Power EnJoy

Emanuele Ghelfi
Emiliano Gagliardi

November 1, 2016

Contents

1 Introduction

1.1 Description of the given problem

The purpose of this document is to support the developement of a software that manages a car sharing services. The software that will be developed has the aim to make the reservation of cars simple and qick. So the system should provide users with real time information about avaiability of cars, and their positions. Cars will be parked in some pre-defined parking area, where user can directly get them afther the reservation. The service will be provide only to user that have made the registration, giving some personal information and data needed to the payment. The price of the ride is computed with a fixed amount of minutes, displayed by the car, and finally charged. To avoid useless reservation, where a user doesn't pick up the car, the system should be able to detect that after a fixed time, retag the car avaiable, and charge with a fee the user. Cars must be loked in the safe areas, and unlokable only by users that has made a reservation on them.

1.2 Glossary and abbreviations

- VIRTUOUS BEHAVIOUR:
  - Behaviour thate makes simple the maintance of the cars and reservation areas.
  - Behaviour helps to reduce the traffic impact.
  - To park in defined safe areas.

1.3 Actors

These are the people that will be involved in the use cases of the software to be developed:

- GUEST: a person that hasn't made the registration to the service.
- USER: a person that has already registered to the system.
- OPERATOR: an employee of the company that takes maintenance of cars and charging area.

1.4 Assumptions

There are few point that aren't well specified int he assignement document. This section's purpose is to formalize some facts that complete the specification.

- Also GUESTs are able to search cars, in order to induce them to create an account.

- There are no information about cars' maintenance, so in the document is made the assumption that cars need OPERATOR presence to keep an acceptable status.
- USERs can't delete reservations.
- The system relies on external payment methods. The accepted methods are Mastercard, Visa, PayPal, PostePay.
- If a payment fails for a insufficient money availabiity of the user, the USER is suspended from the service until the payment is done.
- In order to induce USER to park in safe areas, USER that does not park in safe areas will pay a fee.

1.5 Domain assumptions

- [D1] Cars positions provided by GPS are accurate,
- [D2] The USER and GUEST position provided by GPS are accurate.
- [D3] Cars provide remote comunication API
- [D4] Cars provide screen to display current charge to user.
- [D5] Cars can detect how many passengers are on it.
- [D6] Cars can detect if they are plugged.
- [D7] Cars can detect their battery level.
- [D8] Cars can detect information about maintenance needs.
- [D9] If a USER detect a malfuncion comunicate it.
- [D10] If an OPERATOR is notified by the system about a maintenance need, he take care of that.
- /*se un utente apre sale, va dove vuole andare e non si schianta*/
- /*non danno in giro la password*/

1.6 Goals

The objectives of the software to be developed are the following:

- [G1] USERs must be able to reserve a car and use it.
  - [G1.1] USER should be able to reserve a car.
  - [G1.2] USER should be able to unlock reseved car when they are close to it.

  - [G1.3] USER should be aware of how much they are going to pay during the ride.

- [G2] The reservation of two (or more) cars at a time must be forbidden.
- [G3] USER and GUEST must be able to search cars.
  - [G3.1] USER should be able to search car near his position.
  - [G3.2] USER should be able to search car near a selected position.
- [G4] Induce USER to keep a VIRTUOUS BEHAVIOUR.
  - [G4.1] A discount of 10% should be applied to rides with at least two passengers.
  - [G4.2] If a car is left with no more than 50% battery empty, a discount of 20% should be applied to the last ride.
  - [G4.3] If a car is left at more than 3 km from the nearest charging area, the system should charges 30% more on the last ride.
  - [G4.4] If a car is left with more than 80% battery empty, the system should charges 30% more on the last ride.
  - [G4.5] If a car is left in a charging area and the user plugs the car into the power grid, a discount of 30% should be applied to the last ride.
  - [G4.6] If a car is not picked up within one hour from the reservation, the USER should a fee of 1 EUR.
  - [G4.7] If a payment fails for a insufficient money availabiity of the USER, the USER should be suspended from the service until the payment is done.
  - [G4.8] If a USER parks in an area that is not a safe area, the USER should pay a fee.
- [G5] USER have to pay an amount of money based on the ride's duration.
- [G6] Guarantee a ready maintenance of cars.
  - [G6.1] OPERATOR must be able to do car mantainance, knowing their position and status.
  - [G6.2] USER must be able to comunicate to the system cars damages and malfunctions.

