RASD
Requirement Analysis and Specification Document

Luca Marzi
Valeria Mazzola
Federico Nigro

November 13, 2016

Contents

1 Introduction

1.1 (Hours dedicated to the project)

- Luca Marzi: 28 h
- Valeria Mazzola: 28 h
- Federico Nigro: 28 h

1.2 Problem analysis

We manage to design and develop PowerEnJoy that is a digital management system for car-sharing service that exclusively employs electric cars.
The service is to be provided to the users by means of a mobile application that is available for free both for iOS and Android based devices. The system will provide all the major functionalities that are provided by other concurrent car-sharing services such as EniEnjoy, Car2Go, Share'nGo etc.
The system will require the user to register an account that must be associated to his/her personal information that have legal validity: full name, date of birth, number of the identity card and number of the driving license. At the moment of the account registration the user will be asked to provide at least one payment method to be associated to the account. The system only supports one payment method that is PayPal.
The system will allow registered users to discover available cars nearby their current position. Registered users will be allowed to book a car for a limited

time (1 hour). Within this time the user will either decide to use the car for a ride or delete hit/her reservation; if the user doesn't use the car and doesn't delete his/her reservation, the reservation will expire, they will be charged a fee of 1 EUR and the reserved car will become available again for other users to reserve.
In addition to the main functionalities regarding booking and usage of the cars, the system will incentivize the virtuous behaviors of the users by means of applying discounts or fees in the appropriate contexts.

2 Goals

Goals expressed in terms of "the world and the machine":

2.1 [G1]: Registration:

[G1.1]: The User shall receive a password to access the services if and only if she/he has provided

- valid credentials that are not associated to any other User and
- valid payment information.

2.2 [G2]: Discovering available Cars and Car's informations:

[G2.1]: After providing a valid position, the Registered User shall be able to discover the positions of all available Cars within a certain distance from the specified position.

[G2.2]: The User shall be notified about the positions of other available Cars within a certain distance from specified position when those Cars become available.

[G2.3]: Multiple requests which consist of more than one User who wants to reserve the same Car, shall be correctly managed.

2.3 [G3]: Car booking:

[G3.1]: By providing the identity of a Car from the list of available ones, the Registered User shall be able to see all the current Car's relevant informations and the reservation form for the specific Car.

[G3.2]: By submitting informations required by the reservation form for a Car, the Registered User shall be able to book that Car and see the receipt of the reservation successfully if and only if:

- the Car is available and
- there's no other active registration made by the User for any other car.

2.4 [G4]: Reservation validity:

[G4.1]: After exactly one hour from when an active reservation was made the active reservation expires if and only if

- the active reservation hasn't been deleted by the User and
- the active reservation hasn't any ride associated to it.

[G4.2]: The Registered User shall be notified about the expiration of the reservation and shall pay the expiration fee if and only if

- her/his active reservation expires.

[G4.3]: When an active reservation expires the Car associated to that reservation shall be tagged as available for the other Users by the System.

[G4.4]: The Registered User shall be able to delete her/his active reservation if and only if

- the time elapsed from the beginning of reservetion is still less than one hour.

[G4.5]: When an active reservation is deleted, the Car associated to that reservation shall be tagged as available by the System.

2.5 [G5]: Car's usage:

[G5.1]: The User shall be able to notify the condition of nearness to the reserved Car if and only if

- the User owns an active reservation for the Car
- the GPS position of the User is within a certain distance from the reserved Car.

[G5.2]: A reserved Car shall be automatically unlocked if and only if

- the User who owns the active reservation on the Car notifies the condition of nearness to it.

[G5.3]: When the registered User notifies the condition of nearness to the Car, the active reservation shall be associated to a ride. The User shall be notified about the beginning of the new ride.

[G5.4]: When the User starts the engine of the Car, the ride associated to the active reservation for that Car shall be converted in an active ride. The User shall be notified about the beginning of a new active ride and the charging for a given amount of money per minute shall begin.

[G5.5]: The User shall be able to see the current active ride relevant information on the Car's screen.

[G5.6]: The User shall be notified about the end of the active ride and the charge towards the User shall stop if and only if:

- the Car is parked and
- the User and all the passengers exit the Car.

[G5.7]: At the end of an active ride, the System should charge the User for a parking penalty and notify the User about the charge if and only if:

- the Car is parked in a non safe area.

[G5.8]: The Car locks itself automatically if and only if

- it is parked and the User and all the passengers exit it.

[G5.9]: If the registered User uses the Car with at least two other passengers on board, there shall be applied a discount of 10% on the total cost of the current active ride at the end of it.

[G5.10]: At the end of an active ride there shall be applied a discount of 20% on the total cost of the active ride if

- the Car is left with no more than 50% of the battery empty.

[G5.11]: At the end of an active ride there shall be applied a discount of 30% on the total cost of the active ride if

- the Car is parked at a power grid station and the User plugs the Car into the power grid.

[G5.12]: At the end of an active ride there shall be a charging of 30% more on the total cost of the active ride if

- the Car is parked at more than 3KM from the nearest power grid station.

[G5.13]: When an active ride starts the registered User shall be able to activate the money saving option.

[G5.14]: By activating the money saving option, the registered User shall be able to see a list of suggested parking area.

[G5.15]: At the end of the active ride the System shall apply a discount on the total cost of the active ride if:

- the Car is parked at a parking area that was suggested to the User by the System when the User activated the money saving option.

[G5.16]: At the end of the active ride, there shall be computed the total cost for the active ride, considering all the discounts and penalties that have been applied to the active ride and a receipt shall be sent to the User.

2.6 [G6]: Management of Exceptions:

[G6.1]: The User shall be alerted in case of errors and exceptions that can subsist during the various interactions between him/her and the System. If the particular interrupted operation or set of operations can be repeated again, this is done through a new interaction.

2.7 [G7]: Correct Mailing service management:

[G7.1]: User has to correctly communicates through Mail with the System.

3 Domain properties and assumptions

The Domain Properties are expressed considering the relations among User, System, Payment System, Car's System and other parts of Environment.

3.1 [D1]: Aspects of the Environment regarding the User:

[D1.1]: The Users have the proper software platforms for using the system application (Android, Windows, Mac OS).

[D1.2]: The Users have the proper software components and drivers for allowing the application to run properly.

[D1.3]: The Users have the application installed on their devices and ready for the utilization (interaction with the System).

[D1.4]: The Users have internet access and navigation availability.

[D1.5]: The Users are registered and logged for using the service.

[D1.6]: The Users have GPS and their position can be easily detected when needed.

[D1.7]: An User cannot reserve a prenotation of a Car already in Use.

3.2 [D2]: Aspects of the Environment regarding the Payment System:

For these domain properties the presence of User is also mentioned.

[D2.1]: The Payment System receives informations of total payment that the User is required to pay.

[D2.2]: With the end of single service and during the payment, no error related to the Payment System occurs.

[D2.3]: After the payment the Payment System shall state only two exclusive results: failure or success.

[D2.4]: In case of payment failures due to the User, she/he is required to try again or to signal its problem to the System by the Payment System.

3.3 [D3]: Aspects of the Environment regarding the Car Utilization:

[D3.1]: The accurate position of each Car is known by GPS.

[D3.2]: When an end of utilization for a Car occurs, the User is instantly advised of this event.

[D3.3]: Electronical components of Cars allows the correct interaction between User and the Car's System.

3.4 [D4]: Aspects of the Environment regarding the connection to Internet:

[D4.1]: No connection problems due to the ISP come up during the interaction between each User and the System.

[D4.2]: No connection problems due to the damage of one of the involved hardware devices occur.

[D4.3]: Internet connection is faster enough for allowing the bidirectional flows of informations among entities (Users and System).

3.5 [D5]: Aspects of the Environment regarding the GPS position:

[D5.1]: No loss of the Users' position due to satellite encoding comes up during the interaction between each User and the System.

[D5.2]: No loss of the Users' position due to hardware damage occurs.

[D5.3]: No loss of the Cars' position due to satellite encoding comes up during the interaction between each User and the System.

[D5.4]: No loss of the Cars' position due to hardware damage occurs.

3.6 [D6]: Aspects of the Environment regarding the Car's System:

[D6.1]: The Car's System unlocks the Car when a confirmation for prenotation is valuated positively.

[D6.2]: The Car's System informs the driver of any issues related to possible Car's damages and problems.

[D6.3]: The Car's System informs the driver of any issues coming from the System.

[D6.4]: The Car's System informs the driver of nearest safe area for the Car (valuated by the System).

[D6.5]: The Car's System informs the driver of the instantly amount of money she/he has to pay.

[D6.6]: The Car's System is equipped with a GPS navigator.

[D6.7]: The GPS navigator can tell the User the shortest path to a given destination.

[D6.8]: The Car's System communicates with the System when necessary for signaling an issue.

[D6.9]: The Car's System encodes and tells to the System all the informations about the ride for calculating the payment.

[D6.10]: The Car's System locks the car after the utilization.

[D6.11]: The Car's System permits the communication between an User and an operator in case of assitence.

[D6.12]: The Car's System alerts the User and the System in case of some exceptions which come up in its context.

4 Glossary

- **Available Car:** it's a Car for which there is neither active reservation nor active ride related to it.
- **Active reservation:** it's a reservation that hasn't expired and hasn't yet converted in an active ride.
- **Active ride:** it's a relation between an User and a Car that represents the concrete utilization of the Car by the User.
- **Reservation:** it's a relation between an User and a Car that represents the User's will of preventing other Users to use the Car.
- **Reservation form:** it's a set of information that are required from the User in order to make a new valid reservation.
- **Valid position:** it's either a GPS position or an address position.
- **GPS position:** it's a couple of coordinates (latitude and longitude) provided by a GPS system.

