Requirement Analisys and Specification Document

POWER Enjoy

Authors:
Cannas, Castiglioni, Loiacono

POLITECNICO
MILANO 1863

SUMMARY

PART 1: INTRODUCTION

PURPOSE

This is the Requirement Analysis and Specification Document (RASD from now on) for a digital management service required by PowerEnJoy. The aim of this document is to show the functional and nonfunctional requirements of the system-to-be, based on several important aspects:

- The needs expressed by the stakeholders;
- The constraints which it is subject to;
- The typical scenarios that will happen after its deployment.

The targeted audience is: Power EnJoy employees for the first two parts, and software engineers and developers who have to actually develop the service here described for the third part.

SCOPE

The System will be a software solution for the Power EnJoy's electric car-sharing service. The system will let users:

- to search and reserve a car;
- to open, drive, and leave the car in a fluid and comfortable way;
- to pay for for the ride in a simple and effective way.

In addition to this, the System will focus on a smarter organization of the vehicles deployed - resulting in a more efficient service for the citizens - and on encouraging users to keep a virtuous behaviour when using the service provided.

STAKEHOLDERS

The System has several stakeholders. The stakeholder on the supply side is the **Power EnJoy company**, whose aim is to expand and simplify its car-sharing service in order to make it easier and immediate to use and also to save money by tightening the reservation process and the users' recommended behaviour. On the demand side, the stakeholders are **Power EnJoy Service Users**. Their car-sharing experience will be enhanced using all the platform's functionalities and services. Last but not least, also **developers** are stakeholders because they will be compensated for the development of the system and will be those who will have to improve it over time.

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- **Guest:** a generic user not yet authenticated on the System and therefore not able to access any of the services of the Power EnJoy Application.

- **Registered User:** a user authenticated on the System who can access every service of the Power EnJoy Application. Throughout the document it is referred simply as User [with capitalized u].

- **Safe Area:** a parking Area owned by the Power EnJoy Company, and reserved to electric cars used in the service.

- **Charging Safe Area:** a Safe Area which has plugs connected to the electricity network, in order to recharge cars' batteries.

- **Car:** an electric car which is owned by the Power EnJoy Company and used within the Car Sharing Service.
    - **Available Car:** a car which is available for reservation and usage by Users.
    - **Reserved Car:** a car which is reserved by a User but not yet used for a ride.
    - **OnRide Car:** a car which is currently driven by a User, who has previously reserved it.
    - **Unavailable Car:** a car which is not reservable and searchable by Users.
    - **Unservable Car:** a car which is out-of service, and therefore not reservable and searchable by Users.

- **Reservation of a car:** a User reserve an available car up to one hour before using it.

- **Search of a car:** a User search for an Available car in a preselected map zone, in order to reserve it.

- **Money-Saving Option:** is an option the User has to discover Charging Safe Areas near their destination, and thus saving money.

REFERENCE DOCUMENTS

The following documents were used, in order to produce this one:

- Assignments AA 2016-2017.pdf (Assignment 1).
- International Standard ISO/IEC/IEEE 29148, Requirements engineering First edition 2011-12-01.
- Paper on the Green Move Project I and II.
- alloy-language-reference.pdf

PART 2: OVERALL DESCRIPTION

PRODUCT PERSPECTIVE

The software-to-be is going to be made of several parts: a front-end application for the Users, an application installed directly on the car and an API for developers.
Hardware interfaces are primarily the car's control unit, which is also equipped with a GPS system, a mobile communication system for 4G/LTE network and a Green e-Box (developed by Green Move Project), and Users smartphones, also with a GPS system and 4G/LTE.
Concerning the User Interface, it will be pretty similar to other related apps (for example, Car2Go or Share'NGo), since it will supply similar functions. It will primarily let them search, reserve, and pay for a car. On the car side, the application will let user to know the current level of battery, provide information about cost per minute and about emergency situation.
Finally, the software will make use of different API and Software platforms: Green Move System (developed by Green Move Project), in order to install java software on the car; the most famous smartphones operating System (iOs, Android), on which the User application will be deployed; several API to let the front-end application to contact external payment provider and use updated maps.

Hardware Interfaces

In order to correctly use the Power EnJoy application, a GPS system is required on every car, in order to track them. In addition to this, vehicles should have a mobile communication system for 4G/LTE in order to ensure the possibility of opening/locking cars via internet, monitoring their status and enabling data transfer between the mobile application and the software installed on the car.

User Interfaces

We have two different User Interfaces: a mobile application, and a screen mountend on the Car. For the sake of simplicity we present only the first one. These images are only explanatory on how the application will work, and are not, in any case, representing the final product.

Figure 1: The First page the User will be presented when he opens the application. Form this point he can only Register or Login.

Figure 2: The Login page. The User insert Email, Password, and then press the Submit Button.

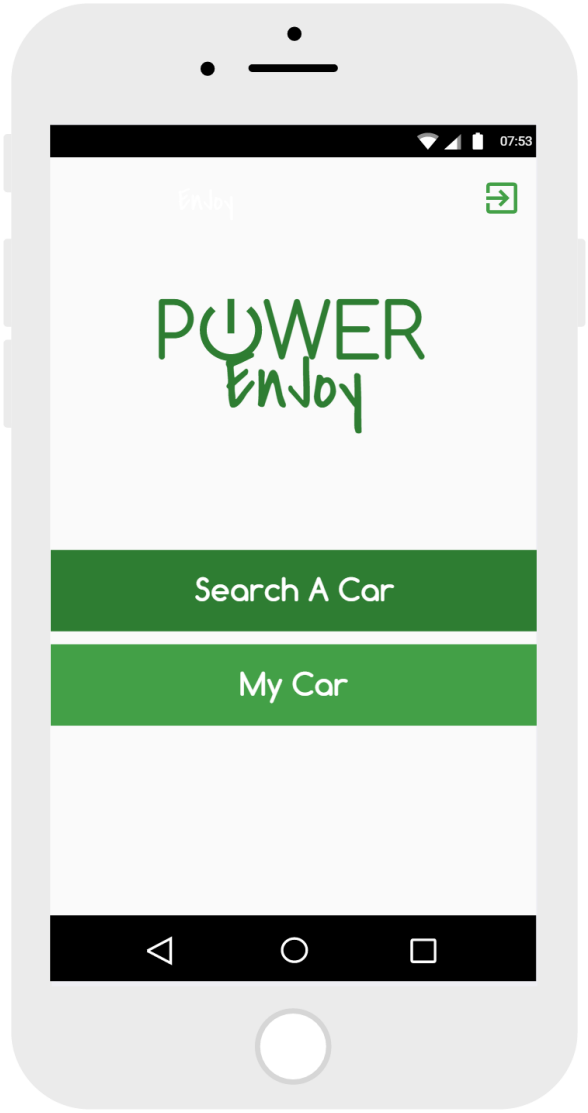


Figure 3: The personal Homepage of the Application. Form here the User can access any of the Power EnJoy Services.

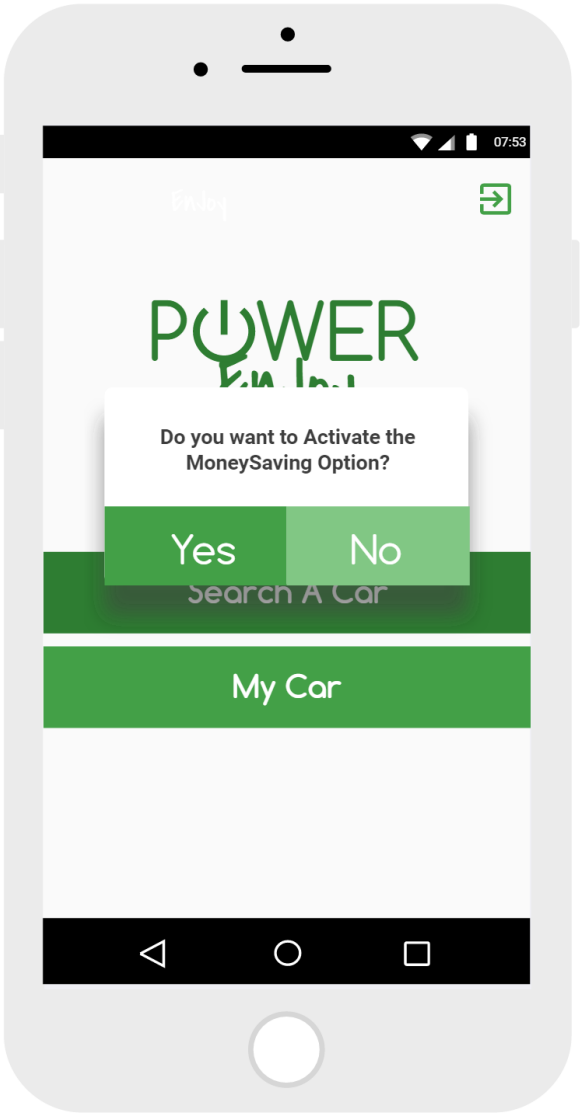


Figure 4: When the user press the Search Car Button, the System show him the possibility to activate the MoineySaving Option

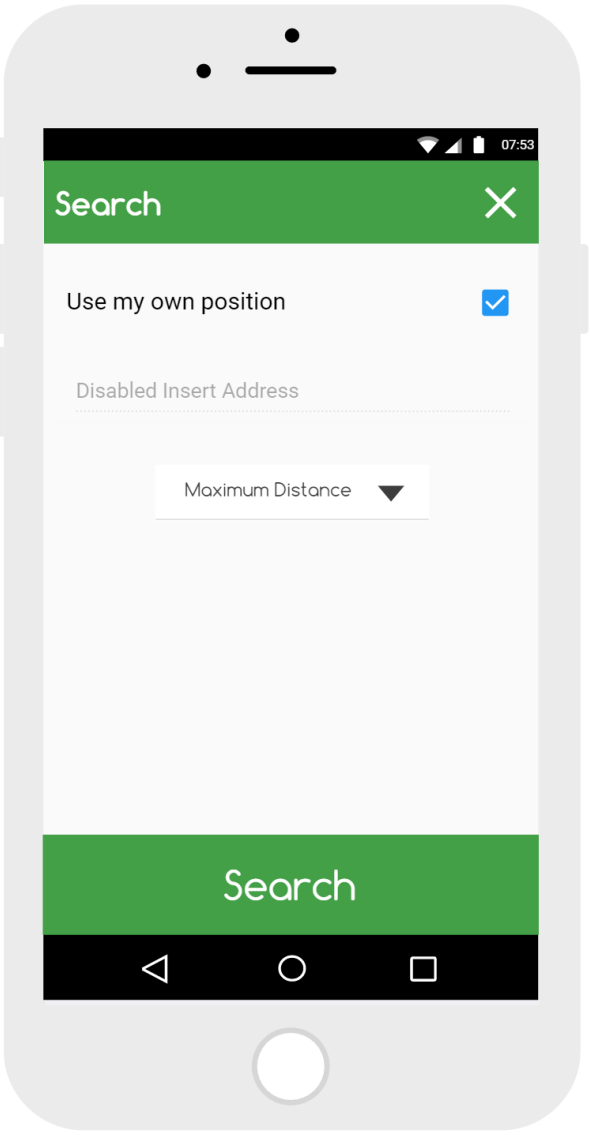


Figure 5: Search Form the user can fill. If he activates his position, he cannot insert an address to perform the search.

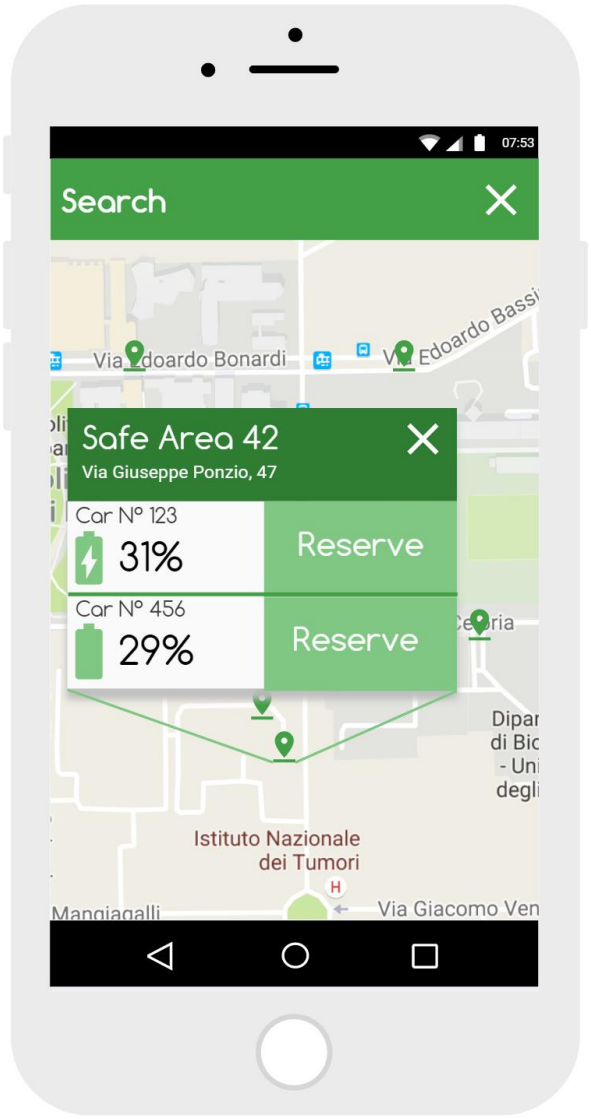


Figure 6: Maps with cars that can be reserved by the User.