1.7 Document overview

/*here we will describe the document contents organization*/

2 Overall description

2.1 Product perspective

In this section the system boundaries are described, like the interfaces porvided by external systems that the software to be developed will use in order to provide the required functionalities.

2.1.1 Payment methods

In order to manage in a good way the payment process, and to ensure a well managing of exceptional situations like users unavailability of money USERs, the software to be developed relies on external transactional system. So the system doesn't care about how the payments are done, but only that the USER has provided one of the payment methods defined in the assuption section.

2.1.2 Car technology

The software to be developed will use the tecnologies provided by a system already present on cars, in order to carry out the needed actions. The used API provides a secure connection between cars and the central system via internet, and the following commands that the central system can impose on car:

- Lock: locks the car doors.
- Unlock: unlocks the car doors.
- Get position: the car sends to the central system its position.
- Get status: the car sends to the central system detailed information about his current status.

2.1.3 Operators, maintenance, malfunctions and damanges.

The cars availability depends heavily on the presence of a staff that does cars maintenance. So the sotware to be developed relies on the presence of the staff in order to guarantees availability of cars, and will provide to operators informations that helps the cars maintenance. Operators will manage damanges to cars and consequence for the responsible, so the system will only help the OPERATORs in maintenance, and guarantee a way to comunicate malfunction and damanges.

3 Requirements

In order to satisfy goals in section [Section Goals] under domain assumptions in Section [Section domain Assumption] we've derived requirements for our system.

- [G1.1] USER should be able to reserve a car:
  - [R1] The system can modify car's status (available, occupied, in use).
  - [R2] The system shall be able to know car's status.
  - [R2] The system shall reserve a car only if the car is available.
  - [R3] The system shall be able to associate a car to the USER who has reserved it.
- [G1.2] USER should be able to unlock reseved car when they are close to it.
  - [R4] The system shall know cars' location according to cars' GPS.
  - [R5] The system shall know user's location according to his GPS.
  - [R6] The system shall provide a functionality to unlock the car.
  - [D1]
  - [D2]
  - [D3]
- [G1.3] USER should be aware of how much they are going to pay during the ride.
  - [R7] The system shall be able to calculate ride's cost based on duration.
  - [R8] The system shall be able to communicate to the car ride's cost.
  - [D4]
- [G2] The reservation of two (or more) cars at a time must be forbidden.
  - [R7] The system shall reserve a car only if the user hasn't already reserved another car.
  - [R3]
- [G3.1] USER or GUEST should be able to search car near his position.
  - [R4]
  - [R5]
  - [R9] The system shall provide the closest cars to user's position.
- [G3.2] USER or GUEST should be able to search car near a selected position.

  - [R4]
  - [R5]
  - [R10] The system shall provide the closest cars to position provided by the USER.

- [G4.1] A discount of 10% should be applied to rides with at least two passengers.
  - [R11] The system shall bel able to detect if in there are at least two passengers in the car.
  - [D5]
  - [R12] The system shall be able to apply a discount on the ride.

- [G4.2] If a car is left with no more than 50% battery empty, a discount of 20% should be applied to the last ride.
  - [R13] The system shall be able to know the battery level of the car.
  - [R12]
  - [D7]

- [G4.3] If a car is left at more than 3 km from the nearest charging area, the system should charges 30% more on the last ride.
  - [R4]
  - [R13]
  - [R14] The system shall be able to apply a charge on the ride.
  - [R15] The system shall be able to calculate distance between car's position and the nearest charging area.

- [G4.4] If a car is left with more than 80% battery empty, the system should charges 30% more on the last ride.
  - [R13]
  - [R14]

- [G4.5] If a car is left in a charging area and the user plugs the car into the power grid, a discount of 30% should be applied to the last ride.
  - [R15] The system shall be able to detect if the user has plugged the car into the power grid.
  - [R12]
  - [R4]
  - [D6]

- [G4.6] If a car is not picked up within one hour from the reservation, the user pays a fee of 1 EUR.