- **Address position:** it's a string that represents a specific position by means of words.

- **Valid credentials:** it's a set of personal informations that have legal validity; this set includes: full name, date of birth, number of the identity card and number of the driving license.

- **Valid payment informations:** it's the set of information that allows the System to indentify an active PayPal active account.

- **Expiration fee:** amount of money (in EUR) that the User has to pay when she/he lets an active reservation expire.

- **Current relevant car's information:** it's a set of informations that represent the current state of a Car and that includes: model of the Car, name of the Car, registration number of the Car ,number of allowed passengers, state of the battery, valid position of the Car.

- **Current active ride information:** it's a set of information that represent the current state of an active ride and that includes: the amount of money that has been charged to the user so far, the duration (in minutes) of the ride so far.

- **Condition of nearness:** it represents the condition in which a User must be able to access the reserved Car because the User is close to the car.

- **Safe area:** it's set of GPS positions that is pre-defined by the System.

- **Parking penalty:** amount of money (in EUR) that the User has to pay when she/he parks the Car in a non safe area.

- **Power grid station:** it's a safe area that provides a power grid to recharge the Car.

- **Money saving option:** it's a option that can be enabled by the User for an active ride.

- **Mobile Phone:** it is one of the devices that the User can use for run the application of the System.

- **Passenger:** she/he could be the User or one of the other people who get on the Car with the User at the beginning of a ride.

- **Driver:** in the text sometimes refers to a User who is using the Car effectively, during an active ride.

- **Vehicle:** in the text sometimes refers to another term to indicate the Car.

- **Car's display:** component of the Car which allows to the User to visualize informations and interacts with the Car's System.

- **Car's sensors:** components of the Car which reveal the presence of a certain number of passengers to tell to the Car's System.
- **Car's control panel:** component of the Car which permits the interaction between the Car's System and the User.
- **On Board System:** term which indicates the Car's System in some Sequence Diagram.

5 Text Assumptions

- The logic of the application is contained in the server tecnologies from which the System works.
- We should develop a multi-platform application.
- Datas of Users are stored in a database owned by the System and are strongly protected.
- Datas of Vehicles of the company are also stored in a database and are strongly protected.
- Necessary database's transactions take place within the database and the System application contexts.
- GPS positions of Cars are always monitored and taken into the account.
- if the System changes some of its policies, the User is correctly advised by Mail or by a notification on its profile.
- If other kinds of exceptions come up during the service and outside the System context (exceptions of Payment System or of database) the User is alerted anyway. The System has to do an action to guarantee the service anyway.

6 Constraints

6.1 Regulatory policies

- The User has to be informed about the policies of the System. In particular, the User who makes a registration, has to confirm her/his acknowledgement about System's rules, her/his responsabilities and the responsabilities of the System. User confirms also the treatment of its sensible informations according to the needs of the System. Finally the System must not use notifications to send SPAM, in the respecting of the privacy low.

- The System has to be adapted to the conventions, rules and policies of the various platforms which will finally offer its services. Hence, the System has to respect market policies of Google Play Store, Apple App Store and Windows Store.

6.2 Hardware Limitations

- The System will be designed exclusively for devices (mobile phones, tablets, laptops in particular configuration) which can be costantly monitored through the web. It is important that the device has an active 3G (or higher) connectivity during all the service.

- The System will be designed exclusively for devices (mobile phones, tablets, laptops in particular configuration) which can be constantly identified geographically. It is important that the device has an active GPS service.

- The device on which the System's application will run needs a certain amount of RAM memory at minimum for processing all the tasks of the System's application. Some disk memory for storing permanent datas coming from the installation, and temporary informations coming from the running of the application, are essential too.

- The System is finally designed for all the newest version of the software (and all the hardware on which the software runs) of the companies cited before.

6.3 Interfaces to other applications

- The System interacts with the Database in which User's and Car's informations are permanently stored through MySQL. So the System is strongly dependent of this tool.

- Every money transaction between the System and the Users is regulated through PayPall (our Payment System in reality).

- The Car's System mediates the interactions among the Users and the System during the active ride context. The Car's System has, in particular, an internal software, with its specifications, which can allow these interactions.

6.4 Parallel operations

- The System has to manage parallel requestes and instances of the service as already discussed in the goals. The System's multitasking behaviour is the base to allow the utilization of the service for multiple Users. This is important for the reliability of operations too because at every instance the System has to manage sensible informations stored in its internal database.

6.5 Criticality of the application

- Having a set of restarting points is a peculiar characteristic for the System. When a critical error compromise an operation, the System has to understand how and when restore it for offering a consistant service to the User anyway.

6.6 Safety and Security considerations

- Since the payments are processed through the PayPall service, there aren't payment issues concerning the System.
- It is different for Database operations because they are managed by the Database, but it is important for the System to correctly manipulate the instances coming from different Users.

7 Identifying stakeholders

The stakeholder is a large company of car renting which wants to offer a new and innovative service in terms of energy saving and nature environment protection. This purpose will be reached accordingly to cities and territorial policies and principal electric cars constructors.

8 Requirements

Assuming the validity of the Domain Properties listed before and in order to fullfill the goals in the **Goals** paragraph, the following requirements can be derived. The requirements are explicitily referred on how the system shall correctly interact with the User and respecting the Domain Properties:

8.1 Functional Requirements

8.1.1 [R1] Registration manegement:

[R1.1]: Referring to Goal **[G1.1]** the System shall correctly accept and process the registration controlling input datas from the User.

[R1.2]: The System shall allert the user if the registration has been maden with errors.

[R1.2]: If a registration has been maden with errors, the System suggests to the User to try again for some fields of the registration form or for all of it depending on the particular intance of error.

[R1.3]: The System shall store datas of a User who has completed successfully the registration.

8.1.2 [R2] Discovering available cars and single car's informations management:

[R2.1]: After a correct registration, the System shall allow User to access and use the service, displaying to it all the pages and the messages needed at every instance.

[R2.2]: Referring to Goal **[G2.1]** the System shall match the position of the User with the position of the available Cars within a certain region. The positions are provided by the GPL service and there shall be a function which returns a List of all Cars within a certain area and available for the User in that moment.

[R2.3]: Referring to Goal **[G2.2]** the System shall instantly match the position of the User with the position of a new available Car in the same area. Hence there shall be a particular function which receives the notification of this particular event and update the List showed to the User with the new value.

[R2.4]: Referring to Goal **[G2.3]** the System shall control and correctly manage instances of prenotations. This means

- having a queue in which requestes are located at time of their formulation and
- a global trace of the first prenotation for the single Car, which permits to avoid a multiple prenotation it.

8.1.3 [R3] Car booking manegement:

[R3.1]: Referring to Goal **[G3.1]** after selecting the car, the System shall send to the User informations about the vehicle and the compilation form for booking it. The informations are retrieved from the internal Database and informations given by the User through the compilation form have to be controlled for validating the reservation.

[R3.2]: Referring to Goal **[G3.2]** the system shall control if the User who wants to book the car, hasn't a pending request on another vehicle. It is important for the System to keep track of all the pending requests and it shall either

- prevents the User of making a new request in case of a pending one
- or permits the condition, but without accepts the request and informing the User of the pending one.

[R3.3]: Referring to Goal **[G3.2]** the System shall confirm the User request of booking.

8.1.4 [R4] Reservation validity management:

[R4.1]: Referring to Goal **[G4.1]** the system shall keep track of time elapsed from the acceptance of a reservation. This is permitted by the meaning of a Timeout. The countdown starts from the Timeout value and arrives to zero. In that moment, if the User hasn't done anything concerning its reservation, the System will set the extra fee to pay.

[R4.2]: Referring to Goal **[G4.1]** the System shall cancel the active reservation if the User expresses this will.

[R4.3]: Referring to Goal **[G4.1]** the System shall cancel the active reservation if an hour from the reservation was made has passed. Hence the reservation is expired.

[R4.4]: Referring to Goal **[G4.1] [G4.3]** the System shall render available the Car in case of an expired reservation.

[R4.5]: Referring to Goal **[G4.1] [G4.5]** the System shall render available the Car in case of a canceled reservation.

[R4.5]: Referring to Goal **[G4.2]** the System shall advise the User if the reservation expires and about the extra fee to pay.

8.1.5 [R5] Car's usage management:

[R5.1]: Referring to Goal **[G5.1]** the System shows correctly to the User the possibility to be notified about the nearness. The System in fact controls the position of the User and the position of the reserved Car at any time and, within a certain distance between the two entities, alerts the User of the event.

[R5.2]: Referring to Goal **[G5.2]** the System unlocks the Car by receiving the answer on the notify showed to the User. When the System informs the User about the nearness to the Car, a pending instance of confirmation is created. The reply to this instance comes from the User and the System, after an effective matching of positions, sends to the Car's System the request of unlocking the Car.

[R5.3]: Referring to Goal **[G5.3]** the System shall:

- convert the active reservation in a new active ride
- inform the User of the beginning of a new active ride via the Car's System.
- start charging the User of an amount of money to pay per minute, the User knows about this by the Car's System too.

[R5.4]: Referring to Goal **[G5.4]** the System shall give to the User all relevant informations about the current ride via the Car's System, the Car's System effectively interacts with the User through Car's display, Car's sensors and Car's control panel.