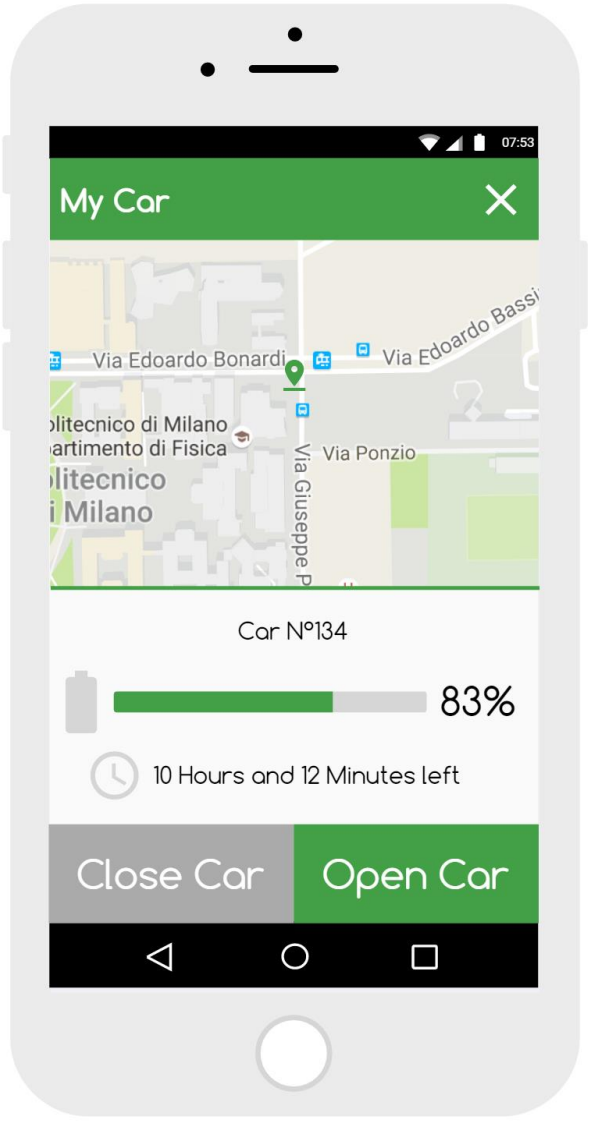


Figure 7: When a User has reserved or is using a car, he can access every information about the car he is using. He can also open and close it directly from his smartphone.

Software Interfaces

- Maps API
  - Google Maps
  - Source: https://developers.google.com/maps/

- Payments API
  - Stripe
  - https://stripe.com/it

- Driver License Validation API
  - EUCARIS
  - https://www.eucaris.net/

- Operating Systems:
  - Android (User software)
  - iOS (User software)
  - Green Move System (Car)

PRODUCT FUNCTIONS

Here is a list of all goals the Power EnJoy Application will accomplish:

Goal1: Users must be registered in order to access any of the Power EnJoy Services; they should provide their credentials every time they want to access the System Services.

Goal2: Users must be able to find *available* cars within a certain distance from their current location or from a specified address.

Goal3: Users must be able to reserve one car at time, if there are any *available* in a specified area.

Goal4: If a car is not picked-up within one hour from the reservation, the reservation expires and the user pays a fee of 1 EUR.

Goal5: Users should be able to save money if they leave the car in a Safe Area with Charging Station near to a destination suggested by the System; Suggested Safe Areas are selected by the System in order to ensure a uniform distribution of the cars.

Goal6: Users should be able to open and access the car they have reserved; however, cars should not be opened in too much advance. From the moment Users access the Car, they should not be charged for the expiration of their reservation.

Goal7: Users should be charged for the use of the Car from the moment they ignite the engine and they should be aware of how much they are spending through an interface device inside the Car.

Goal8: Users should be able to make stops during their rides. In any case, the time of these stops is still charged on the User, they can be outside a Safe Area, but the stop can last only for a determined amount of time (12 hours), otherwise they will be charged for the overall renting plus a fixed fine.

Goal9: Users should be supported in case of accidents and empty battery.

Goal10: Users should be aware of the level of charge of the Car's battery, in order to stop to the nearest Safe Area and plug the Car to a power station, and should pay a fixed fee if he leaves the battery empty.

Goal11: Charging should be stopped from the moment Users park the Car in a Safe Area and shut down the engine.

Goal12: Users should be charged after the end of their ride. They should be advised of all the information about their ride and the payment in a short time after they have left the Car.

Goal13: Cars automatically lock when the users leave them in the safe area. Then, they are available for other reservations.

Goal14: At the end of every ride, users are granted with charge discounts or more fees: 10% discount for users who offer a ride to more than two passengers; 20% discount for users who leave the car with no more than 50% of the battery empty; 30% discount for users who park the car in a Safe Area with a power grid station, and take care of recharging the battery. If the user instead leaves the car at more than 3KM from the nearest power station, or the battery empty for more than the 80%, it should be charged of another 30% on the total bill.

DOMAIN ASSUMPTIONS

- Maximum number of Passengers is 4.
- Users' Driver Licences are validated by an External Licence Provider within 24 hours.
- Payment information are handled by an External Payment Provider.
- The estimation of the duration of the battery of the car for every kilometer is known.
- The car is always connected to the internet (mobile network, 3G…)
- The user is always connected to the internet.
- The user has always a mobile phone charged.
- If attached to a charging station, a car cannot start.
- The position of all the Safe Areas is pre-defined and known.
- The user who reserves a car, also drives it.
- Cars have embedded software systems and sensors that trace their GPS position, number of passengers, battery level, engine status, and kilometers traversed.
- It is always possible for the system to access Car's status.
- The payment fee is computed in EUR/MIN.
- It is impossible to put the Car in charge if the engine is started.
- After a ride, the user leaves the car in a reasonable time.
- All the cars are functioning and works without any trouble.
- If the user offers a ride to other passengers, they share the entire ride with the user.
- The user always closes the car's door when they leave it after a ride.
- Cars can only be put in charge inside Charging Safe Area.
- The number of plugs in a Charging Safe Area is equal to its capacity.
- Unservable and Unavailable Cars are always put back in charge in a Charging Safe Area by an Operator.

PART 3: REQUIREMENTS

FUNCTIONAL REQUIREMENTS

Goal1: Users must be registered in order to access any of the Power EnJoy Services; they should provide their credentials every time they want to access the System Services.

- **Req1:** The System shall prevent guests from accessing any service before being registered.
- **Req2:** The System shall allows guests to register.
- **Req3:** The System shall requires name, surname, e-mail, credit card's details and driver licence number, from guests who tries to register.
- **Req4:** The System shall register only guests whose credentials were validated.
- **Req5:** When a guest is registered, The System shall send him a Personal Account Password on his e-mail address.
- **Req6:** The System shall prevent users from accessing any service before being logged in.
- **Req7:** The System shall allows registered users to log in with their e-mail and Personal Account Password.
- **Req8:** The System shall prevent users from having multiple sessions at the same time.

Goal2: Users must be able to find *Available* Cars within a certain distance from their current location or from a specified address.

- **Req1:** The system shall request the User an address - or his current location - and a maximum distance within which cars are searched.
- **Req2:** The System shall search for *Available* Cars within a specified area delimited by the maximum distance selected and the specified location.
- **Req3:** The System shall show the User the position, battery level, and charging information of every *available* Car within the specified area.
- **Req4:** The System shall not show the user *Unavailable* Cars, *Reserved* Cars, and Cars which are under 20% of battery.
- **Req5:** If no Car is available in the selected Area, the System shall allow the user to widen the search.

Goal3: Users must be able to reserve one car at time, if there are any *Available* in a specified area.

- **Req1:** The System shall change the status of the selected car as *Reserved.*
- **Req2:** The System shall prevent the user to reserve and use more than one car at time.

- **Req3:** When an user selects a car, The System shall request the External Payment Provider if the user is able to pay for the longest possible ride.
- **Req4:** The System shall allow the reservation of a car only if payment information about the user are verified.

Goal4: If a Car is not picked-up within one hour from the reservation, the reservation expires and the user pays a fee of 1 EUR.

- **Req1:** The system shall change the Car status from *Reserved* to *Available* if the reservation lasted for more than an hour, and the Car was not picked up..
- **Req2:** The system shall require the external payment provider to charge the user for a fee of 1 EUR if the user's reservation expires.

Goal5: Users should be able to save money if they leave the Car in a Safe Area with Charging Station near to a destination suggested by the System; Suggested Safe Areas are selected by the System in order to ensure a uniform distribution of the Cars.

- **Req1:** Before searching for an *Available* Car, the System shall request if the user wants to activate the Money-Saving Option.
- **Req2:** If The Money-Saving Option is active, the System shall request the user's destination address, and a maximum distance within which Safe Areas are searched.
- **Req3:** The System shall search for free Charging Safe Areas within the specified area.
- **Req4:** If one Charging Safe Area is available, the System shall suggest it to the user.
- **Req5:** If two ore more Charging Safe Areas are available, the System shall suggest the one with less number of cars plugged in.
- **Req6:** If no Charging Safe Area is available in the selected Area, the System shall allow the user to widen the research.
- **Req7:** When there are more than one Charging Safe Area with the same minimum number of plugged-in Cars, the system shall suggest the one nearest to the destination.
- **Req8:** When searching for a Car with the Money-Saving Option active, the system shall show the User only Cars which have enough battery to travel to the suggested Charging Safe Area.

Goal6: Users should be able to open and access the Car they have reserved; however, cars should not be opened in too much advance. From the moment Users access the Car, they should not be charged for the expiration of their reservation.

- **Req1:** The system shall show the User an interface through which he could open the car.
- **Req2:** The system shall check the User's GPS position.

- **Req3:** The system shall prevent the User from opening the Car if its position is farther than 5 meters from it.
- **Req4:** The System shall request the car to open if the User who reserved it is within 5 meters from it.
- **Req5:** The system shall change the state of the Car from *Reserved* to *OnRide* from the moment the User opens it.

Goal7: Users should be charged for the use of the Car from the moment they ignite the engine and they should be aware of how much they are spending through an interface device inside the Car.

- **Req1:** The system shall start charging the user for the amount of time he/she is using the car from the moment he/she ignites the engine.
- **Req2:** The System shall show the User the actual charge through the Car's screen.

Goal8: Users should be able to make stops during their rides. In any case, the time of these stops is still charged on the User, they can be outside a Safe Area, but the stop can last only for a determined amount of time (12 hours), otherwise they will be charged for the overall renting plus a fixed fine.

- **Req1:** The System shall check the Car's position.
- **Req2:** If the Car's position is outside a Safe Area, the System shall presents the User an interface through which he/she can close and open the Car.
- **Req3:** The System shall keep charging the User for the amount of time he/she leaves the car parked outside a Safe Area.
- **Req4:** The System shall keep track of the time the Car is left outside a Safe Area.
- **Req5:** If a Car is parked outside a Safe Area for more than 12 hours, the System shall send a message to an Operator.
- **Req6:** The System shall change the car status to *Unservable*.

Goal9: Users should be supported in case of accidents and empty battery.

- **Req1:** The System shall check for problems and control unit warning messages
- **Req2:** The System shall send a message to an Operator, with all the necessary information (Car's position, type of disservice).
- **Req3:** The System shall change the car status to *Unservable*.
- **Req4:** When a *Unservable* car is put in charge in a Charging Safe Area, The System shall change its status to *Unavailable*.

Goal10: Users should be aware of the level of charge of the Car's battery, in order to stop to the nearest Safe Area and plug the Car to a power station, and should pay a fixed fee if he leaves the battery empty.

- **Req1:** The System should show the User the battery level of the Car, and the remaining time until it runs out of power.
- **Req2:** The System should notify the User if the battery level is low.
- **Req3:** The System should show the User the nearest Safe Area where he can leave the Car.
- **Req4:** If the Car completely runs out of power, the System shall charge the User for the time he/she used the Car, plus a fixed fee.

Goal11: Charging should be stopped from the moment Users park the Car in a Safe Area and shut down the engine.