  - [R16] The system shall be able to monitor car's reservations.
  - [R17] The system shall be able to know if a user has picked up a car.
  - [R2]

- [G4.7] If a payment fails for a insufficient money availability of the USER, the USER should be suspended from the service until the payment is done.
  - [R18] The system shall take care of the result of the payment.
  - [R19] The system shall suspend USER that can't afford the payment.
  - [R20] The system shall unsuspend a USER when his old undone payment is done.

- [G5] USER have to pay an amount of money based on the ride's duration.
  - [R7]
  - [R21] The system shall charge an extenernal payment service with doing the payment.
  - [R9]
  - [D4]

- [G6] Guarantee a ready maintenance of cars.
  - [G6.1]
  - [G6.2]
  - [D9]
  - [D10]

- [G6.1] OPERATOR must be able to do car mantainance, knowing their position and status.
  - [R22] The system shall offer a backend panel to Operator
  - [R23] The system shall notify the operator when a car in his competence area needs some kind of maintenance.
  - [D7]
  - [D8]
  - [R1]
  - [R2]

- [G6.2] USER must be able to comunicate to the system cars damages and malfunctions.
  - [R24] The system shall offer a functionality that let USERs to comunicate manfunctions and damages.

3.1 Functional Requirements

After having defined main features of our system we can identify some functional requirements grouped under each defined actor:

- GUEST, he can:
  - Sign up.
  - Search for a car near his location.
  - Search for a car near a selected location.
- USER, he can:
  - Log in.
  - Reserve car.
  - Use a reserved car (unlocking it).
  - View his profile (also his discounts).
  - Modify his profile.
  - View his recent reservations.
  - Search car near to his location.
  - Search car near a selected location.
  - Comunicate malfunction or damages to the system.
- OPERATOR, he can:
  - View car's status (battery and location).
  - Receive notification when a car need to be charged.
  - Set a car as available or under maintainance.

3.2 Non-functional Requirements

3.2.1 User Interface

In order to make PowerEnJoy available for many people as possible a Cross-Platform application has to be developed, with a Framework like Xamarin. In this way could be developed a single application and then could be deployed for Windows Phone, Android and iOS. A Web Application has also to be developed in order to allow people to use our system in many way as possible.
The User Interface needs to be as user friendly as possible, a user that uses for the first time Power EnJoy should be able to learn in a few seconds how to make use of it.
Here are presented a few screen of the app to provide an example.

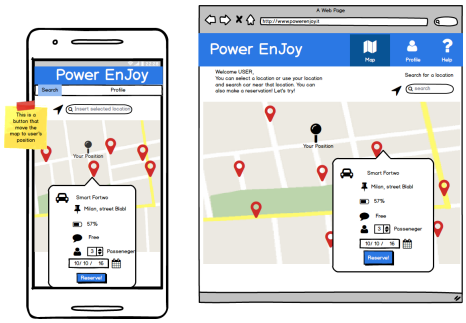

Figure 1: Map UI

3.2.2 Security

The system has to satisfy an high level of security regarding the communication protocol with the cars.
The communication with the cars needs to be encrypted in order to avoid unauthorized unlocks and use. Futhermore the system should have the way to stop cars when used in an unauthorized way.
Regarding the security of payment, the system does't care about it because it relies on external payment system.
Another critical security point is the managing of user's information: all user informations need to be encrypted before being sent.

3.2.3 Availability

The application needs to be online 24h/day and 7day/week in order to satisfy all users. This is a constraints because an hour of unavailablity could induce users to use another car sharing service. To achieve an high level of availability could be necessary to use a dedicated server and all the system could be hosted in a cloud platform like OVH. The system should be able to support a high number of users connected at the same time and could be necessary to scale resources depending on this.

3.2.4 Portability

The client-part of the system (application) could be used on any mobile OS in order to make it available for as many people as possible. It is expected a Web Application to increase the possible number of users.
The server-part of the system could be used on any OS which supports JVM and DBMS.

3.2.5 Maintainability

The system code will be documented in order to make easier future improvements or changes by other developers. It needs to be clear how the system works, how the system communicate with cars, and how the system has been developed.