[R5.5]: Referring to Goal **[G5.5]** the System shall stop the ride instance and the charge towards the User.

[R5.6]: Referring to Goal **[G5.6]** the System shall detects if the User parked the Car in a non safe area via GPL signalation. Hence it shall notify the User of the extra fee to pay in this case by sending her/him a notification on its device.

[R5.7]: Referring to Goal **[G5.7]** the System shall detects if the User and all the other passengers get off the car via GPL signalation and the end of the service signalations. Hence the Car's System is allerted and the car is locked.

[R5.8]: Referring to Goal **[G5.8]** the System shall detects if the User is driving the car with at least two other passengers on board. This shall be possible because the Car's System detects the presence of a certain amount of people through Car's sensors and sends a notification to the System. Once Verified the situation, the System applies the discount at the end of the ride and using the Payment System for interacting with the User.

[R5.9]: Referring to Goal **[G5.9]** the System shall detect the battery level at the end of the current drive. If it is left no more than 50% empty, the System keeps track of the discount of 20% that it shall apply on the total cost and, afterwards, it sends datas for the Payment System.

[R5.10]: Referring to Goal **[G5.10]** the System shall be allerted by the Car's System if the Car is left by the User in a power grid station and if the Car is plugged to the power grid. In this case the System shall apply the mentioned discount. Datas for the payment are sended to the Payment System afterwards.

[R5.11]: Referring to Goal **[G5.11]** the System shall detect if the Car has been parked at more than 3 KM from a power greed station. In this case the last User is required to pay the mentioned fee. The User is alerted of this by the utilization of the application on its device.

[R5.12]: Referring to Goal **[G5.12]** the System shall be advised by the User and through the Transactional System if the Money Saving Option is enabled at the beginning of the ride. If it so, the appropriate solution will be implemented for this kind of utilization.

[R5.13]: Referring to Goal **[G5.13]** if the money saving option is enabled the System shall compute in real time the distance among car position and the nearest safe parking areas. The resulting datas are sended again to the Car's System, which displays results to the User via Car's display.

[R5.14]: Referring to Goal **[G5.14]** if the money saving option is enabled and the System detects that the User has parked the Car in one of the last and suggested safe parking area, the mentioned discount is applied. The resulted information is known by the User through the Car's System and/or visualizing a notification on its device.

[R5.15]: Referring to Goal **[G5.15]** at the end of an active ride the System has all the fields related to discounts and amends that are correctly compiled and ready to be sended to the Payment System in order to make a receipt towards the User.

8.1.6 [R6] Management of the Exceptions by the System:

[R6.1]: Referring to Goal **[G6.1]** the System shall correctly manage every kind of exception that can be generated in the context. Some of those exceptions are not showed directly to the User but other has to be. In any case it's important for the System to configured the point in which has to restart in case of error and which action is required from the User after the communication. The deep analysis of all the cases of exceptions shall be true for all the situations mentioned.

8.1.7 [R7] Correct manegement of Mailing service by the System:

[R7.1]: Referring to Goal **[G7.1]** the System shall configure automatic messages to be sent to the User in case of events that need this kind of service.

8.2 Nonfunctional requirements

8.2.1 User Interface

1. **Main Menu:** it is the first interface showed to the User after her/his log to the service. In the Main Menu the User can see the principal informations associated by the System to her/him: her/his name and surname and her/his email address. There are some links to the other following interfaces:

   (a) **User Profile**

   (b) **Reserve a Car.**

2. **User Profile:** it is the profile of the User in which there are detailed informations about her/him and in particular, the informations already

showed in the Main Menu, her/his home address, her/his telephone number and her/his mobile number, there is also the possibility to show an image for the profile. The logged User can always modify the parameters through clicking on the appropriate button.

3. **Reserve a Car:** it is the interface in which it is displayed to the User the map of the area in which her/his position is currently revealed by the System. In particular, the User is in the centre of the Map and she/he can find the available Cars around her/him in real time. If one of the Car within the area becomes unavailable, the User is instantly informed of it by seeing on the screen of its mobile the Car becoming from green to orange and after a while disappearing from the map. Once the User clicks on one of available Car, the System redirect her/him on the

   (a) **Reservation Menu.**

4. **Reservetion Menu:** it is the interface in which the User compiles effectively the reservation form for booking the selected Car. Some characteristics of the Car are immediatly showed by the System to the User (other detailed characteristics are retrieved after the prenotation). After the confirmation the User is redirected again to the previous interface, but on the map there is only the reserved Car right now. There is also the "nearby button" which will be enabled by the System after the condition of nearness between the User and the Car is verified. For this case there is also the shortest path from the User to the Car, which can help the User to reach the reserved vehicle.

8.2.2 Documentation

We will refer to the following documents for having a global view of the System in terms of its functionalities to be implemented without redundancy afterwards. These documents are designed to be a complete sketch of what the System is intended to be:

- **RASD (Requirement Analysis and Specification Document):** it is a first attempt to deeply understand what the System shall do and not do accordingly to the goals. RASD is particularly designed to sharply design which are the requirements for modeling the System. Domain Properties and Text Assumptions aids instead to take which are the "border" of our System into account in terms of functionalities and which other responsabilities are of other Actors inside the Environments. Diagrams and Alloy modeling tries finally to shows the different cases of utilization and the different situations the System could be during the offering of the service.
- **DD (Design Document):** it is the document in which specific aspects of the System will be studied. There will be the reference to the adopted tecnologies for the creation of the System and a particular emphasis on its structure.

9 Scenarios

9.1 Scenario: Luca registers his new PowerEnJoy account.

Luca accesses the system via the mobile app for the first time and has to choose whether to log in with his credentials or create a new account. Luca chooses to register a new account; the system shows a form that must be filled with Luca's credentials and payment information. Luca fills the form and gives confirmation to the system. Luca receives an email with the confirmation of the registration and the password that Luca will use to access the system.

9.2 Scenario: Federico searches for nearby cars.

Federico is in need for a car. First he opens the PowerEnJoy application by providing his credentials, then he uses the app to search for a nearby available car. Federico is shown the location of all available cars within a fixed number of kilometers.

9.3 Scenario: Valeria reserves a car for usage.

Valeria uses the PowerEnJoy application to look for available cars and finds one near to her current location. She decides to reserve the car and pick it up later. When Valeria reaches the car less than an hour later, she can unlock it and start her drive.

9.4 Scenario: Charlie books a car and doesn't pick up it within one hour from the reservation.

Charlie reserves a free car by means of the mobile app. As time goes by, Charlie realizes that he doesn't need the car anymore but doesn't cancel his reservation. After one hour from when Charlie made it, the reservation is cancelled and the car can be reserved by some other user. Charlie receives a notification on his smartphone that informs him about the cancellation of his reservation and of the fee of 1 EUR he is receiving because the reservation expired.

9.5 Scenario: Sally gets close to a reserved car to unlock it.

Sally has previously reserved a car she intended to use. When she gets near it, she uses the PowerEnJoy application to confirm her location. The system then unlocks the car. Sally is now able to enter the car and use it.

9.6 Scenario: Schroeder keeps track of the cost of his ride through the screen on the car.

After Schroeder unlocks his car and enters it, he sees the car's screen is now displaying information about the duration of the ride and its current cost. As time goes by, he can see the cost rise so he can keep track of it.

9.7 Scenario: Patty parks the car in a safe area and leaves it.

Patty arrives in a safe area and decides to park the car. Patty stops the engine, leaves the key in the appropriate location and exits the car. As she closes the door, the system locks the car automatically and stops charging the user. Patty receives a notification on his smartphone that informs him about the end of the ride.

9.8 Scenario: Marcie parks and the car and leaves it with no more than 50% of the battery empty.

Marcie parks the car in a safe area and leaves it. The system behaves as described in Scenario 3.7. At the end of the ride, the system checks the status the car's battery. If the battery is left with no more than 50% of the battery empty, the system applies a 20% discount on the ride. Marcie receives a notification on her smartphone that informs her about the end of the ride and the discount that has been applied thanks to her virtuous behavior.

9.9 Scenario: Linus searches for cars near to a specific location.

Linus is about to visit a friend's house. He needs to know if he's going to be able to find any available car in its proximity, for his trip back home. Linus uses the PowerEnJoy app to search for available cars near his friend's address. He is shown all the available cars within a fixed number of kilometers from his friend's house.

9.10 Scenario: Snoopy brings two passengers on a ride and receives a 10% discount.

Snoopy and two friends long to spend some time together driving on the hills and taking some fotos of the countryside. While they are still in the village they check their mobile devices to search for a PowerEnJoy car and reserve one for their use. As soon as they enter the car, the system is notified that there are at least two passengers in the car beside the driver. The three friends are then shown a message on the car screen, informing them that their behaviour is worth Snoopy a 10% discount on the ride.

9.11 Scenario: Lucy recharges the car after the ride and gets a 30% discount.

Lucy has been using a PowerEnJoy car. She parks the car at a parking area equipped with charging stations. She then plugs the car into one of the rechargers. When she checks her PowerEnJoy app for the price of the ride, she sees that a discount of 30% has been applied to it.

9.12 Scenario: Mario leaves the car too far from a charging station.

At the end of his ride, Mario leaves the PowerEnJoy car far from the city. The system detects the car is farther than 3 km from the nearest charging facility. As a result, Mario is charged a fee of 30% the cost of the ride.