- **Req1:** The System shall check the Car's engine.
- **Req2:** The System shall stop charging the User from the moment the Car's position is inside a Safe Area and its engine is turned off.

Goal12: Users should be charged after the end of their ride. They should be advised of all the information about their ride and the payment in a short time after they have left the Car.

- **Req1:** The System shall require the payment to the external payment provider when the User leave the car and the car is closed.
- **Req2:**The System shall present the user a summary of his ride, and information about how much it costed when the payment is complete.

Goal13: Cars automatically lock when the users leave them in the safe area. Then, they are available for other reservations.

- **Req1:** The System shall change the Car status from *OnRide* to *Available* when it is locked in a safe area and the battery is equal or over 20%.
- **Req2:** The System shall change the Car status from *OnRide* to *Unavailable* when it is locked in a safe area and the battery is under 20%.
- **Req3:** The system shall lock the car when the user leaves it and is at least 5 meters away from it.
- **Req4:** The system shall not allow to restart Car's engine after then the car is parked in a safe area and the users shut it down.

Goal14: At the end of every ride, users are granted with charge discounts or more fees: 10% discount for users who offer a ride to more than two passengers; 20%

discount for users who leave the car with no more than 50% of the battery empty; 30% discount for users who park the car in a Safe Area with a power grid station, and take care of recharging the battery. If the user instead leaves the car at more than 3KM from the nearest power station, or the battery empty for more than the 80%, it should be charged of another 30% on the total bill.

- **Req1:** The system shall consider applying a 10% discount on the user's last ride if the number of passengers is greater or equal than 2.
- **Req2:** The system shall consider applying a 20% discount on the user's last ride if the battery level is full for more than 50% when the car's engine is stopped in a Safe Area.
- **Req3:** The system shall consider applying a 30% discount on the user's last ride if he/she parks the car in a Safe Area and recharges the battery.
- **Req4:** The system shall apply only the greatest discount in case of multiple applicable discounts.
- **Req5:** When the system recognizes that the car is left more than 3KM far from the nearest power grid station, the system shall consider applying an overcharge of the 30% on the last ride bill.
- **Req6:** When the system recognizes that the car has its battery empty for more than the 80%, the system shall consider applying an overcharge of the 30% on the last ride bill.
- **Req7:** The system shall apply only one overcharge.
- **Req9:** The system shall computes the overall bill as a summation of the applicable discount, overcharges and the initial ride charge.
- **Req10:** the system shall not apply overcharges on the last ride if the user put the car in charge.

SCENARIOS

Sc1: John is planning a new trip in Milan. John does not have a personal car, so he decides that for his stay in Milan he will need to use the Power EnJoy Application, so he can use an electric car for his travellings. When he opens the app, he is told by the application that he needs to register or to sign in (if already registered) in order to access all Power EnJoy services. He does not have an account, so he is redirected to a form which he compiles, filling it with his personal data. After a few minutes, he receives a confirmation message, on his email address, which contains a personal password to access all Power EnJoy services.

Using the app, John tries to sign in, using the received password, but he makes a mistake, and he enters a wrong username. The application notifies the error and asks him if he wants to try again. John tries again, being careful of not making any mistake, and this time he successfully signs in, being now able to use all Power EnJoy car sharing services.

Sc2: Mary Jane is a busy school teacher who needs everyday to cross the city because she has two different classes in two different schools. She does not have enough money for a car, so she uses the Power EnJoy Application, and their car sharing system. She signs in in the service, and selects the *Search and Reserve* option. She is then redirected to a form asking her if she wants to activate the *Money-Saving Option*: she selects no. then she's asked to enter her current location or a specified address, and the maximum range to which extend the search. She selects her current location and a range of 2 km. The application shows her a map with the available cars, and their battery level. Then, she selects the nearest one. The system tells her the car is now reserved and that the reservation will last only one hour.

Sc3: John has reserved a car with the Power EnJoy Application before going away from his workplace, in the center of Milan. Due to an unexpected problem he was forced to stay there and work for nearly another hour. After leaving his workplace, he receives a message on his phone: the car he selected was no more reserved, and a fee of 1 EUR was charged on his credit card. John tries then to reserve the same car again, but the car is no more available, and so he has to reserve another car, farther from his current position.

Sc4: Antoine is an artist. He doesn't have much money, so, when using the Power EnJoy app, he activate the *Money-Saving Option* as soon as he begins to search for a car. The System asks him for a destination address and a maximum range within which a Safe Area is searched; he enters them. Unfortunately, there are no Safe Area in the specified Area, so the System ask Antoine if he wants to widen the search. Antoine accepts, and the System find Safe Area with a Charging Station 2km far from his destination. Antoine then proceeds at searching and reserving a car as usual.

Sc5: Paolo Golini is a retired high-school teacher living in Milan, that 5 years ago sold his car. Today he has to go with his wife to a mall far from the place where they

live, so he decides to use a Power EnJoy car to go to there and get back home. Before going out, Paolo opens his Power EnJoy Application and reserves a car. Then he calls his wife and together they go toward the Safe Area where Paolo reserved their Car. When they are 100m far from it, Paolo opens the PowerEnjoy Application again and clicks on the *MyReservation* button to open their Car. However, the app shows a warning message asking Paolo to get closer to the Safe Area. Thus, when they are entering it Paolo clicks the *Open Car* button again and this time the app gives an affirmative response: in fact, when they get to the Car they find it already open.

Sc6: When Paolo and Clara get to the mall, Paolo parks the Car in a normal parking. When he turns off the engine, his phone thrills: the PowerEnjoy Application has sent him a notification. He and Clara get out of the Car, and then Paolo from the *MyReservation* button of the System clicks the *Close Car* button. They ear the sound of the Car's doors closing, and go shopping.

Sc7: Paolo and Clara are finishing their commissions, then they go outside the mall. Approaching the parking site, Paolo uses the Power EnJoy Application to inform the system that they are nearby. When they arrive to the car, he unlocks it through the app and they get into it. Paolo sets the car seat and mirrors, and then ignites the engine. The car screen lights on, and displays the actual charge of Paolo and Clara's car's renting.

Sc8: Paolo and Clara are getting back home. When they arrive at their house, Paolo helps Clara unload the grocery from the car; then he rides towards the Safe Area where they usually pick up the car for going to the mall. He parks the car, and stops the engine. He gets out from the Car and carefully checks if he has forgotten anything inside it and if all the doors have been closed. Then he walks away, and when he's far enough he ears the sound of the car being locked. When he gets home, Paolo checks his phone: the PowerEnjoy Application has notified him the total charge for the car trip from the mall to his house, and informs him that the payment will be charged on his credit card.

Sc9: Luca and Marco are two neighbours. They also both work in two close places and always search for how to go to work in a faster and cheaper way. Luca discovered Power EnJoy some time ago and started using occasionally. One evening he finds out that there is a 10% discount if the car takes two or more passengers. Then, he asks Marco to share the car with him. The day after they go to work with the same Power EnJoy car and, at the and of the ride ,the Power Enjoy Application shows Luca a 10% discount on the last bill.

Luca is not totally satisfied by the price so searches other ways to save money and discovers that there is a 30% discount if he parks the car in a Charging Safe Area, and he puts the car in charge. He remembers that there is a Power Enjoy Charging Safe Area near home so, when he comes back to work with Marco, parks there and starts charging the car. Leaving the car, the app notifies him of the 30% discount on the last ride.

Sc10: Carlo and his girlfriend Sara are ex-colleagues who both graduated at the Politecnico of Milan. Being both two passionate cinephiles, they usually go the the cinema on weekends, using the Power EnJoy Car Sharing Service. This Saturday, unfortunately, while waiting at a semaphore during their ride toward the cinema, a car behind them crashed into their vehicle. Both them and the other car's driver are fine; however, the Power EnJoy car is seriously damaged.

Sara, ears her phone thrilling: the Power EnJoy Application notified her that an External Operator has already been informed of the accident and that he is going to their position to take care of the situation.

Sc11: Maria is an accustomed user of the Power EnJoy Car Sharing Service. Today he picked up a Car to make visit to an old friend of her. While getting back home, she notices that the battery of her Car is quite low: indeed, in a few minutes the car screen turns red, showing a Low Battery Alert, and all the nearest Safe Areas where she can leave the Car and recharge it.

However, Maria is quite convinced that the remaining power of the battery should allow her to get to the Safe Area near to her house. What happens though, is that the power is not enough, and her Car stops few kilometers away from her destination. Her phone thrills, but she knows what it is going to read: the Power Enjoy Application has informed her that an Operator is arriving at her position to remove the Car, and that the whole ride cost plus a fee for the covering of the Operator's intervention have been charged on her credit card.

USE CASE DIAGRAM

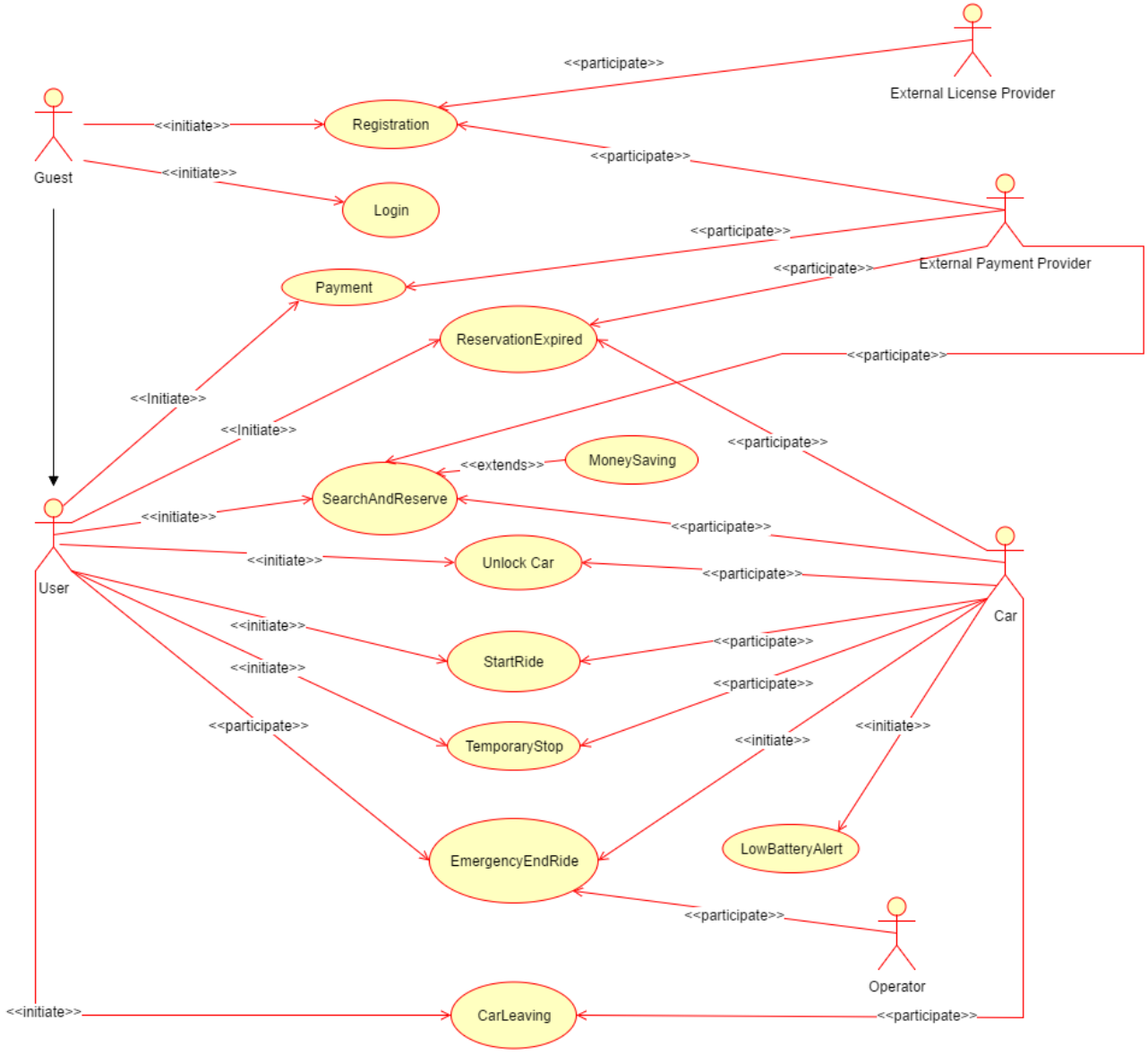


Figure 8: Use cases and actors.