3.2.6 Documentation

These documents will be released in order to well-organize the work in the way to obtain the best quality-cost ratio as possible:

- ***RASD***: Requirements Analysis and Specification Document, to well-understand the given problem and to analyze system's boundaries and goal. RASD is also important to understan system's requirements and specifications in order to reach the goals.
- ***DD*** :Design Document, to define the structure of the the system, his tiers, and the interaction between them.
- ***ITPD***: Integration Test Plan Document, to describe the planning to accomplish the integration test. This document is supposed to be written before the integration test really happens. Moreover this document needs to explain to the developer team what to test, in which sequence, which tools are needed for testing and wich need to be devloped.
- ***PP***: Project Plan, aims at defining a planning for the project. It regard in particular the estimation of effort and cost, the scheduling for project's task and the allocation of resources to tasks.

4 Scenario Identifyng

4.1 Scenario 1

Paul, a businessman, has to go to Milan by train in order to participate to an important meeting.
As the office where they have to meet is 5 km far (away?) from the railway station and taxis are too much expensive, he decides to use *PowerEnJoy* from his smartphone. He notices, thanks to the map on the app, that there is a parking area next to the railway station with a lot of available cars. He then reserves one of these electric cars so that, as soon as he arrives there, he'll be able to reach the office very quickly and in a green way.

4.2 Scenario 2

Paul is a student that goes to Politecnico University in Milan. He live in a city near Milan but unfortunately train's schedule is very prohibitive: he should wake up at 5 o'clock in the morning in order to arrive on time at the lecture that starts at 8:00.
He found out about *PowerEnJoy* from two of his friends in his city that attend the same University. They decide to use together car-sharing service so they can save money using the "two passegners" discount.
Since their city is full of electric-car, everyday they make a reservation of a car and go to the University together.
In this way, not only they save money, but they are also much more comfortable in reaching the University.

4.3 Scenario 3

Paul has just learnt about *PowerEnJoy*.
This evening he has to go to dinner with his girlfriend, Sarah, so he decides to try it. Then he downloads the app and looks for cars near his location. He finds many cars, he's about to reserve one of them but reservation is not permitted to guests.
For this reason he has to sign up: following the application instructions he provides his personal informations, his driving license and he specifies his payment method.
After that he's able to reserve the car for special occasion.

4.4 Scenario 4 - OPERATOR maintenance

Jonathan is doing his periodical check of the car in his competence area. He finds out that a dishonest user has left a car with a broken light, without comunicate that. So Jonhatan, by his maintenance service system, makes the car unavailable and starts to solve the maintenance problem.

4.5 Scenario 5 - OPERATOR notification

Amy is driving an electrical car reserved with *PowerEnJoy*. As always she is very late today, and she notice that, if she want to arrive in time, she wont manage to bring the car at the station where it can be charged. So she decides to park the car as close as possible to her destination. After she has parked, the OPERATOR Jonathan, receives a notification on his maintenance service view, that tells where the car just left by Amy is, and that it is almost completely discharged. Now Jonathan can go take the car and make it again available for the next customer.

4.6 Scenario 6 - USER comunicates a malfunction to the system

Amy is driving an electrical car reserved with *PowerEnJoy*. Arrived at is destination, he start a parking maneuver, but she hurt a light pole, doing a damage to the machine. Amy is an honesty person, and following the application instruction she comunicates to the system that she has damaged the car. The system will comunicate to an OPERATOR about the damange.

5 UML Models

5.1 Use Case Diagram

With the Use Case Diagram are highlighted the main function of our system and the interaction between ACTORS and the related Use Case.