9.13 Scenario: Luigi leaves the car with a nearly empty battery.

At the end of his ride, Luigi leaves the PowerEnJoy car far from the city. The system detects the car's battery is less than 20% full. As a result, Luigi is charged a fee of 30% the cost of the ride.

9.14 Scenario: Peach reserves a car and enables the "money saving option".

Peach wants to reserve a car. Luca chooses the car and decides to active the money saving option in order to get a discount on the ride. The system shows on the map the locations where she can leave the car in order to the get the discount. Peach proceeds with the confirmation of the reservation.

10 Actors

10.1 The User.

The human end user who will interact with the PowerEnJoy mobile application.

10.2 The User's device.

The hardware-software system that could be the user's mobile phone, including both the operating system and the actual hardware, with the necessary support for GPS, communication, and a working system necessary for the app to work properly.

10.3 The Car's System.

The System will communicate with the cyber-physical system that is the Car through an appropriate communication protocol.

10.4 The Network.

The System will use the existing networking infrastructure for all necessary communications.

10.5 The Payment System.

All payment transactions will happen through an external existing payment system. The PowerEnJoy system will communicate with such payment system in order for the necessary information flow to happen between the two.

11 Use Case identification

11.1 NewUserRegistration

Use Case name	NewUserRegistration
Actors	• NonLoggedUser • Network
Entry condition	NonLoggedUser communicates the will to register a new User
Flow of events	• The System shows the registration form the NonLoggedUser • NonLoggedUser fills the registration form with the information required by the registration form • NonLoggedUser submits the filled registration form to the System via the Network
Exit condition	The System confirms the creation of a new User and sends an email, carrying the password that has been generated by the system and associated to the new User, to the email address that was specified in the registration form.
Exceptions	• If the Network doesn't deliver the information provided by the NonLoggedUser, the System informs the NonLoggedUser and allows the NonLoggedUser to perform the operation again • If the NonLoggedUser provides invalid information, the System informs the NonLoggedUser about what's wrong with the provided information • If the NonLoggedUser doesn't receive the email carrying the password, the NonLoggedUser informs the System.
Special requirements	The System must deliver the email carrying the password to the user within 1 minute from when the new User was created.

11.2 UserLogin

Use Case name	UserLogin
Actors	• NonLoggedUser • Network
Entry condition	NonLoggedUser communicates the will to log in
Flow of events	• The System shows the login form to the NonLoggedUser • NonLoggedUser fills the registration form with the personal email and the password that was provided via email after the registration of a new User. • NonLoggedUser submits the filled login form to the System via the Network
Exit condition	The system confirms the log in
Exceptions	• If the Network doesn't deliver the information provided by, the System informs the NonLoggedUser and allows the NonLoggedUser to perform the operation again • If the NonLoggedUser provides invalid information, the System informs the NonLoggedUser about what's wrong with the provided information

11.3 FindAvailableCars

Use Case name	FindAvailableCars
Actors	• LoggedUser • Network • MobilePhone • AvailableCar
Entry condition	LoggedUser communicates the will to see available cars around his/her current position
Flow of events	• The LoggedUser sumbits a valid position to the System. • The System searches AvailableCars within a certain distance from the position submitted by the User
Exit condition	The System shows to the LoggedUser a map showing the position of AvailableCars that have been found
Exceptions	• If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again • If there aren't AvailableCars nearby the position specified by the LoggedUser, the system explicitly informs the LoggedUser about this condition

11.4 ProvideGPSPosition

Use Case name	ProvideGPSPosition
Actors	• LoggedUser • MobilePhone • Network
Entry condition	LoggedUser is asked to provide a valid position by the System, the LoggedUser decides to provide a GPS position
Flow of events	• The MobilePhone returns the current GPS position
Exit condition	The System receives a GPS position as response to the request of providing a valid position
Exceptions	• If the MobilePhone isn't able to provide the current GPS position, the System informs the LoggedUser and suggests him to provide an address

11.5 ProvideAddressPosition

Use Case name	ProvideAddressPosition
Actors	• LoggedUser • Network
Entry condition	LoggedUser is asked to provide a valid position by the System, the LoggedUser decides to provide an address
Flow of events	• The System shows a text field to be filled with an address. • The LoggedUser fills the text field • The LoggedUser submits the information to the System
Exit condition	The System receives an address as response to the request of providing a valid position
Exceptions	• If the LoggedUser provides an invalid position, the System informs the LoggedUser about what's wrong with the provided position

11.6 ReserveAvailableCar

Use Case name	ReserveAvailableCar
Actors	• LoggedUser • AvailableCar • Network
Entry condition	LoggedUser selects an AvailableCar from the ones that are on the map that shows the AvailableCars nearby the LoggedUser position
Flow of events	• The System shows the reservation form for the specific car whose identity is provided by the LoggedUser • The LoggedUser sees all the information about the AvailableCar provided by the reservation form • The LoggedUser confirms the will to reserve the AvailableCar via the Network
Exit condition	The System confirms the reservation for the AvailableCar and shows to the LoggedUser the information about the active reservation that has been placed
Exceptions	• If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again • If the System isn't unable to perform the operation, the System informs the LoggedUser about what's the problem
Special requirements	The insertion of a new active reservation by a LoggedUser should be transactional with respect to other LoggedUsers that want to insert a new active reservation for the same AvailableCar

11.7 ViewActiveReservation

Use Case name	ViewActiveReservation
Actors	• LoggedUser • Network
Entry conditions	• The System has placed a new active reservation for the LoggedUser • The LoggedUser communicates to the System the will to see the information about her/his active reservation via the Network
Flow of events	• The System shows the information about the active reservation of the LoggedUser
Exit condition	The LoggedUser sees all the information about her/his active reservation
Exceptions	• If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again • If the System is unable to perform the operation, the System informs the LoggedUser about the problem

11.8 UserDeletesActiveReservation

Use Case name	UserDeletesActiveReservation
Actors	• LoggedUser • Network
Entry condition	The LoggedUser communicates to the System the will of deleting her/his active reservation via the Network
Flow of events	• The System receives the information required for deleting an active reservation • The System asks the LoggedUser to confirm the will to delete the active reservation • The LoggedUsed confirms
Exit condition	The System deletes the active reservation of the LoggedUser
Exceptions	• If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again • If the System isn't unable to perform the operation, the System informs the LoggedUser about what's the problem
Special requirements	The deletion by LoggedUser deletion of an active reservation must be transactional

11.9 ExpireActiveReservation

Use Case name	ExpireActiveReservation
Actors	• LoggedUser • Network
Entry condition	The System has counted one hour from when the active reservation was made by a LoggedUser.
Flow of events	• The System deletes the active reservation • The System charges the LoggedUser of the expiration fee
Exit condition	The System inform the LoggedUser about the reservation expiration and the payment of the expiration fee via a notification and an email.
Exceptions	none
Special requirements	The deletion by expiration of an active reservation must be transactional

11.10 StartRide

Use Case name	StartRide
Actors	• LoggedUser • MobilePhone • Network • ReservedCar
Entry condition	The LoggedUser views her/his active reservation for the ReservedCar and the GPS position of the LoggedUser is within a certain distance from the ReservedCar
Flow of events	• The System sends to the LoggedUser a notification about the possibility to notify her/his condition of nearness to the ReservedCar • The LoggedUser submits to the System her/his condition of nearness to the ReservedCar via the Network. • The System unlocks the ReservedCar • The System associates a ride to the active reservation • The System starts charging the LoggedUser for the the parking fee • The System informs the LoggedUser about the beginning of the ride and the state of the charging via the on board system of the ReservedCar
Exit condition	The System notifies the beginning of the ride to the LoggedUser
Exceptions	• If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again • If the System is unable to determine the GPS position of the LoggedUser, the System informs the LoggedUser and suggests her/him about how to solve the problem • If the System is unable to perform the operation, the System informs the LoggedUser about the problem
Special requirements	• The ReservedCar must be unlocked by the System within 1 minute from when the LoggedUser notified the condition of nearness to the ReservedCar • The association the ride to the the active reservation shall be transactional.

11.11 EnableMoneySavingOption

Use Case name	EnableMoneySavingOption
Actors	- LoggedUser
Entry condition	The LoggedUser activates the money saving option
Flow of events	- The System asks the LoggedUser to provide an address - The LoggeUser provides an address - The System shows to the LoggedUser a list of stations that are within a certain distance from the the position specified by the LoggedUser
Exit condition	The System associates the list of stations shown to the LoggedUser to the ride of the LoggedUser
Exceptions	- If the Network doesn't deliver the information provided by LoggedUser, the System informs the LoggedUser and allows the LoggedUser to perform the operation again

11.12 StartActiveRide

Use Case name	StartActiveRide
Actors	• LoggedUser • ReservedCar
Entry condition	The LoggedUser starts the engine of the ReservedCar
Flow of events	• The System associates a new active ride to the ride • The System starts charging the LoggedUser for the active ride's tariff • On the basis of the state of the ReservedCar the System applies discounts and penalties to the active reservation of the LoggedUser. • The System informs the LoggedUser about the beginning of the active ride and the state of the charging via the Car's System of the ReservedCar
Exit condition	The System notifies the beginning of the active ride to the LoggedUser
Exceptions	none