USE CASES

Use Case 1	
Name	Registration
Description	Registration to the System of a Guest
Participating Actors	Guest, External Payment Provider, External License Provider
Entry Condition	None
Flow of Events	• The System shows the Homepage to the Guest. • The Guest clicks on the *Register* button. • The system shows a form on screen. • The Guest inserts his name, surname, e-mail address, driver license number, and credit card's details; then he clicks on the *Submit* button. • The System require the external providers to check for the authenticity of the driver licence and credit card's details. It then waits for an answer. • If the e-mail, credit card details, and driver licenses are validated the System sends an e-mail to the user with a Password to access the services.
Exit Condition	The Guest is now a User registered to the service.
Exceptions	• If the Guest is already registered, the system prevents him from registering again, showing a warning message. • If the Guest fills the form with inconsistent or not well-formed data, the system send him a warning message, notifying that it was not able to complete the registration request.

Use Case 2	
Name	Login
Description	A Registered User logs in the system
Participating Actors	Guest, Registered User
Entry Condition	The Guest must have registered
Flow of Events	• A Guest access the application, entering the Homepage. • The Guest clicks on the *Login* button. • The System shows a form on screen. • The Guest inserts his email and Password and clicks on the *Submit* button. • The System checks if the inserted Username and Password are valid, and match with an existing account.
Exit Condition	The Guest is redirected to a personal Home Page and recognised as a Registered User from now on.
Exceptions	• If the Guest fills the form with inconsistent or not well-formed data, or he is already logged in, the system shows him a warning message, and return him to the previous page.

Use Case 3	
Name	SearchAndReserve
Description	Search and reservation of a car by a registered user, within a maximum range from his current position, or a specified address.
Participating Actors	Registered User, Car, External Payment Provider
Entry Condition	The User must be logged in
Flow of Events	• From his personal Homepage, the User clicks on the *Search and Reserve* Button. • The System ask the user if he wants to activate the *Money-Saving Option*. • The User does not activate the *Money-Saving Option*. • The System shows a form for the search. • The User compiles the form, selecting his own position, and the maximum range at which perform the search. • The System search for every *available* car in the selected zone. • The System shows a map - centered on the position selected - which also shows the position of every available cars within range and their battery level. • The User selects one of the available cars, to reserve it. • The System request the External Payment Provider if the user is able to pay for the longest possible ride. • The System reserves the car for the user, changing its status to *Reserved*.
Alternative Flows	• Step 1, 2, 3 and 4 perform as descripted above. • The User enters the selected address from which he wants to perform the search and the maximum range at which perform it. • Step 6, 7, 8 and 9 perform as descripted above.
Exit Condition	The User has now made a reservation for a Car within a selected region.
Exceptions	• If there are no available cars within the selected region, the System notifies the user and asks him if he wants to perform the search again within a larger range. • If the User selects a non existent address, the System notifies him, and redirects him on his personal Home

	Page. ● If the external payment provider notifies that the User doesn't have enough money on his credit card, the System send a warning message to the User, redirecting him on the Homepage.

Use Case 4	
Name	ReservationExpired
Description	A reservation which lasted for more than one hour expires.
Participating Actors	Registered User, Car, External Payment Provider
Entry Condition	A Registered User reserved a car one hour ago
Flow of Events	• The System recognizes that an hour has passed since the car was reserved. • The System set the car as *Available* again, and cancel the reservation. • The System send a notification to the User, to inform him that his reservation has expired. • The System require the External Payment Provider to apply a fee of 1 EUR on the User credit card.
Exit Condition	The reserved car is now available again for being searched and reserved.
Exceptions	No Exception for this use case.

Use Case 5	
Name	MoneySaving
Description	The user search for a Safe Area with a charging station near his destination, in order to save money.
Participating Actors	Registered User
Entry Condition	The User has clicked on the *Search and Reserve* button.
Flow of Events	• The System ask the user if he wants to activate the *Money-Saving Option*. • The User does activate the *Money-Saving Option*. • The System shows a form on the screen. • The User enters the destination address and the maximum range at which perform the search of a Charging Safe Area. • The System shows a map - centered on the selected address - which also shows the position of the Charging Safe Area suggested. • The User click on the *Continue Search* button.
Exit Condition	The User continue the search as described in Use Case 4.
Exceptions	• If there are no available Charging Safe Areas within the selected region, the System notifies the User and asks him if he wants to perform the search again within a larger range. • If the User selects a non existent address, the System notifies him, and redirects him on the Homepage.

Use Case 6	
Name	UnlockCar
Description	Car unlocking and access by the user
Participating Actors	Registered User, Car
Entry Condition	• The User must be logged in • The User should have reserved a Car • The Car reservation should be not expired or • The User should have already accessed the Car • The User has parked the Car outside a Safe Area • The User must be logged in
Flow of Events	• From his personal Homepage, the User access to the *MyReservation* function. • The System shows the details of the User reservation. • The User clicks on the *Unlock Car* button. • The System checks the user position and the Car position. • If the User position is within 5 meters from the reserved car, the System sends a message to the Car. • The Car unlocks. • The System tags the car as *OnRide*.
Exit Condition	The Car is open and the User can access to it.
Exceptions	• If the User too far from the car, the System shows him an error message, asking him/she to get closer to the Car. • If the User has already accessed the Car and parked it, the System tags the Car as *OnRide*.

Use Case 7	
Name	StartRide
Description	The User ignites the Car and the System starts charging him/her for its use
Participating Actors	Registered User, Car
Entry Condition	The User must have accessed a *Reserved* Car
Flow of Events	• The User clicks on the *Start/Stop* button to ignite the Car's engine. • The Car's send a message to the System. • The System recognizes the Car has been turned on. • The System starts charging the User. • The Car shows through a screen the actual charging of the User. • The System tags the car as *OnRide*.
Exit Condition	The User uses the Car and the System keeps track of the amount of time he/she uses it.
Exceptions	No exceptions for this Use Case.

Use Case 8	
Name	TemporaryStop
Description	The User parks the Car outside a Safe Area with the intention to continue his/her ride.
Participating Actors	Registered User, Car
Entry Condition	• The User must have logged in. • The User must have already accessed the Car. • The User must have parked the Car outside a Safe Area.
Flow of Events	• The User clicks on the *Start/Stop* button to turn off the Car's engine. • The Car sends a message to the System. • The System recognizes the Car has been turned off. • The System sends a notification to the User. • The System shows to the User a *Lock Car* button within the *MyReservation* page. • The User gets out of the Car. • The User clicks the *Lock Car* button. • The Car closes.
Exit Condition	The System continues charging the User for the amount of time he/she leaves the Car parked outside the Safe Area.
Exceptions	No exceptions for this Use Case.

Use Case 9	
Name	EmergencyEndRide
Description	The Car stops for an emergency (empty battery, car crash) or because the car was *OnRide* for more than 12 hours. The System acknowledges its position and informs an Operator.
Participating Actors	Registered User, Car, Operator
Entry Condition	• The Car status must be *OnRide*. • An emergency must be notified by the Car's control unit. or • The Car status must be *OnRide*. • The Car was *OnRide* for more than 12 hours.
Flow of Events	• The Car sends an *Emergency Message* to the System with his position and the necessary information. • The System gathers the information sent by the Car. • The System sends a message to the Operator with all the information necessary. • The System sends a notification to the User asking him/her to wait for the arrival of the Operator. • The System tags the car as *Unservable*.
Exit Condition	The System charges the User if the stop is due to his/her fault.
Exceptions	No exceptions for this Use Case.

Use Case 10	
Name	LowBatteryAlert
Description	The System alerts the User of a low battery level of its Car and shows him some Safe Areas where he could park and plug the Car to a power grid.
Participating Actors	User, PowerEnjoy Car
Entry Condition	The Car status must be OnRide.
Flow of Events	• The Car sends a *Warning Message* to the System with all information about the battery level. • The System shows a *Low Battery Alert* to the User through its screen. • The System acknowledges the Car's position and remaining autonomy time. • The System searches the reachable Safe Areas from the Car's position. • The System shows the User the position of the Safe Areas through the screen.
Exit Condition	The User knows where he can leave the Car before running out of power.
Exceptions	There are no exception for this Use Case.

Use Case 11	
Name	CarLeaving
Description	the User park the Car in a Safe Area and leave it.
Participating Actors	Registered User, Car
Entry Condition	the User enters in a Safe Area with an *OnRide* Car
Flow of Events	• The User stops the engine. • The Car sends a message to the System to inform that the Engine is Stopped. • The System checks the position of the Car. • The System stops charging the User. • The User goes away from the Car. • The System closes the Car. • The System set the Car as *Available*
alternative flow	• The User put the Car in charge.
Exit Condition	The Car is locked and available again for another Ride.
Exceptions	• If the User doesn't stop the engine, when he leaves the Car, the System notifies him that he is being still charged for the ride. • If the Car is under 20% of battery, the System changes Car's status to *Unavailable.*

Use Case 12	
Name	Payment
Description	The Systems request the payment for a ride, and notifies the User the total charging.
Participating Actors	Registered User, External Payment Provider
Entry Condition	• The Car is parked inside a Safe Area • All Car's doors are closed
Flow of Events	• The User has walked away from the Car. • the System consider applicable discounts and overcharges. • The System computes the total charge. • he System requires the payment to the External Payment Provider • The System notifies the User the total charge for its ride and the information about the payment. • The System tags the Car as *Available*.
Exit Condition	The payment is notified to the user.
Exceptions	No exception for this Use Case