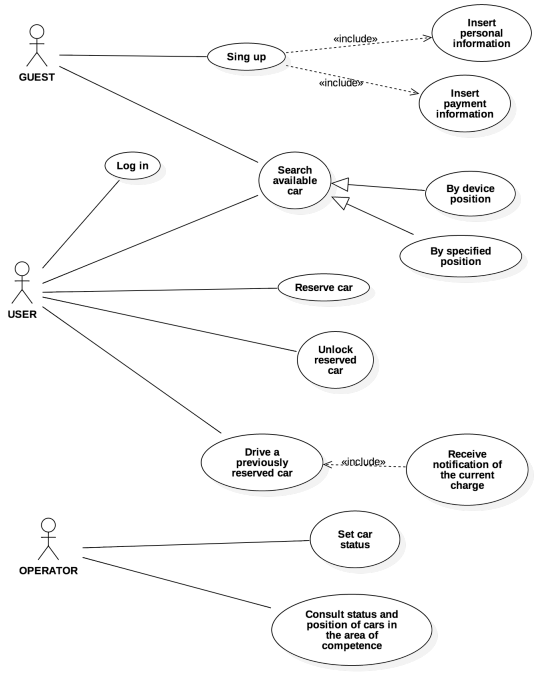

Figure 2: Use Case Diagram

5.2 Sequence Diagram

With Sequence Diagram are highlighted the main function of the system and the High-Level sequence of actions that focuses on the way in which happens the interaction between ACTORS and system.

5.2.1 Registration

Name	Registration
Actors	Guest
Goal	All
Entry **Conditions**	Guest isn't already registered to the application.
EventFlow	1. Guest opens the app. 2. Guest click on Sign Up button. 3. Guest fills in all mandatory fields like payment informations, his vital statistics etc. 4. Form is sent to the server that saves all data in the DB. 5. Email with registration confirmation is sent by the server to the email address provided in the form.
Output Condition	GUEST ends registration procedure and become a USER. Now he/she can login and use all functionalities offeredby the system.
Exception	1. Payment informations are not valid. 2. Driving License in not valid. 3. Username is already used by another user. 4. One or more mandatory fields of the form are not valid. In all this case the visitor is alert and the app explains the problem. Then the app goes back to point 3 of Event Flow.

5.2.2 Login

Name	Login
Actors	USER
Goal	All
EntryConditions	USER is registered into the system
EventFlow	1. User opens the app. 2. App shows login page. 3. User inserts his email and his password. 4. Email and password are sent to server in a secure way. 5. Server verifies user's credentials and confirms login.
OutputCondition	User can use the app, reserve cars and consult his profile.
Exception	1. User's credentials are incorrect. 2. User is banned to the system (due to an incorrect behaviour). In the first case the user is notified that his/her credentials are incorrect. In the second case the user is notified that he can't use the service because of his behaviour.