11.13 EndActiveRide

Use Case name	EndActiveRide
Actors	• LoggedUser • ReservedCar
Entry condition	The LoggedUser stops the engine of the ReservedCar
Flow of events	• The System ends the active ride. • The System stops charging the LoggedUser with the active ride's tariff • The System starts charging the LoggedUser for the the parking fee • The System informs the LoggedUser about the ending of the active ride and the state of the charging via the Car's System of the ReservedCar
Exit condition	The System notifies the ending of the active ride to the LoggedUser on the Car's System of the ReservedCar
Exceptions	none

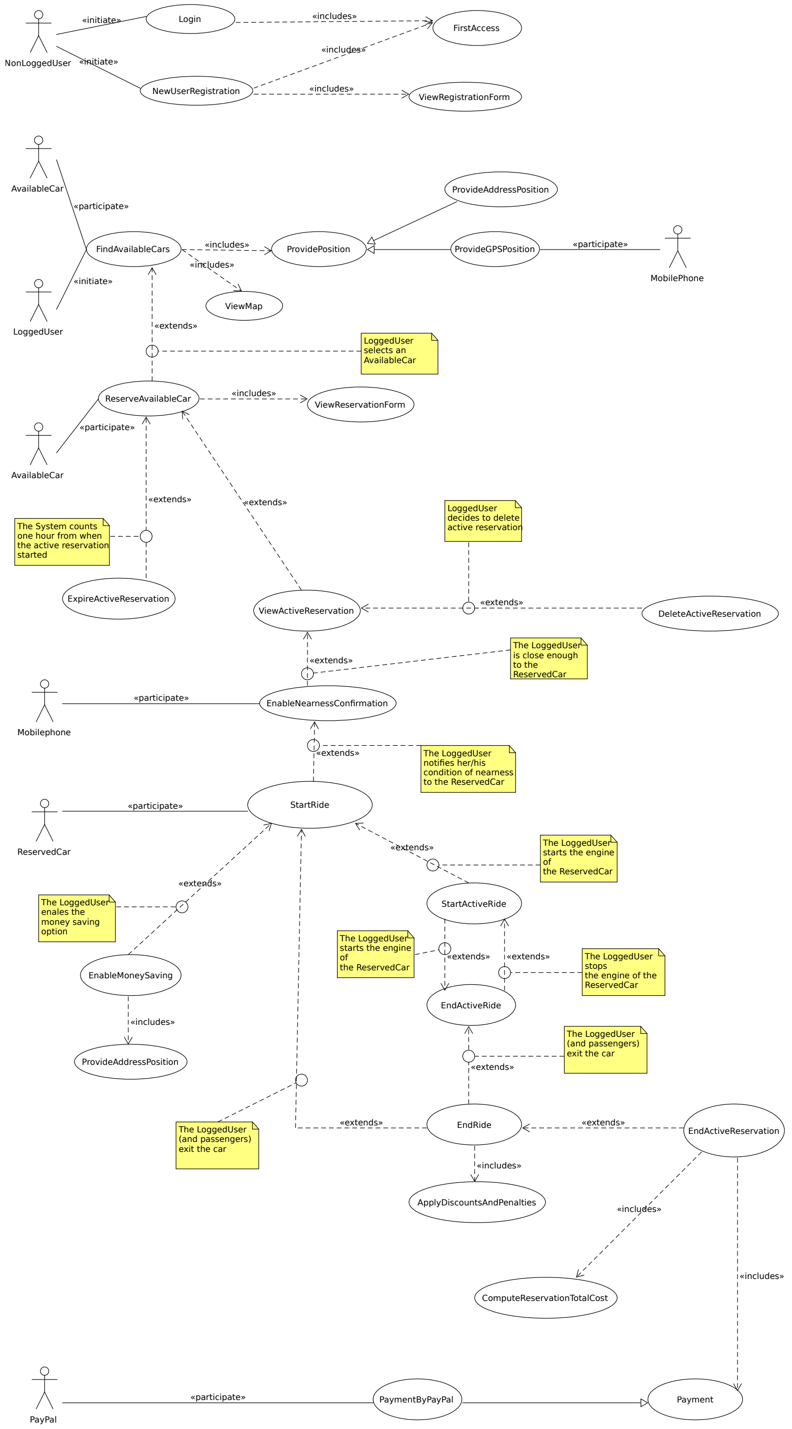
11.14 EndRide

Use Case name	EndRide
Actors	• LoggedUser • ReservedCar
Entry condition	The LoggedUser and other passengers exit the ReservedCar
Flow of events	• The System ends the ride associated to the active reservation of the LoggedUser for the ReservedCar • On the basis of the state of the ReservedCar, the System associates discounts and penalties to the activer reservation • The System locks te ReservedCar • The System stops charging the LoggedUser for the parking ride's tariff • The System informs the LoggedUser about the ending of the ride and the state of the charging via a notification
Exit condition	The System notifies the ending of the ride to the LoggedUser
Exceptions	none
Special requirements	The System shall lock the car immediately as the LoggedUser exits the ReservedCar

11.15 EndActiveReservation

Use Case name	EndActiveReservation
Actors	- LoggedUser - ReservedCar
Entry condition	The LoggedUser ends a ride
Flow of events	- The System ends the active reservation of the LoggedUser for the ReservedCar - The System computes the total cost of the active reservation and charges the LoggedUser for that amount of money - The System executes the payment for the total cost of the active reservation - The System sends a receipt of the ended active reservation and the payment to the LoggedUser via email - The System informs the LoggedUser about the ending of the active reservation via a notification
Exit condition	The System informs the LoggedUser about the ending of the active reservation
Exceptions	none
Special requirements	The System must send the receipt of the ended active reservation within 1 minute

12 UML Use Case diagram



13 UML Sequence diagrams

13.1 Registration phase

1: A non registered User asks for obtaining a registration to the System, the System understands who is the User by reading his/her IP address and the Socket Port designed for the application.

1.1: The System puts the User's request inside a queue. The User waits for a System aswer during this time.

1.2: The System informs the User that his/her request is OK. There's a confirmation message inside the aswer.

2: In a second time The System sends to the User the registration form to be compiled. UserParameters refers to the object which will contain Datas of the User.

2.1: The User compiles the form sent by the System.

2.2: The System receives the compiled form from the User.

2.3: The System processes User's compiled form and analyzes if every field is well compiled.

loop: Two System's variables govern the loop process: Registration: it is setted as True if and only if the System successfully completes the Registration of the current User and Exit: it points out the User decision to leave the context without complete its registration.

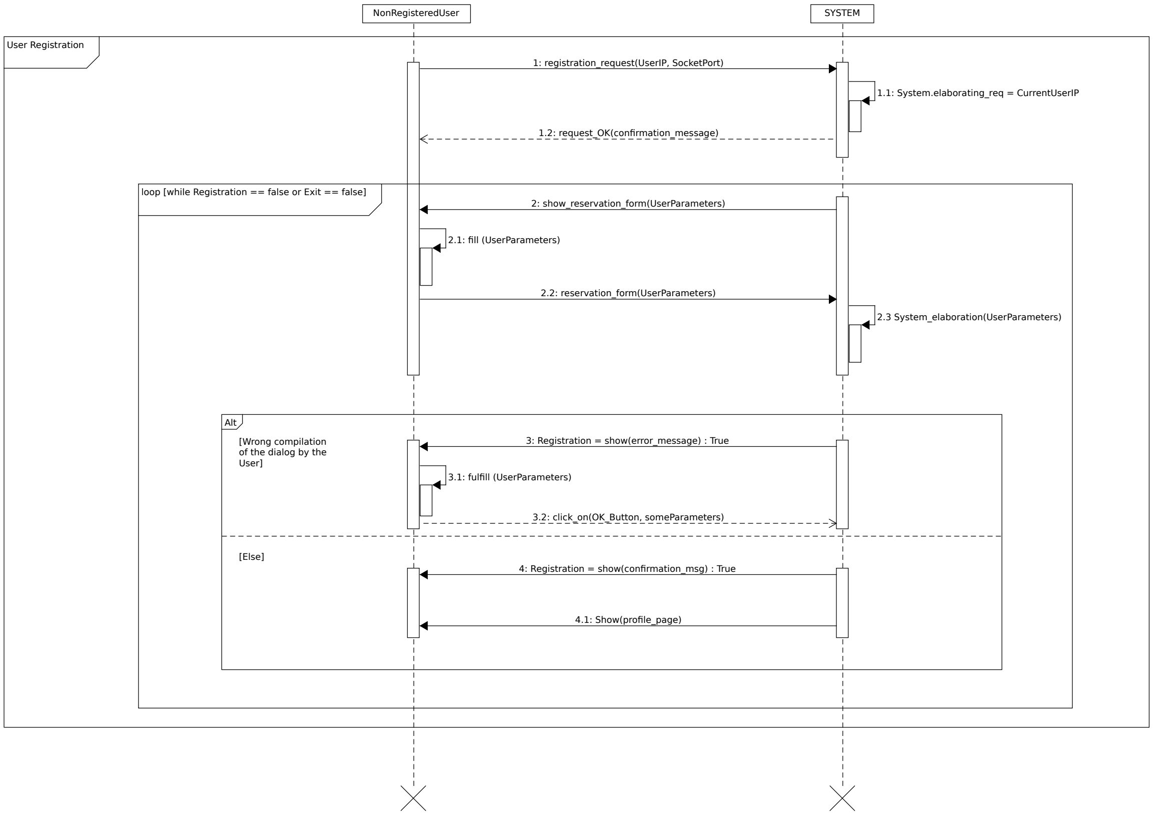
3: If there's an error or more inside the compiled form, the System informs the User about the specific lack through the error message. Hence the User is required to check the error and compile again the missing part of the form.