SEQUENCE DIAGRAMS

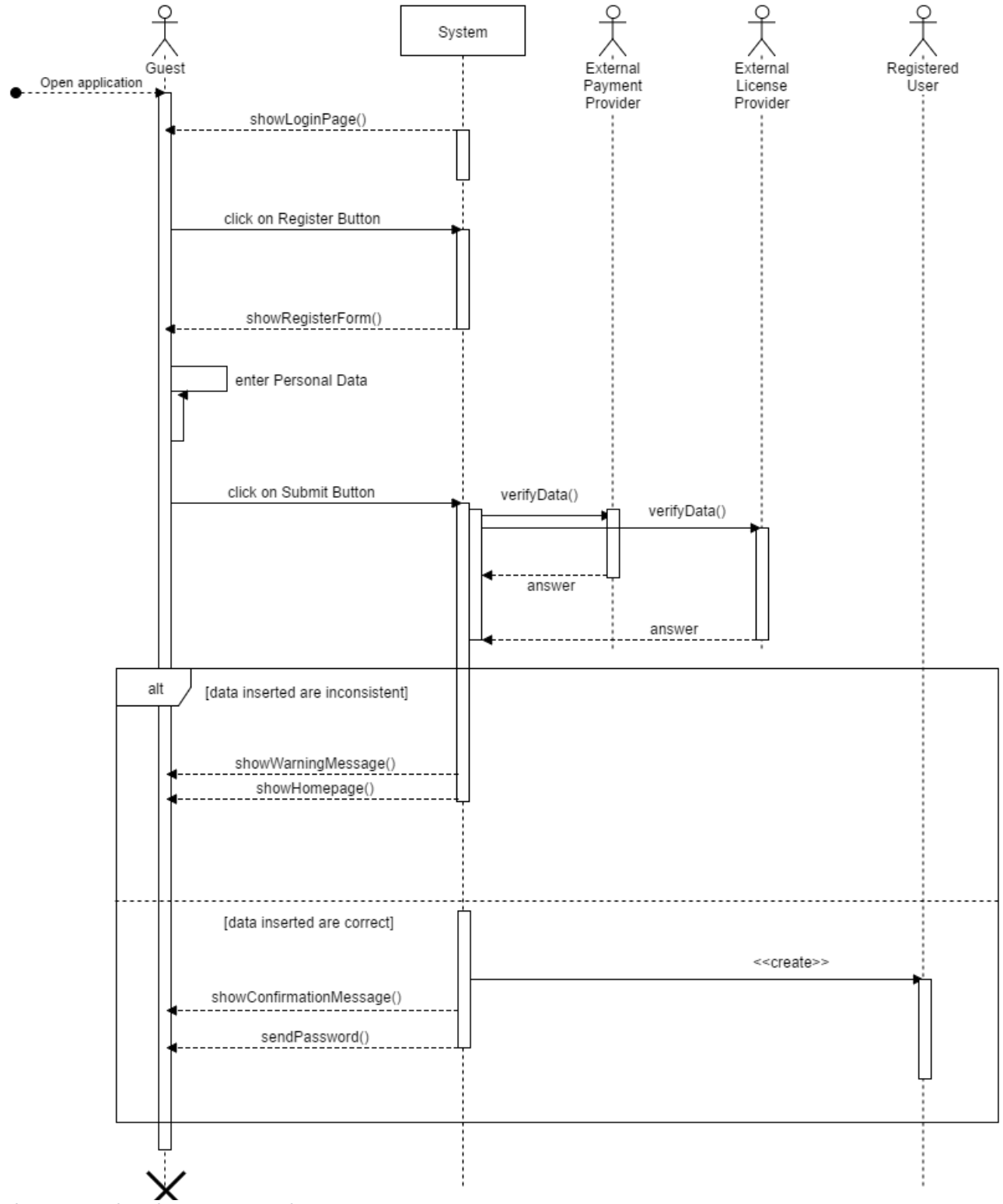


Figure 9: Registration Sequence Diagram

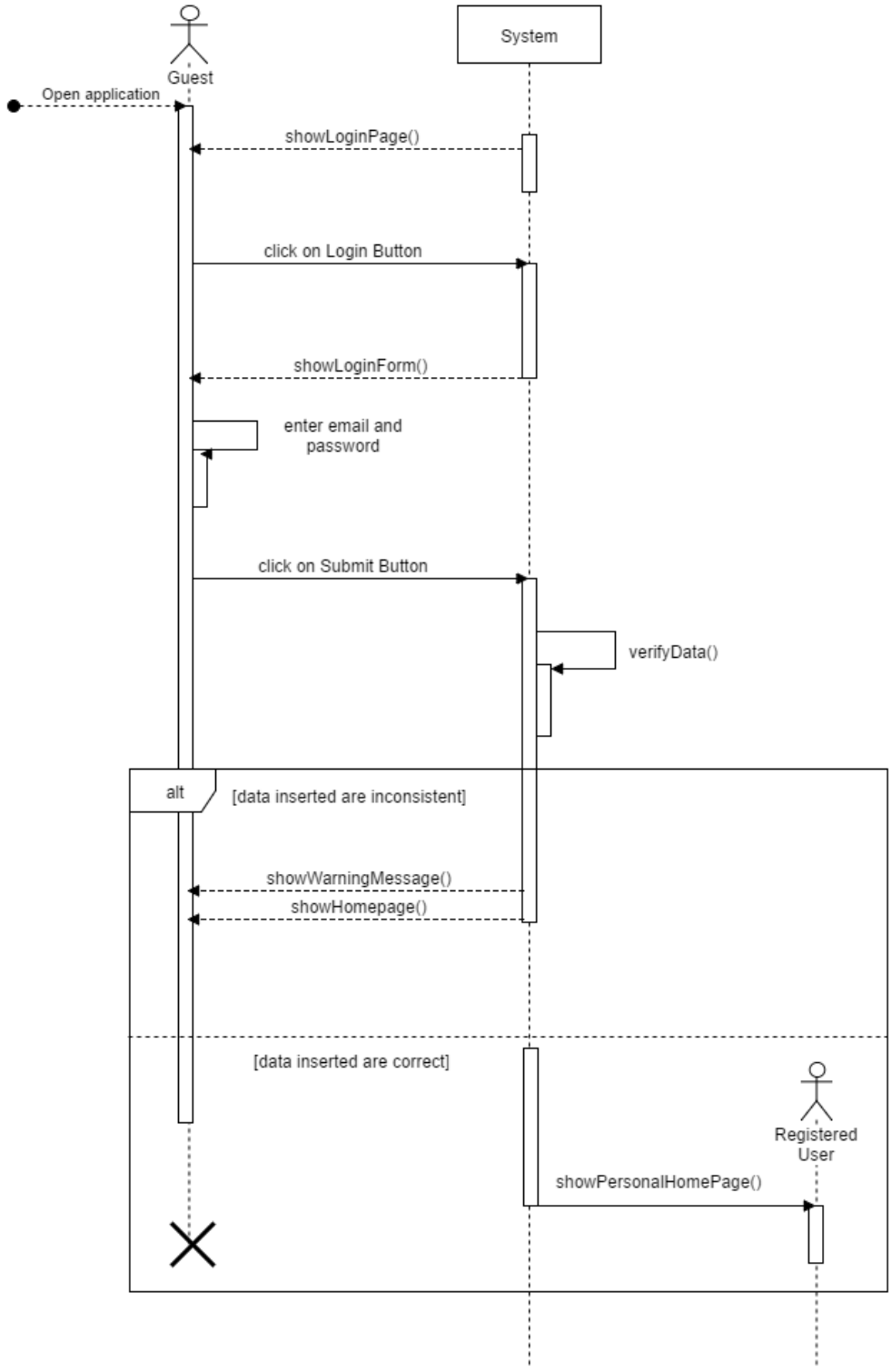


Figure 10: Login Sequence Diagram

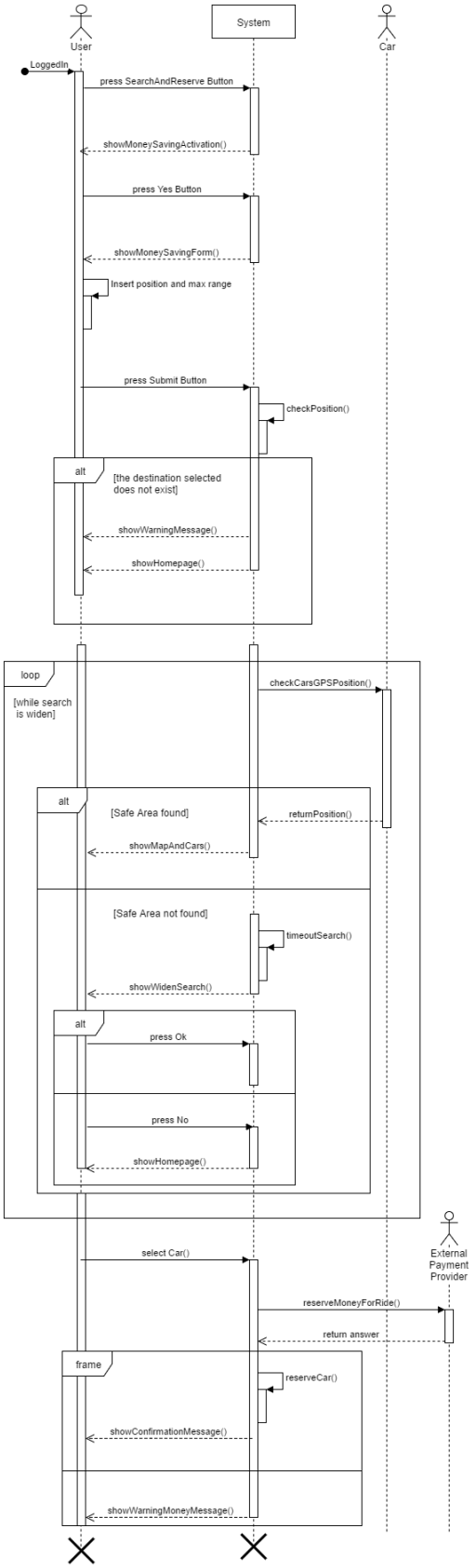


Figure 11: Search And Reservation Sequence Diagram

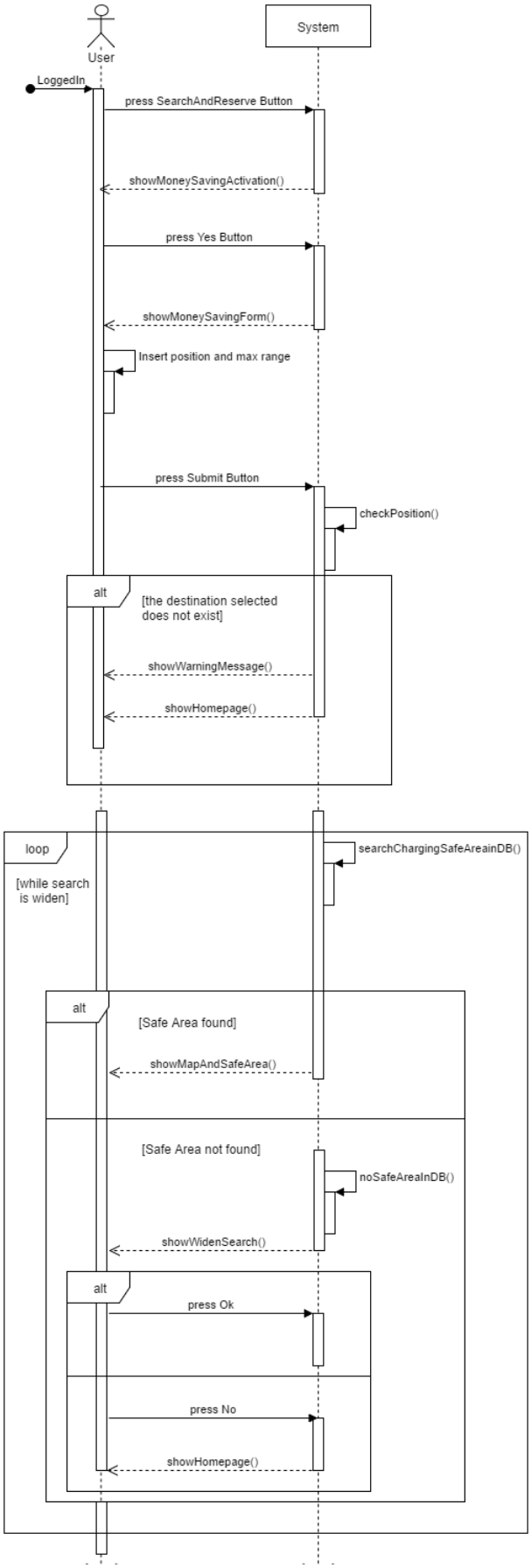


Figure 12: MoneySaving Option Sequence Diagram

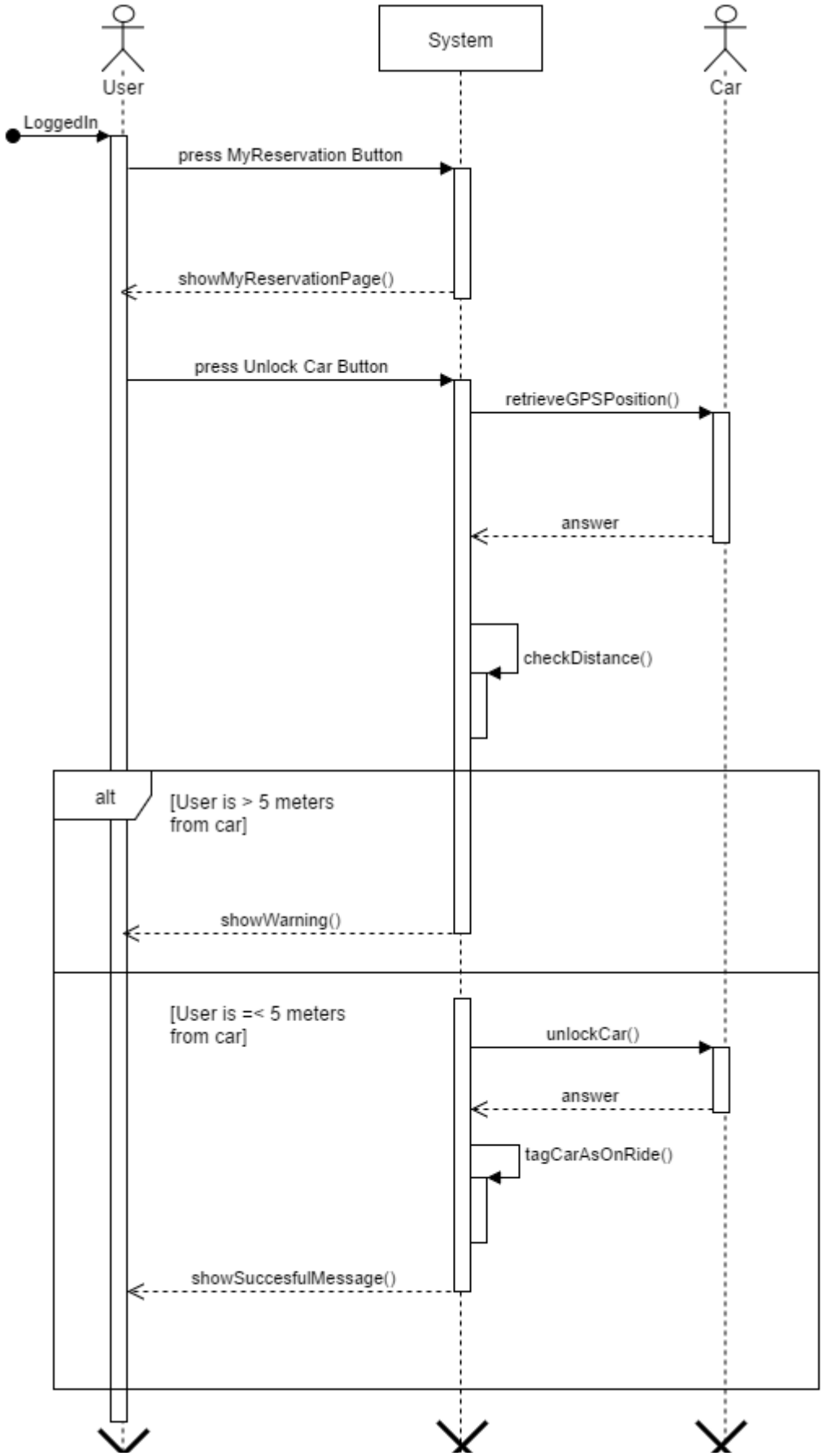