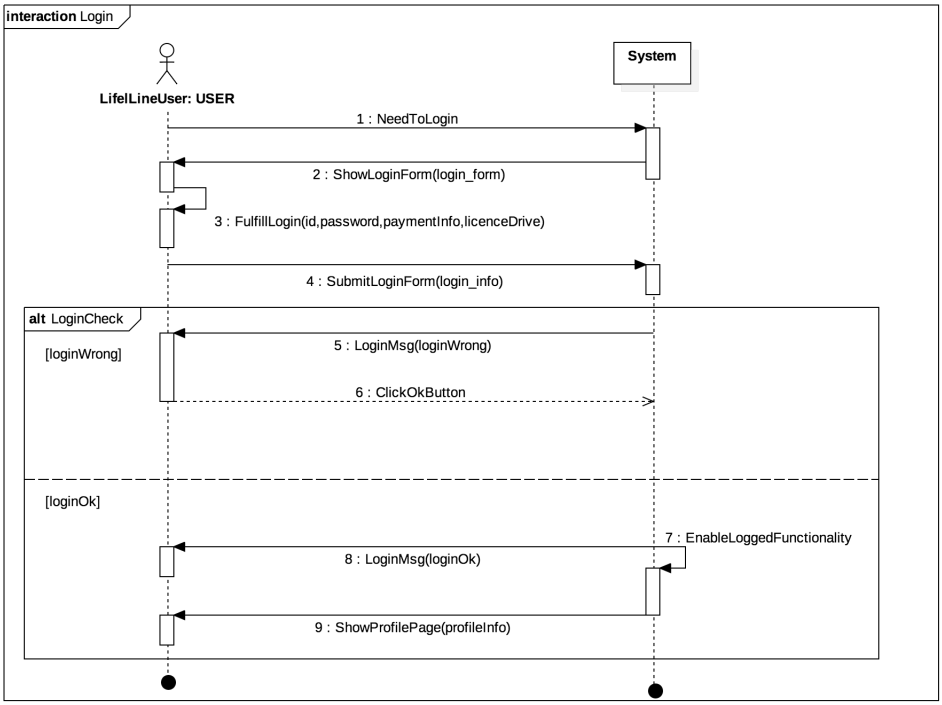

Figure 3: Login

5.2.3 Car Reservation

Name	Car Reservation
Actors	USER
Goal	[G1.1]
EntryConditions	USER is registered into the system and has already logged in.
EventFlow	1. USER selects a car. 2. User fills the reservation form with the reservation hour. 3. Form is sent to the server. 4. Server verifyies the form and the validity of reservation. 5. Server updates his informations with the new reservation. 6. App confirms reservation to the user. 7. App adds to current user's reservation the actual reservation.
OutputCondition	User has reserved a car and he can use it.
Exception	1. Car is already reserved by another user. 2. User has already reserved another car. In the first case app notifies the user that he needs to select another car in order to make his reservation. App returns to point 1 of Event Flow. In the second case appp notifies user that he has already reserved another car and that he can't reserve more than one car at the same time.

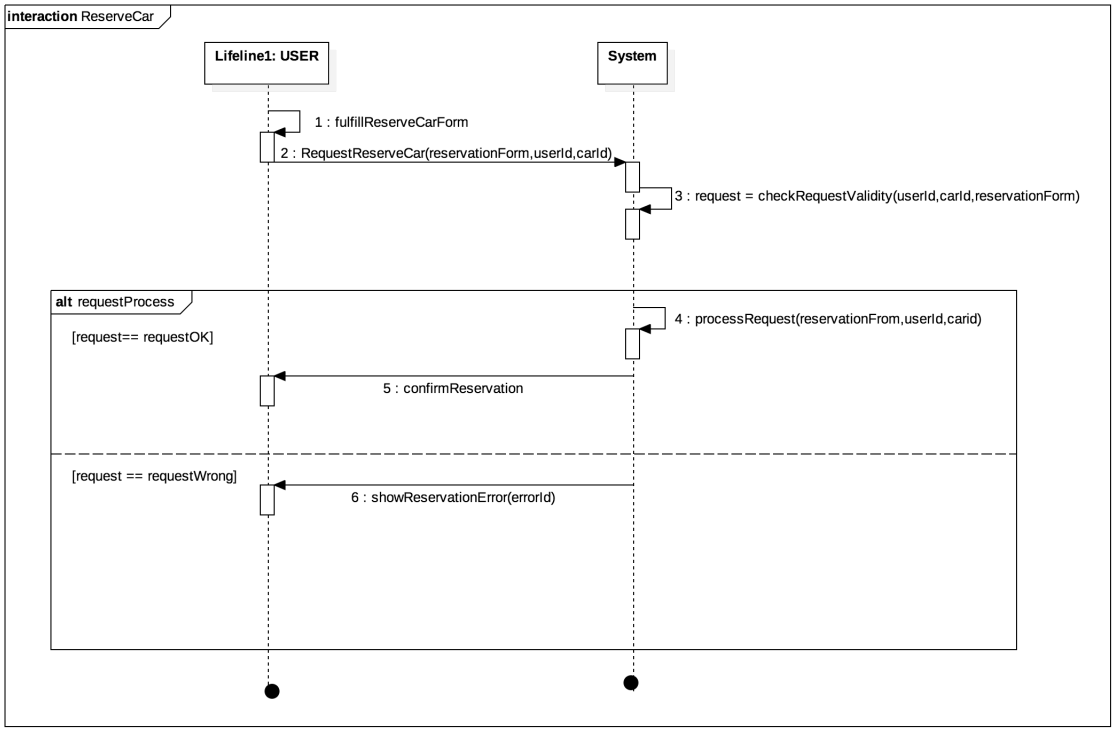

Figure 4: Car Reservation

5.2.4 Search Cars

Name	Search for Cars
Actors	USER or GUEST
Goal	[G3.1], [G3.2]
EntryConditions	NULL
EventFlow	1. User opens the app. 2. User select the search method: search car near his position or near a selected position. 3. //TODO 4. Form is sent to the server. 5. Server verifyies the form and the validity of reservation. 6. Server updates his informations with the new reservation. 7. App confirms reservation to the user. 8. App adds to current user's reservation the actual reservation.
OutputCondition	User has reserved a car and he can use it.
Exception	1. Car is already reserved by another user. 2. User has already reserved another car. In the first case app notifies the user that he needs to select another car in order to make his reservation. App returns to point 1 of Event Flow. In the second case appp notifies user that he has already reserved another car and that he can't reserve more than one car at the same time.