3.1: The User effectively compiles again the form.

3.2: The User confirms again the compilation. The missing parameters are contained in a variable someParameters, which will be trasferred to the System during the sending.

4: If the form compiled by the User is OK, the System informs the User of the correctness of all the transaction. Now the User is registered to the System and can gain its services.

4.1: The System automatically logs the User for the first time.


NonRegisteredUser
SYSTEM
User Registration
1: registration_request(UserIP, SocketPort)
1.1: System.elaborating_req = CurrentUserIP
1.2: request_OK(confirmation_message)
loop [while Registration == false or Exit == false]
2: show_reservation_form(UserParameters)
2.1: fill (UserParameters)
2.2: reservation_form(UserParameters)
2.3 System_elaboration(UserParameters)
Alt
[Wrong compilation of the dialog by the User]
3: Registration = show(error_message) : True
3.1: fulfill (UserParameters)
3.2: click_on(OK_Button, someParameters)
[Else]
4: Registration = show(confirmation_msg) : True
4.1: Show(profile_page)

13.2 Autentication phase

1: The User sends to the System an autentication request. The request contains User's name and User's password.

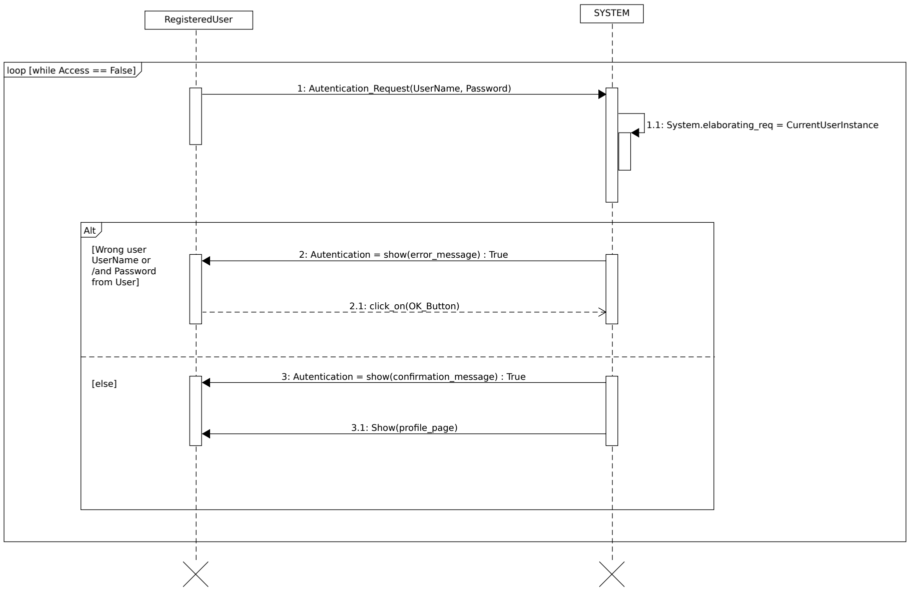
1.1: The System puts the request in a queue and, after the elaborations of older ones, analyzes it.

2: The Access is denied if User's password and/or User's name aren't correct, the System informs the User by sending her/him an error message.

2.2: The User is asked to confirm the receipt of the error message instance.

3: Once the System has verified that inputs from the User are correct, it sends to the User the confirmation message for the autentication.

3.1: The profile page is finally showed to the User.


RegisteredUser
SYSTEM
loop [while Access == False]
1: Autentication_Request(UserName, Password)
1.1: System.elaborating_req = CurrentUserInstance
Alt
[Wrong user UserName or /and Password from User]
2: Autentication = show(error_message) : True
2.1: click_on(OK_Button)
[else]
3: Autentication = show(confirmation_message) : True
3.1: Show(profile_page)

13.3 Searching and Booking a nearby Car

1: The User sends a request of displaying all the nearby Cars to the System.

1.1: A list of all nearby Cars is returned from the previous request.

2: The User selects a Car from the list.

2.1: The System checks if the Car is still available (another User could have asked for that car before and quickly).

3: The confirmation is setted as false and User is advised of this event.

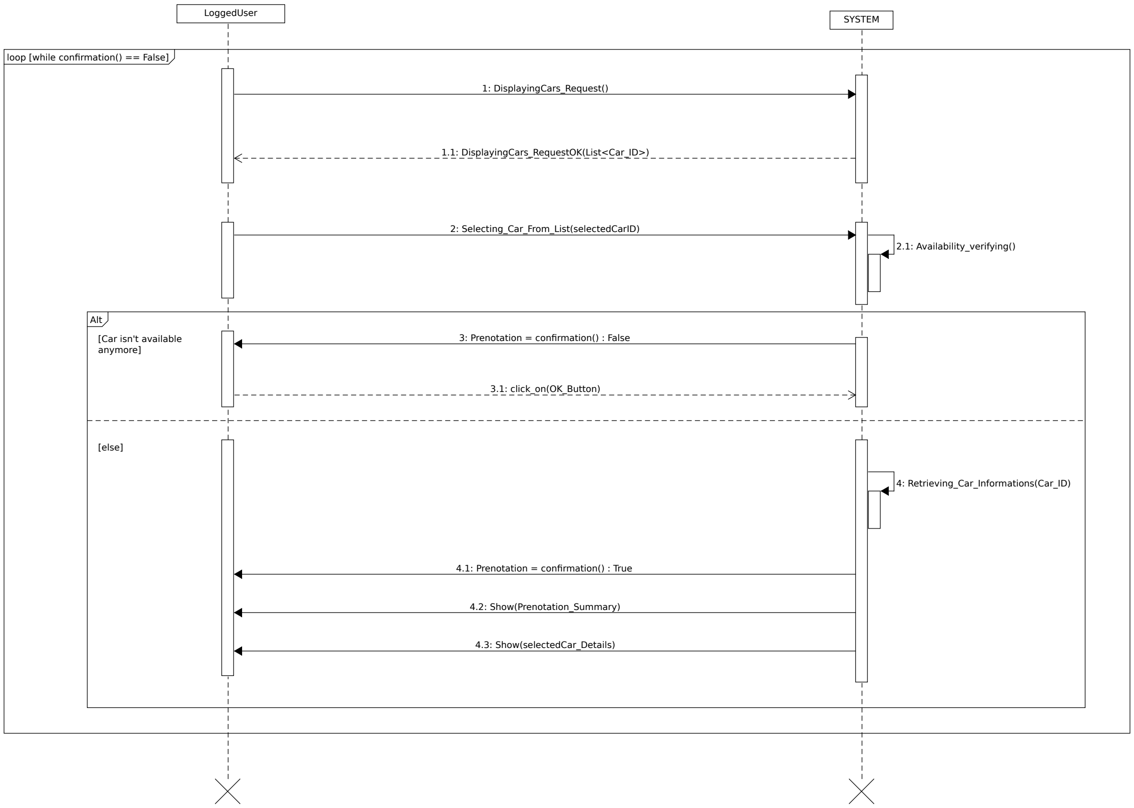
3.1: The User confirm the notification, she/he is afterwards called to select another Car from available ones.

4: Confirmation is true and the System retrieves Car's informations.

4.1 The User is informed of the correctness of selection.

4.2: System shows to the User prenotation details automatically.

4.3: System shows to the User selected Car's details automatically.


LoggedUser
SYSTEM
loop [while confirmation() == False]
1: DisplayingCars_Request()
1.1: DisplayingCars_RequestOK(List<Car_ID>)
2: Selecting_Car_From_List(selectedCarID)
2.1: Availability_verifying()
Alt
[Car isn't available anymore]
3: Prenotation = confirmation() : False
3.1: click_on(OK_Button)
[else]
4: Retrieving_Car_Informations(Car_ID)
4.1: Prenotation = confirmation() : True
4.2: Show(Prenotation_Summary)
4.3: Show(selectedCar_Details)

13.4 Time Expiration

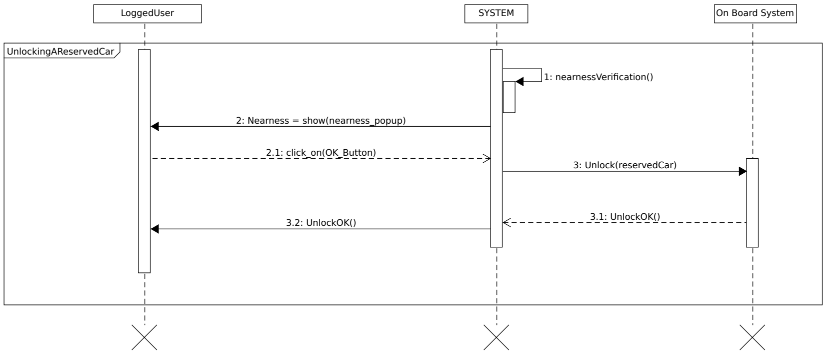
4: A System's confirmation of occurred Reservation is sent to the User.

4.1: The System waits for the User's proximity to the Car, meanwhile it updates the countdown starting from the variable Timeout = 60 minutes (the decrementing is supposed to be in seconds).

4.2: If the countdown reaches 0 s and the User isn't in proximity of the Car yet, the System alerts the User of the expiration of the reservation.

4.3: The User is required to confirm the System notification.

4.4: The User is warned about the extra fee she/he has to pay for this case.



13.5 The System unlocks a Car

1: The System checks the nearness condition betweed the User and the reserved Car.