Figure 13: Unlock Car Sequence Diagram

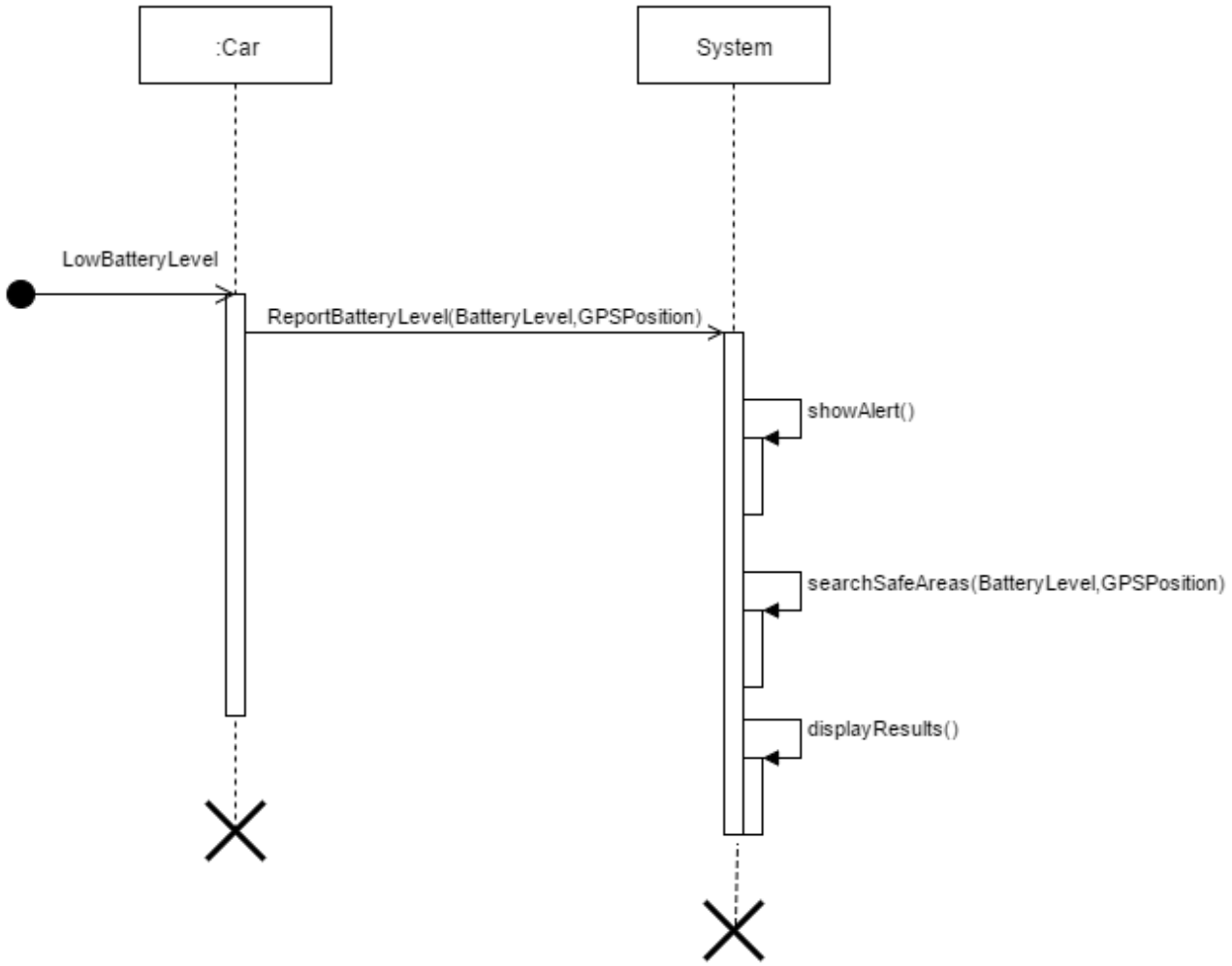


Figure 14: LowBattery Alert Sequence Diagram

STATE DIAGRAMS

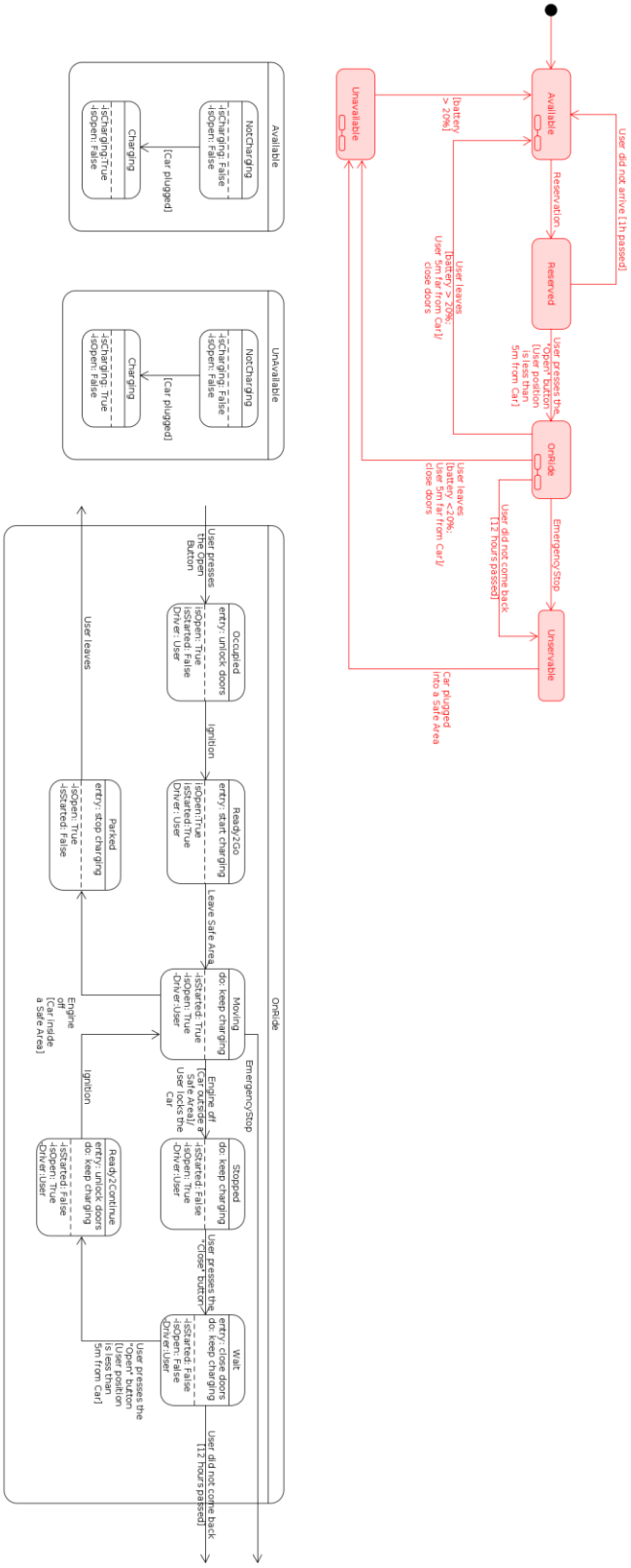


Figure 15: Car State Diagram

ACTIVITY DIAGRAMS

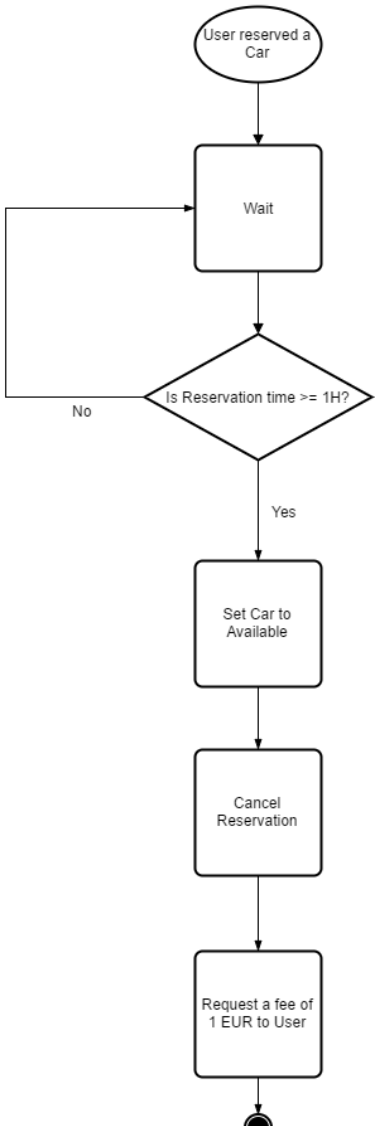


Figure 16: Reservation Expired Activity Diagram

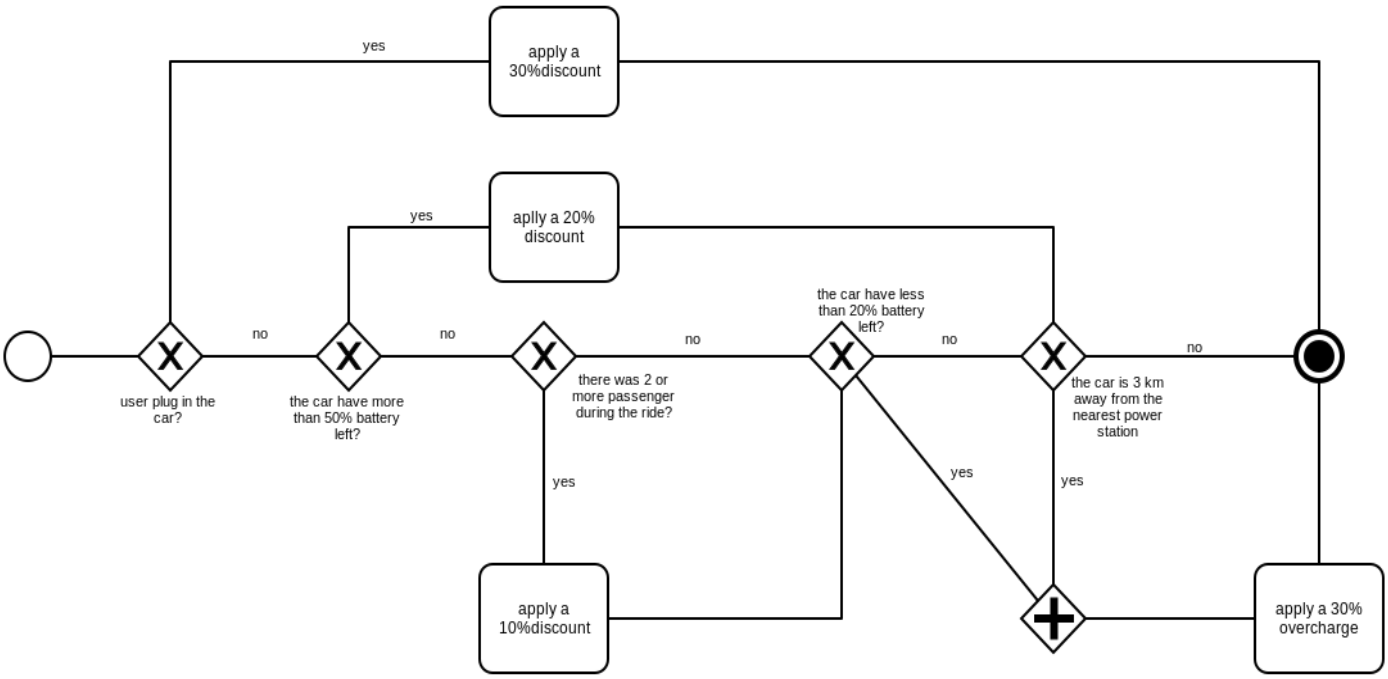
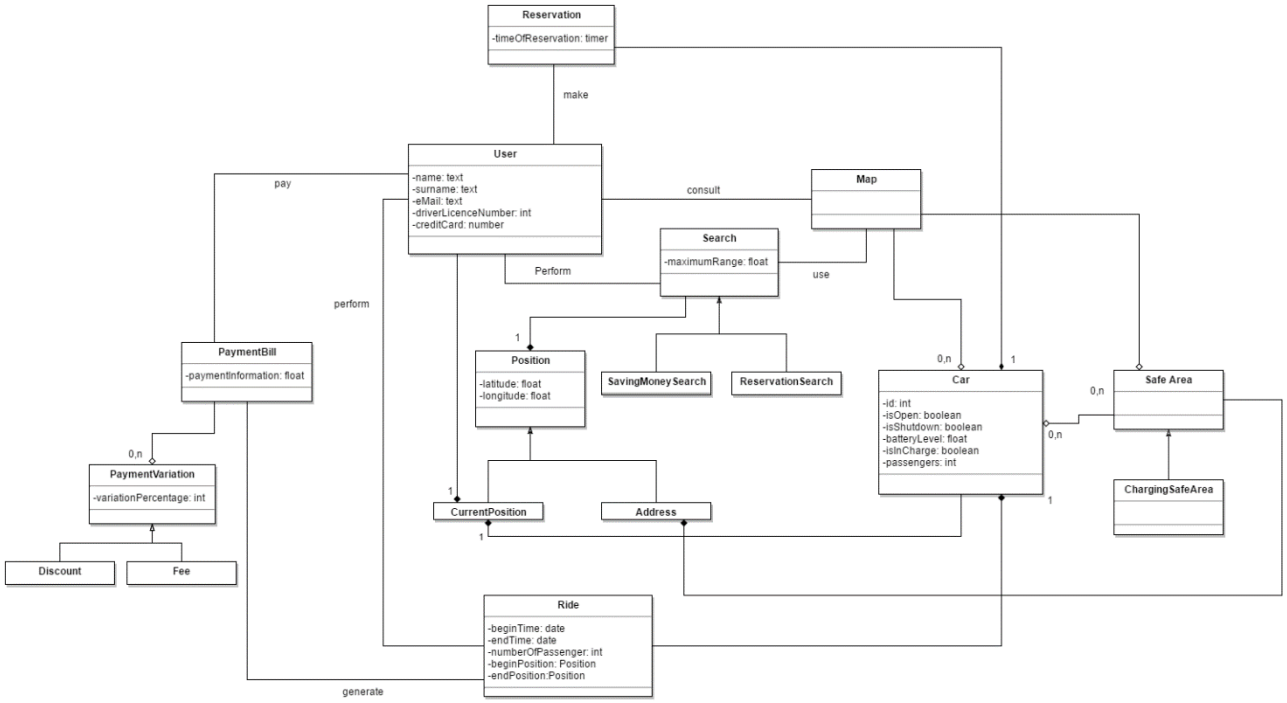