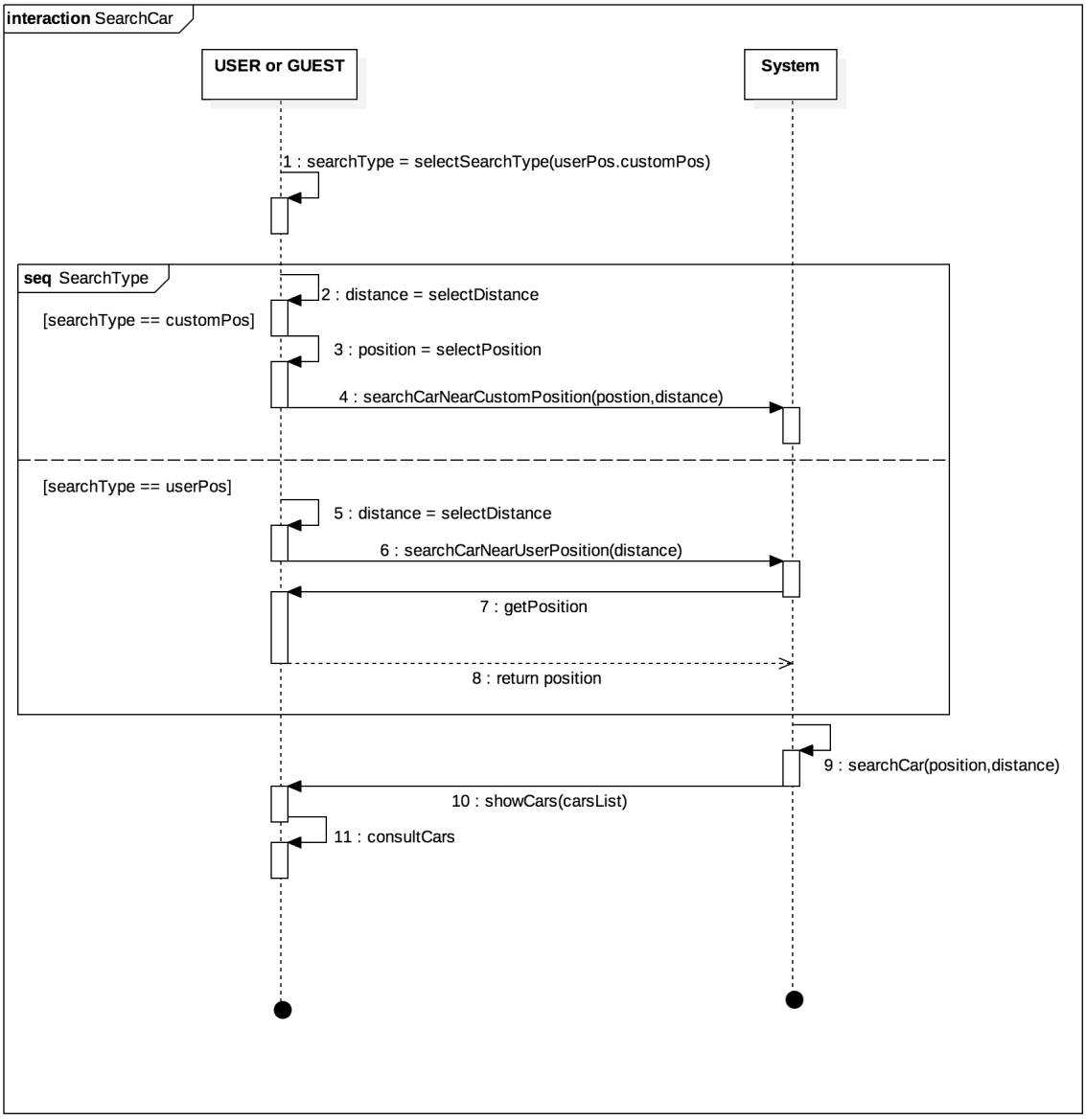

Figure 5: Search for a car

5.2.5 Unlock Car

