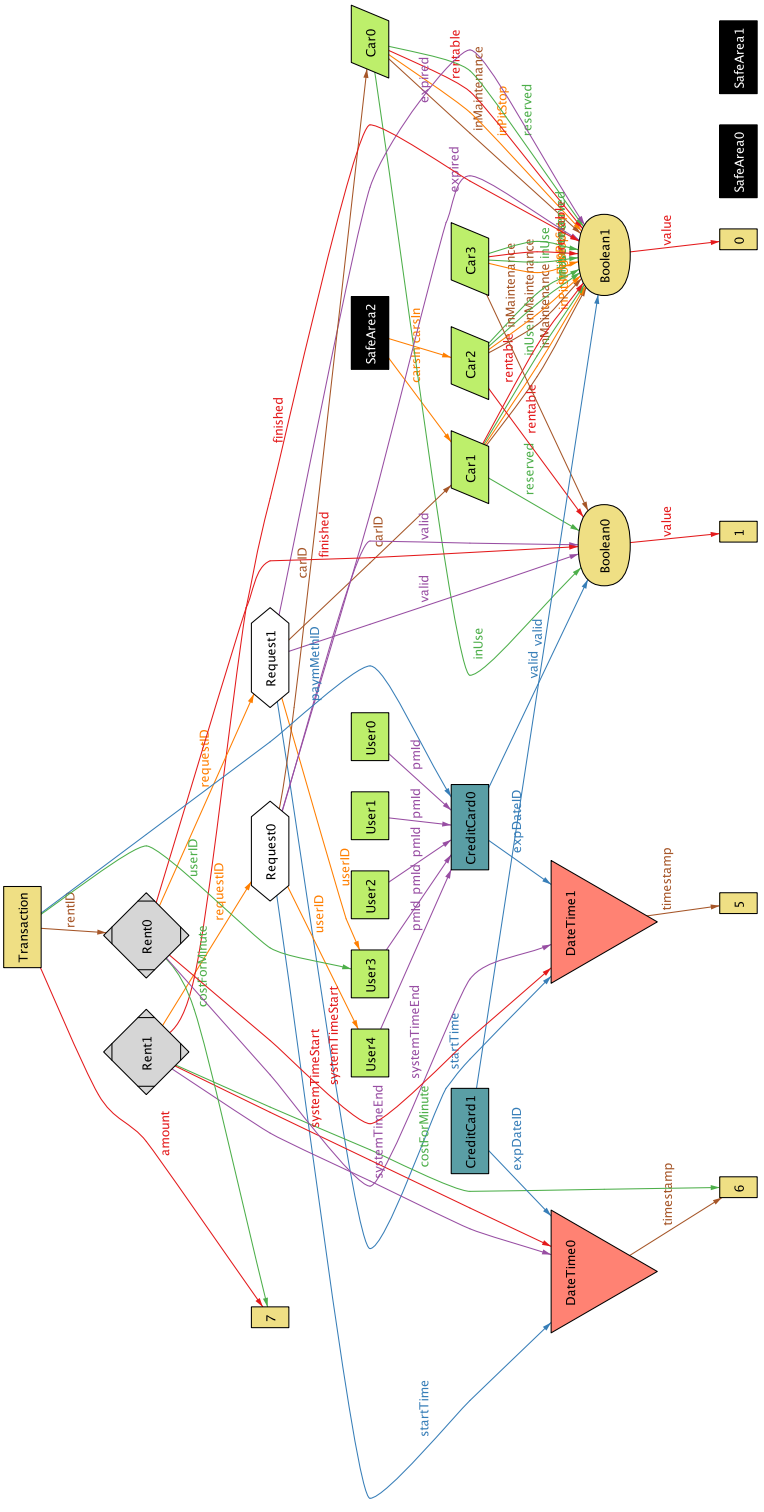

Figure 4.1: Alloy generated world #1

4.2.2 World 2

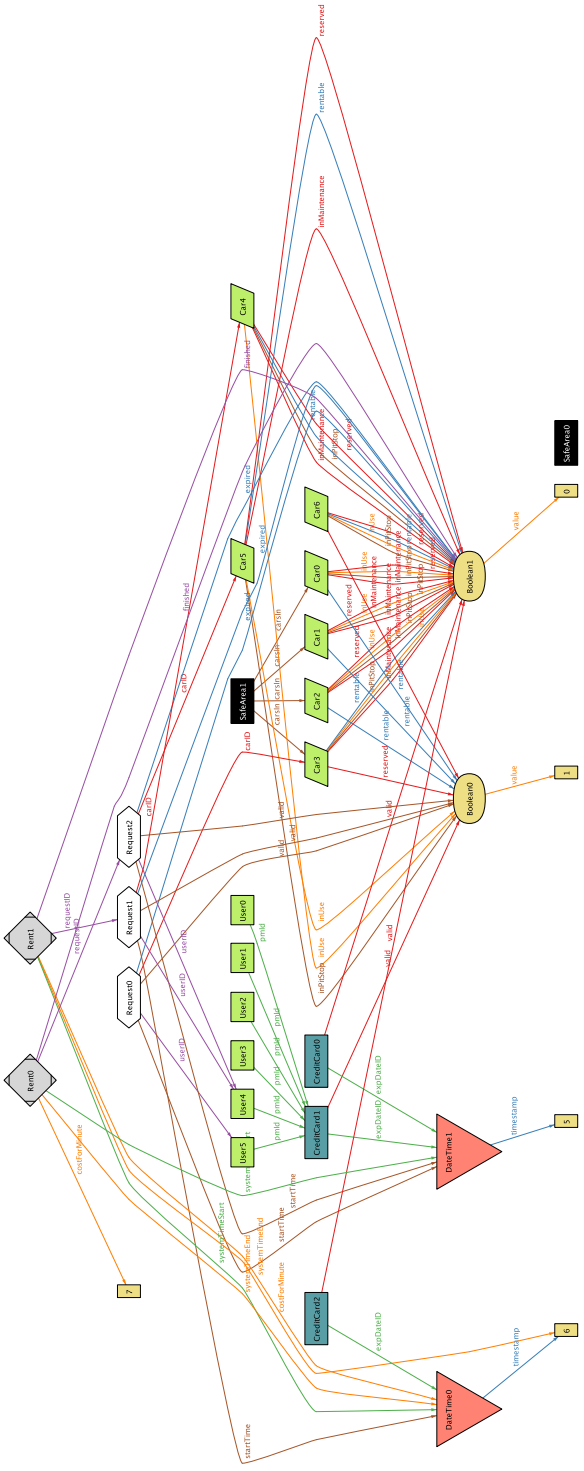

Figure 4.2: Alloy generated world #2

Chapter 5

Used tools and working hours

5.1 Used tools

The tools used to create this document are the following:

- MikTex 2.9: to format the document using LaTex.
- Pencilç to create mookups.
- Visual paradigm: to create sequence diagrams.
- LucidChart: to create use case, state chart and class diagrams.
- Alloy Analizer 4.2: to realize the model and to check its consistency

5.2 Working hours

	Alessandro	Roberta	Giorgio
28/10	1.5	1.5	1.5
29/10		0.5	
30/10	2		1
31/10	1.5	3	2.5
1/11	0.5	0.5	4
2/11	0.5	1	2.5
3/11	3	4	2
4/11	3	2	
5/11		3	3
6/11	1.5	3	
7/11	3.5	1.5	1.5
8/11	2	3	2
9/11	1.5		2
10/11	0.5		
11/11	3	3	3
12/11	3	3	3
13/11	1	1	1
Total	28	30	29

Changelog

v1.1

- Added caption to images.