Figure 17: Payment Activity Diagram

CLASS DIAGRAM



ALLOY'S MODEL

```
module PowerEnjoy

// SIGNATURES

abstract sig Boolean {}
one sig True, False extends Boolean {}

sig Car {
	id: Int,
	driver: lone User,
	battery: BatteryLevel,
	passengers: Boolean, //Driver excluded; set to True for at most 2
passengers
	isOpen: Boolean,
	isCharging: Boolean,
	isStarted: Boolean,
	status: CarStatus,
	carPosition: Position
}{id>0}

abstract sig CarStatus {}

one sig Available,NotAvailable,Reserved,OnRide extends CarStatus{}

abstract sig BatteryLevel{}

one sig HighBattery,MediumBattery,LowBattery extends BatteryLevel{}

sig Position{
	latitude: Int,
	longitude: Int
}

sig SafeArea{
position:  Position,
capacity: Int,
parkedCars: set Car,
isChargingStation: Boolean,
chargingCars: set Car
}{capacity > 0 && #parkedCars <= capacity}

sig Reservation{
time: Int,
reservator:  one User,
reservedCar: one  Car,
}

sig User{
id:Int,
position:  Position,
prenotationPos: lone Position,  // Position of the address where the User is
reserving to pick up the Car
```

```
MoneySavingDestination:lone Position, //destination that the User wants to
reach when using the MoneySaving option
MoneySavingSafeArea: lone SafeArea //SafeArea where the System
suggests the User to leave the car
}{id>0}

//Relation of promixity between positions
one sig NearRel{
        near:Position->Position
}

//Discounts and overcharge appliable to the User
sig UserRideCharge{
        discount30,discount20,discount10: Boolean,
        overcharge30:Boolean,
        payer:User
}

//FACTS

//Car's facts

//Cars plugged to a power grid cannot be started
fact InChargeNoStart {
        all c:Car | c.isCharging = True => c.isStarted = False}

//Cars with a battery level below a certain level of power are either OnRide
with a User nor not available for reservations
fact lowBattery {
        all c:Car | c.battery=LowBattery => c.status = NotAvailable ||
c.status = OnRide
}

//Cars without a Driver are locked
fact LockedCar{
        all c:Car | c.driver=none => c.isOpen = False
}

fact OnlyOnRideEngineOn{
        all c:Car| c.isStarted=True => c.status=OnRide
}

fact StartedCarsAreOpen{
        all c:Car | c.isStarted = True => c.isOpen = True
}

//a Car is parked in a Safe Area when it's engine is off and its position
coincides with the one of the SafeArea
fact parkedCarDefinition{
        all c:Car,s:SafeArea |  c in s.parkedCars <=> (c.isStarted = False
&&  c.carPosition = s.position )
}

//a Car charging in a Safe Area is Car parked in that Safe Area too
fact chargingCarsDefinition {
```

```
all s:SafeArea, p:Car | p.isCharging = True && p in s.parkedCars <=> p in
s.chargingCars
}

fact AvailableCarsAreParked{
	all c:Car|c.status=Available => c in SafeArea.parkedCars
}

fact ReservedCarsAreParked{
	all c:Car| c.status=Reserved => c in SafeArea.parkedCars
}

// Reservations facts

fact ReservedCarAreInOnlyOneReservation {
	all c:Car |  c.status = Reserved  <=> (one r:Reservation | c in
r.reservedCar )
}

fact MaxOneReservationPerUser{
	all u:User | u in Reservation.reservator => no disj r,r':Reservation |
r.reservator = r'.reservator
}

fact OnlyReservedCarsInReservation {
	all c:Car | c.status = Reserved <=> c in Reservation.reservedCar
}

fact DriversAreOnTheCarWhenParking{
	all c:Car| c in SafeArea.parkedCars => (c.status=OnRide <=>
c.driver.position=c.carPosition)
}

fact ParkedCarStillOpenWithDriverNearTo{
	all c:Car | c in SafeArea.parkedCars && c.status=OnRide =>
c.isOpen = True
}

fact OnlyOnRideCarsHaveDriver{
		all c:Car| c.driver != none <=> c.status=OnRide
}

fact OnlyOnRideHavePassenger{
	all c:Car|  c.status!=OnRide => c.passengers=False
}

fact aDriverHaveOnlyOneCar{
	all disj c,c':Car |  c.status=OnRide && c'.status=OnRide => c.driver
!= c'.driver
}

fact NoReservationWhileDriving{
	all c:Car | c.driver != none => not (c.driver in Reservation.reservator)
}
```

```
//Users reserve Cars in a Position different from the one of the Car's
themselves
fact ReservedCarUserNotNear{
	all c:Car,u:User,r:Reservation| r.reservator=u&&r.reservedCar=c =>
c.carPosition!=u.position
}

//Available Cars are within a range from the Position inserted by the User
fact AvailableCarsNearReservationPosition{
	all r:Reservation|r.reservedCar.carPosition in
search[r.reservator.prenotationPos]
}

//Users have a prenotation Position if and only if they have an active
Reservation
fact onlyCarReservedHaveReservedPosition{
		all u:User|u.prenotationPos=none <=>u not in
Reservation.reservator
}

//SafeAreas facts

fact noCharging {
	all s:SafeArea | s.isChargingStation = False => s.chargingCars =
none
}

//two Safe Areas cannot have the same position
fact twoSANoSamePosition{
	all s,s1:SafeArea | s != s1 => s.position != s1.position
}

// Search facts

//Users cannot have a MoneySavingSafeArea too far from the destination
Position they have selected
fact MoneySavingDestination{
	all u:User| u.MoneySavingDestination!=none
<=>u.MoneySavingSafeArea!=none &&	u.MoneySavingSafeArea in
MoneySavingAreasAvailable[u.MoneySavingDestination]
}

//Users have tagged a MoneySavingArea from the moment they reserve a
Car to the moment they leave it
fact DestinationOnlyForResAndOnRide{
		all u:User|	u.MoneySavingSafeArea != none =>u  in
Car.driver+Reservation.reservator
}

//Charge's facts

fact MaxOneDiscount{
	all p:UserRideCharge| (p.discount30=True =>
p.discount20=False&&p.discount10=False) &&
	p.discount20=True=>p.discount10=False
```

```
}

fact NoOverchargeWithDiscount30{
	all p:UserRideCharge| p.discount30=True=>p.overcharge30=False
}

//Position and proximity facts

fact twoPositionHaveDifferentCoor {
	all p,p':Position | p!=p' => p.latitude != p'.latitude || p.longitude !=
p'.longitude
}

fact Riflex{
	all p:Position| p in NearRel.near.p
}
fact Sim{
	NearRel.near=~(NearRel.near)
}

//PREDICATES

//Predicates on Car's status changes

pred NotAvailableToAvailable[c,c':Car]{
		c.status=NotAvailable && c.battery=LowBattery
&&c.isCharging=True
		c'.status=Available && c'.battery=MediumBattery&&
c'.isCharging=True&& c.carPosition=c'.carPosition && c.id=c'.id
}

pred AvailableToReserved[c,c':Car,u:User,r:Reservation] {
	//precondition
	c.status=Available
	//postcondition
	 c'.carPosition=c.carPosition&& c'.status = Reserved &&
r.reservedCar=c'&& r.reservator=u&& c'.isCharging=c.isCharging
	&& c.battery=c'.battery &&c.id=c'.id
}

pred ReservedToOnRide[ c,c':Car,u,u':User,r:Reservation]{
	//precondition
	r.reservedCar=c && r.reservator=u
	//postcondition
	c'.carPosition=c.carPosition &&	c'.status=OnRide &&
c'.driver=u' && c'.isOpen=True && c'.isCharging = c.isCharging &&
c'.isStarted = c.isStarted
	&& c.battery=c'.battery &&c.id=c'.id && u.id=u'.id &&
	u'.position in NearRel.near.(u.position)
}

pred EndOfRide [c,c':Car,u,u':User,p:UserRideCharge] {
//precondition
	c in SafeArea.parkedCars && c.status=OnRide && c.driver= u
//postcondition
```

```
        c'.carPosition = c.carPosition && c'.status != OnRide  &&
u'.position!=u.position &&
        u' not in Reservation.reservator+Car.driver && c.battery=c'.battery
&&c.id=c'.id && u.id=u'.id &&c.isCharging=c'.isCharging
        c'.battery!=LowBattery=>c'.status=Available
        p=createPayment[c,u']
}

//FUNCTIONS

//creates the UserRideCharge instance associated with the end of a Ride
fun createPayment[c:Car,u:User]:one UserRideCharge{
        {p:UserRideCharge|p.payer=u &&
                (c.isCharging=True<=>p.discount30=True) &&
                (c.battery=HighBattery &&
p.discount30=False<=>p.discount20=True )&&
                (c.passengers=True && p.discount20=False&&
p.discount30=False <=>p.discount10=True) &&
                (p.discount30=False&& ((no s:SafeArea|
s.isChargingStation=True&& s.position in
c.carPosition)||c.battery=LowBattery)  <=>p.overcharge30=True)
        }
}

//found the Positions near the selected Address (Position)
fun search [p: Position] : set Position {
        NearRel.near.p-p
}

//lists the available SafeAreas with power grids near the selected Position
fun MoneySavingAreasAvailable[p:Position]:set SafeArea{
        {s:SafeArea| s.position in search[p] && s.isChargingStation=True
&& #s.parkedCars<s.capacity}
}
```

```
Executing "Run show"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   5466 vars. 498 primary vars. 11775 clauses. 44ms.
   Instance found. Predicate is consistent. 28ms.

Executing "Run AvailableToReserved"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   5885 vars. 510 primary vars. 12810 clauses. 37ms.
   Instance found. Predicate is consistent. 32ms.

Executing "Run ReservedToOnRide"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   6147 vars. 513 primary vars. 13497 clauses. 30ms.
   Instance found. Predicate is consistent. 27ms.

Executing "Run NotAvailableToAvailable"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   5793 vars. 504 primary vars. 12564 clauses. 33ms.
   Instance found. Predicate is consistent. 33ms.

Executing "Run EndOfRide"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   6216 vars. 513 primary vars. 13832 clauses. 33ms.
   Instance found. Predicate is consistent. 24ms.
```