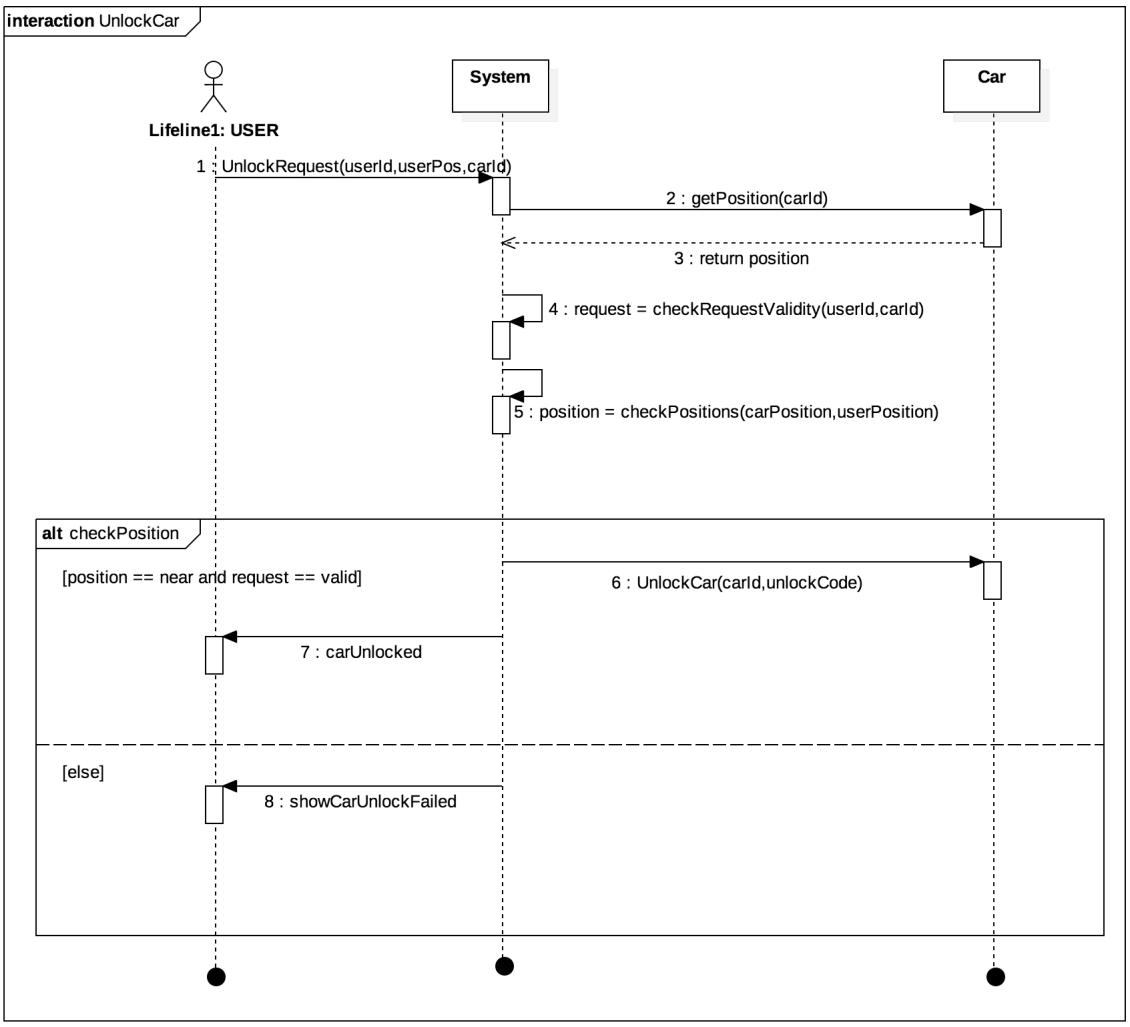

Figure 6: Unlock Car

5.2.6 Comunicate Damages And Malfucntions