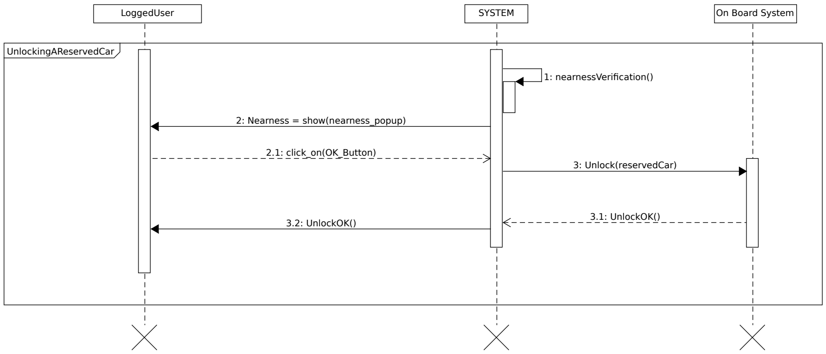
2: When the nearness condition is satisfied, the User is alerted and a popup to be clicked is sended to him (it is supposed to be a button enabled at time of nearness).

2.1: The popup is clicked by the User, the System receives the corresponding notification.

3: The System unlocks the reserved Car.

3.1: Car's System notify the System that the Car is now unlocked.

3.2: The System alerts the User that the Car is now unlocked.



13.6 End of a ride and payment

1: The User informs the Car's System of the end of the Ride.

2: Car's System asks a confirmation of end of ride to the User.

2.1: The User confirms the end of the ride.

3: Car's System alerts the System of the end of the ride for the current User.

4: The System asks for User Datas relatively to the ended ride.

4.1: Car's System sends required Datas to the System (currentRide is the object in which all this datas will be allocated).

5: The System processes the request with Datas obtained from the Car's System and for the Payment System.

6: The System launches a payment instance to the Payment System in which a specific request for User's payment is formulated.

6.1: The Payment System processes the request sent by The System.

6.2: The Payment System informs the System that the request was well processed.

7: The Payment System launches a payment request towards the specified User.

7.1: There is a payment transaction which could be or not be well done. In this last case the User is required to try again with the payment.

8: The payment completed with errors for some reasons. The System is informed of this event.

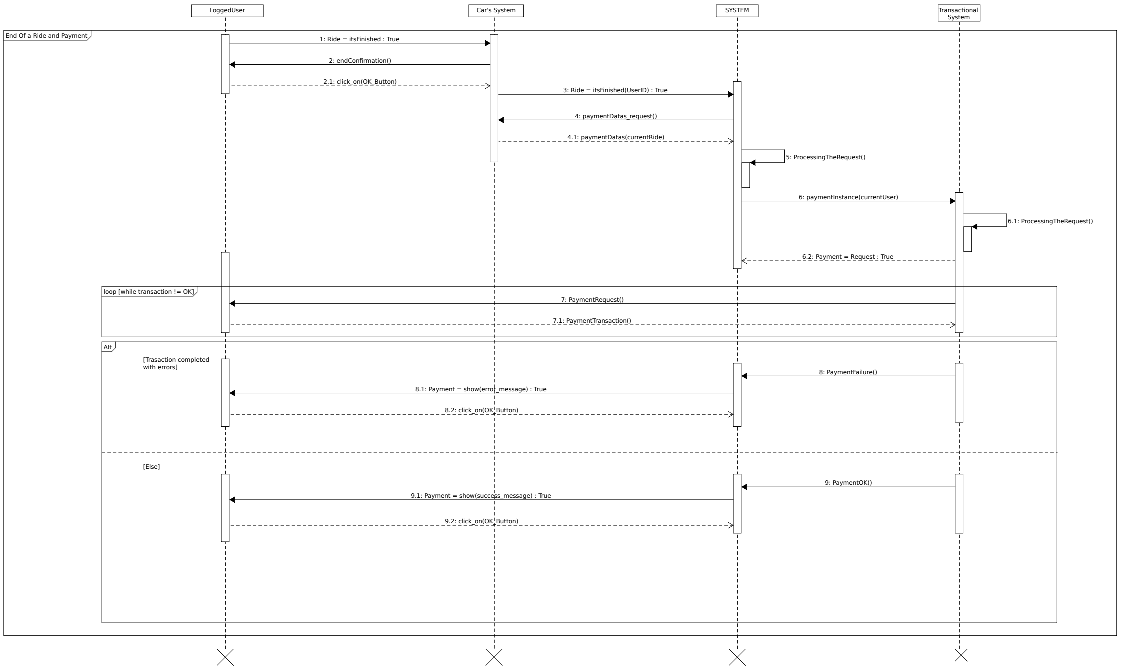
8.1: The User is informed of the fault in the payment by the System.

8.2: The User confirms the notification of the System. Now the User is required to try again with the payment or to signaling its problem to the System directly.

9: The payment is OK. The Payment System informs the System of this event.

9.1: The User is also informed by the System of the success of the payment.

9.2: The User confirms the notification of the System.


LoggedUser
Car's System
SYSTEM
Transactional System
End Of a Ride and Payment
1: Ride = itsFinished : True
2: endConfirmation()
2.1: click_on(OK_Button)
3: Ride = itsFinished(UserID) : True
4: paymentDatas_request()
4.1: paymentDatas(currentRide)
5: ProcessingTheRequest()
6: paymentInstance(currentUser)
6.1: ProcessingTheRequest()
6.2: Payment = Request : True
loop [while transaction != OK]
7: PaymentRequest()
7.1: PaymentTransaction()
Alt
[Trasaction completed with errors]
8: PaymentFailure()
8.1: Payment = show(error_message) : True
8.2: click_on(OK_Button)
[Else]
9: PaymentOK()
9.1: Payment = show(success_message) : True
9.2: click_on(OK_Button)

13.7 Activating the Money Saving Option

1: The User activates the Money Saving Option by notifying it to the Car's System (or, alternatively, to the System during the reservation phase).

2: the Car's System wants to know from the User the final destination of her/his ride. The term Destination refers to an object where geographic coordinates of the destination will be compiled by the User.

2.1: The User fills the form showed by the Car's System with the coordinates of the destination.

2.2: The User sends his/her reply to the Car's System.

3: The Car's System checks if the given destination is correct.

loop: the Car's System continuing on the asking of a correct Destination if and only if: CorrectDestCondition is false, is the case in which the User has specified an incorrect position, or the Exit condition is true and this is the case in which the User decides to come back to the previous context without finishing the operation.

4: the Car's System alerts the User that the given destination is an incorrect one.

4.1: the User fills again the coordinates of the destination.

4.2: the User gives back the Destination instance to the Car's System.

5: the Car's System asks the System to give to it a set of Safe Area within a certain distance from the destination given by the User.

5.1: the System processes the request of the Car's System, calculating the coordinates of Safe Areas from the given destination.

5.2: the System returns Safe Areas to the Car's System.

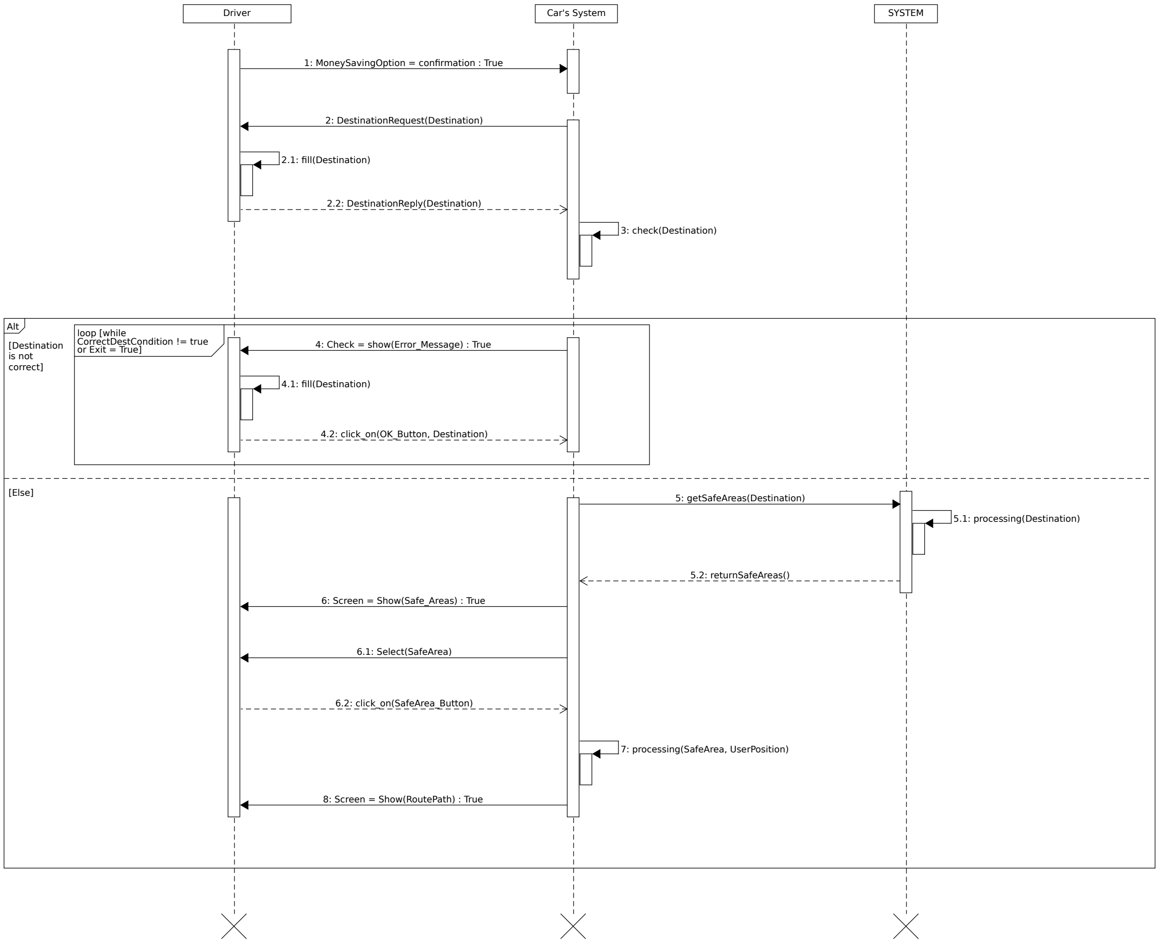
6: the Car's System shows the Safe Areas to the User by diplaying them on the screen on board.