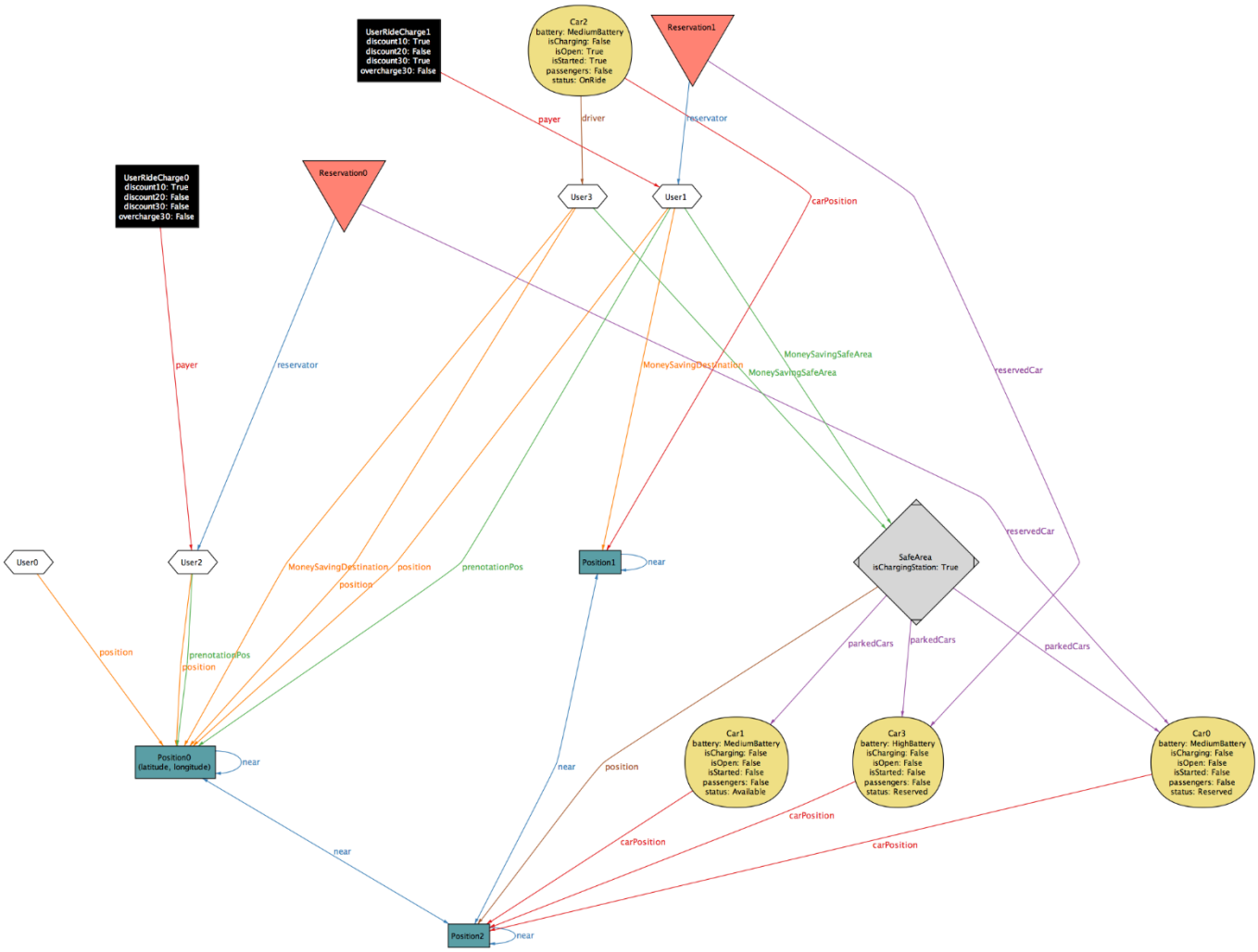


Figure19: A snapshot of the global Model generated by Alloy, through the predicate Show

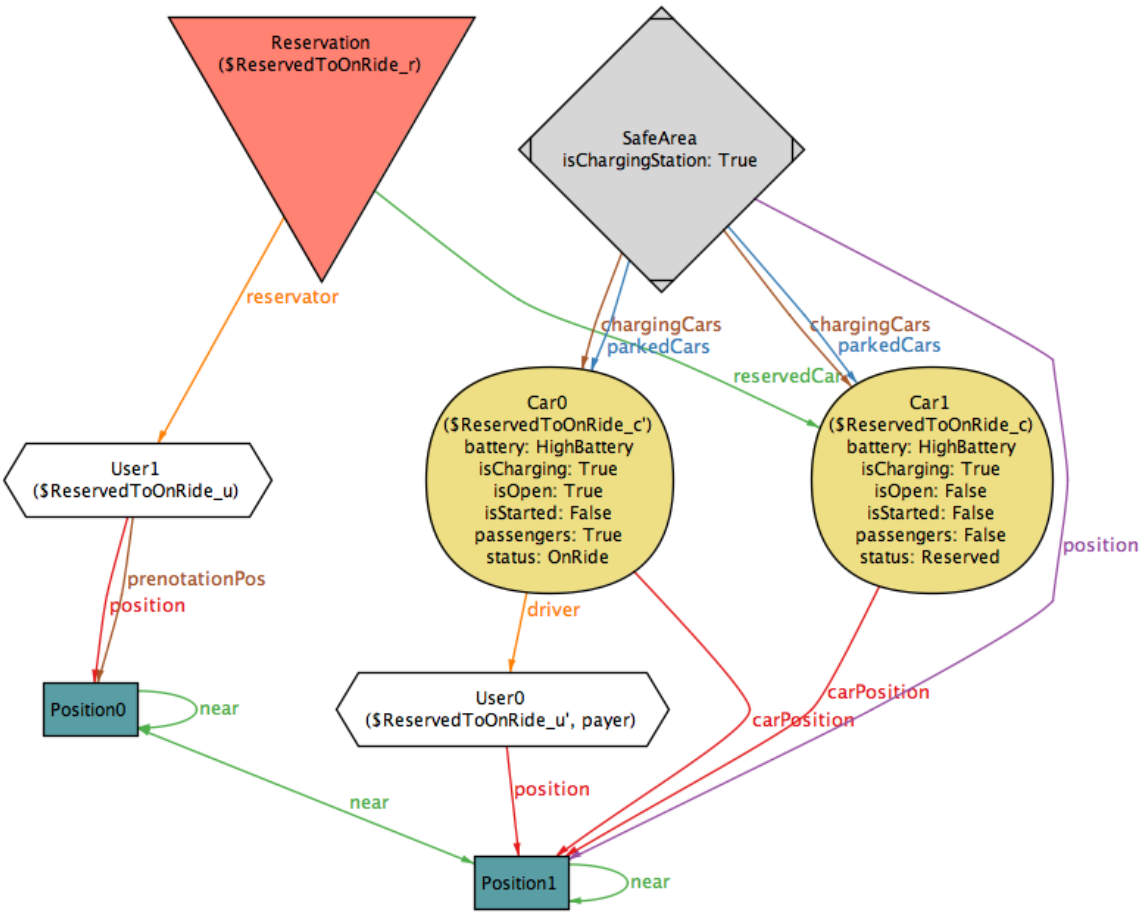


Figure20: A snapshot of the Model generated by Alloy, through the predicate ReservedToOnRide. This predicate represents the change of a car's status from Reserved to OnRide, when a user access it.

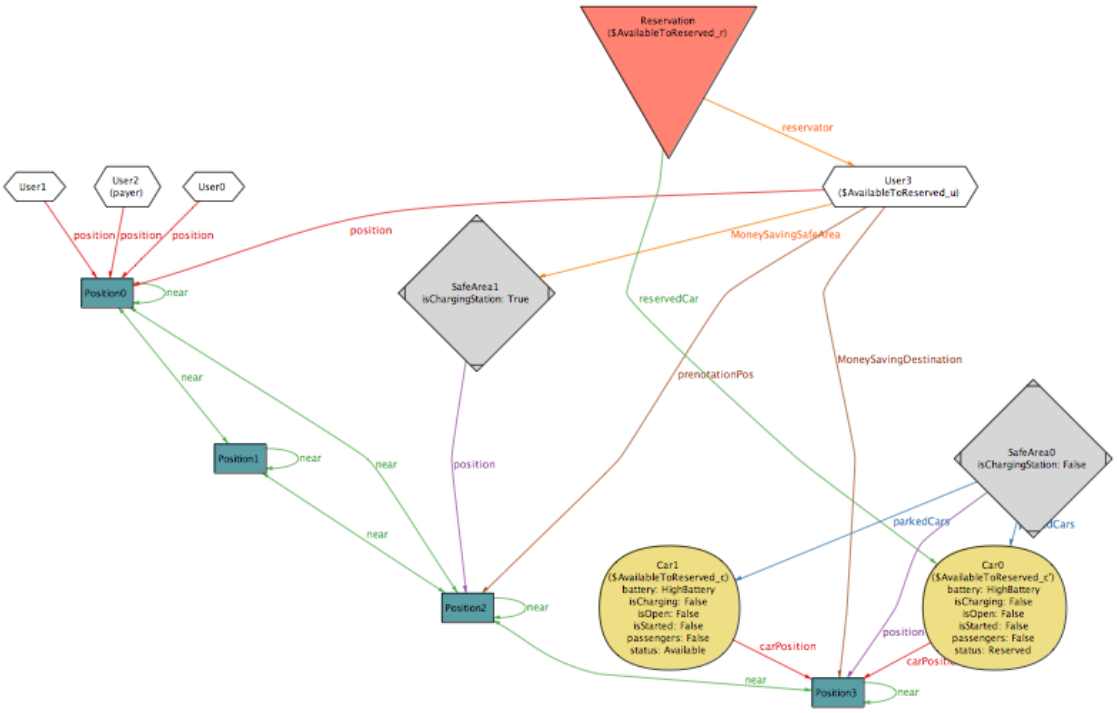


Figure21: A snapshot of the Model generated by Alloy, through the predicate Available to Reserve. This predicate represents the change of a car's status from Available to Reserve, after its reservation is complete.

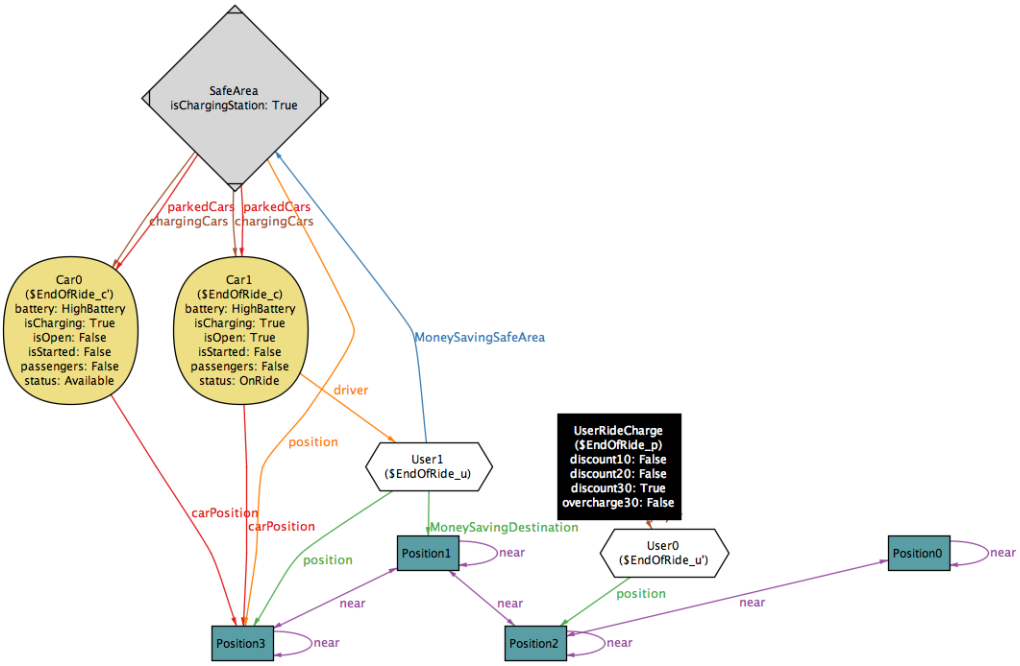


Figure22: A snapshot of the Model generated by Alloy, through the predicate EndOfRide. This predicate represents the change of a car's status at the end of the ride of a car. The istance of UserRideCharge is attached to the new istance of User u'.

FUTURE DEVELOPMENTS

After the deployment of the app, future features will be added. Here are some of them:

- Cancelling Reservation.
- Navigation to Safe Area.
- Possibility for a User to recharge a Car in a Safe Area while continuing to reserve it (Pit-Stop Feature).

HOUR OF WORK

	Cannas	Castiglioni	Loiacono
20.10	1.30		
21.10	5.30		
23.10	1	1	1.45
25.10	2.30	2	2.30
26.10	1.15	0	1.15
29.10	3.30	3	
30.10	3	2	4.20
31.10	5	4	5
1.11	4	2	3
3.11	4		
4.11	1.30	3	4
5.11	0	1	4
6.11	6	4	5
7.11	1.30		
8.11	2.30	2.30	3
9.11	3.45	2	0
10.11	4	4	1
SUM	45	43	46