6.1: the User is required to select one of the given Safe Areas.

6.2: for selecting one of the given Safe Areas, the User clicks on one of them.

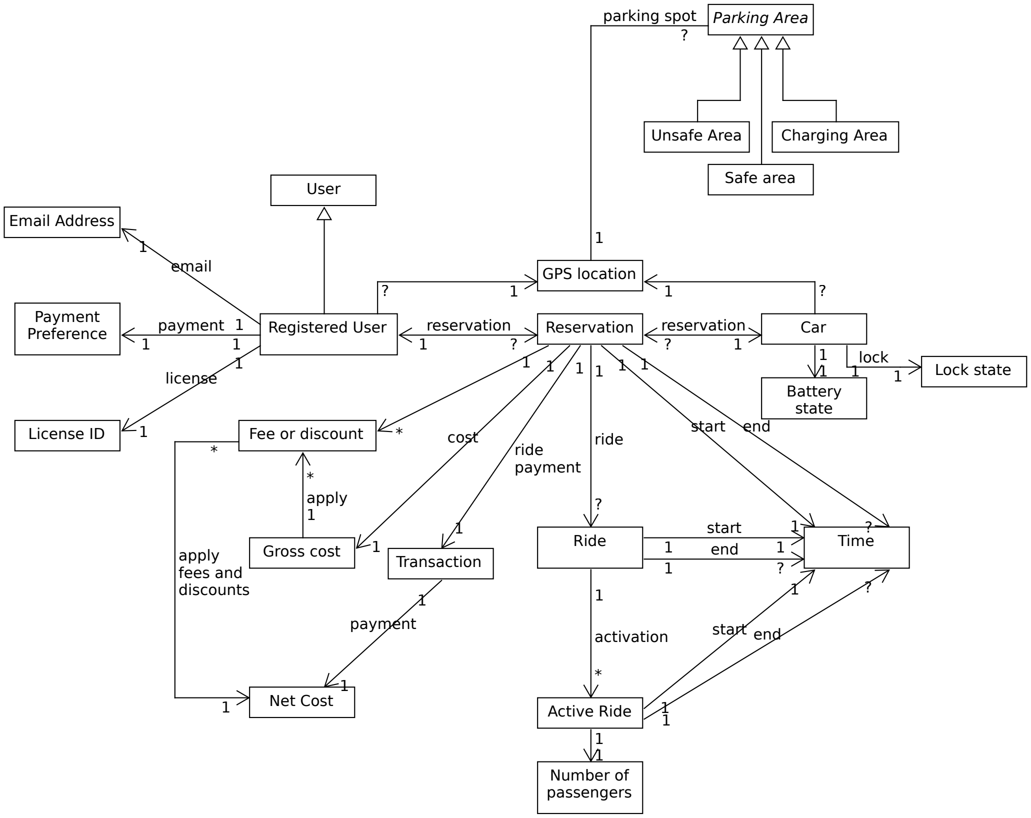
7: the Car's System calculates the shorter path from the actual position of the Car to the given destination.

8: the Car's System shows the calculated path to the Driver by displaying it on the Car's display.


Driver
Car's System
SYSTEM
1: MoneySavingOption = confirmation : True
2: DestinationRequest(Destination)
2.1: fill(Destination)
2.2: DestinationReply(Destination)
3: check(Destination)
Alt
[Destination is not correct]
loop [while CorrectDestCondition != true or Exit = True]
4: Check = show(Error_Message) : True
4.1: fill(Destination)
4.2: click_on(OK_Button, Destination)
[Else]
5: getSafeAreas(Destination)
5.1: processing(Destination)
5.2: returnSafeAreas()
6: Screen = Show(Safe_Areas) : True
6.1: Select(SafeArea)
6.2: click_on(SafeArea_Button)
7: processing(SafeArea, UserPosition)
8: Screen = Show(RoutePath) : True

14 System model

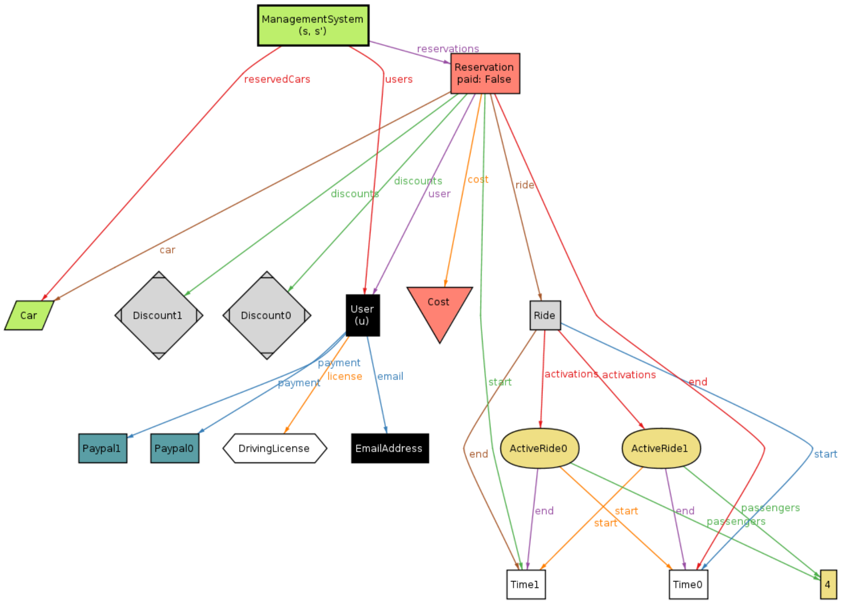
14.1 UML Class Diagram

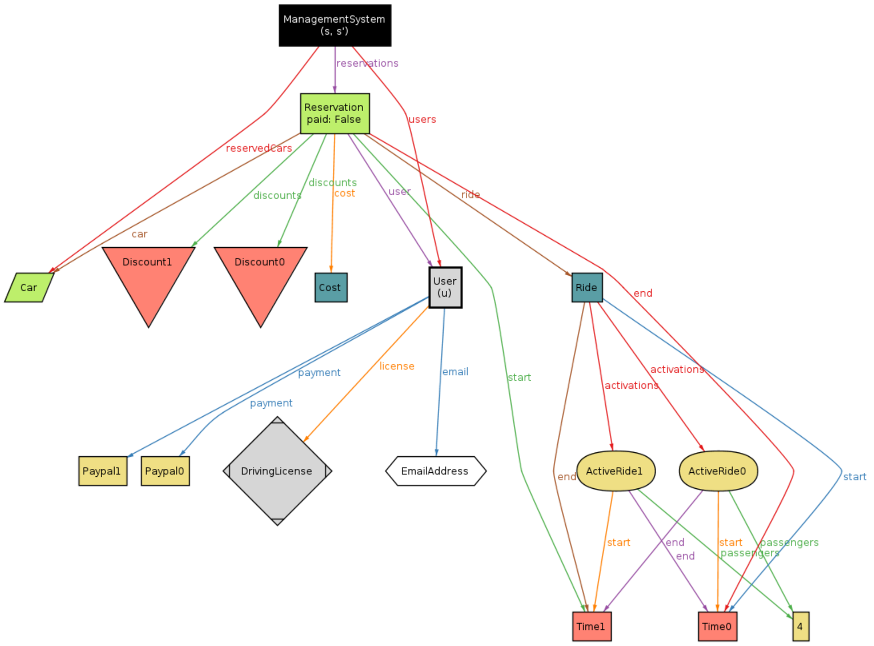


14.2 Alloy description

A simple Alloy model has been produced to describe the proposed system:

14.3 Examples of Alloy generated worlds


ManagementSystem
(s, s')
reservations
Reservation
paid: False
reservedCars
users
discounts
discounts
user
cost
ride
car
Car
Discount1
Discount0
User
(u)
Cost
Ride
payment
license
payment
email
start
activations
activations
end
start
Paypal1
Paypal0
DrivingLicense
EmailAddress
end
ActiveRide0
ActiveRide1
end
start
start
end
passengers
passengers
Time1
Time0
4


ManagementSystem
(s, s')
reservations
Reservation
paid: False
users
reservedCars
discounts
discounts
cost
user
ride
car
Car
Discount1
Discount0
Cost
User
(u)
Ride
end
payment
license
email
start
activations
activations
payment
Paypal1
Paypal0
DrivingLicense
EmailAddress
end
ActiveRide1
ActiveRide0
start
start
end
end
passengers
passengers
start
Time1
Time0
4