PowerEnJoy

## Requirement Analysis and Specification Document

Cattaneo Davide El Hariry Matteo Frontino Francesco

Contents

1. ***Introduction***
   1. Purpose
   2. Goals
   3. Definitions, acronyms, abbreviations
   4. Reference documents
   5. Document overview
2. ***Overall description***
   1. Product perspective
   2. Product functions
   3. User characteristics
   4. Constraints
   5. Assumptions and dependencies
   6. Future possible implementation
3. ***Specific requirements***
   1. External interface requirements
      1. User interfaces
      2. Hardware interfaces
      3. Software interfaces
      4. Communication interfaces
   2. Functional requirements
   3. Non-functional requirements
   4. Performance requirements
   5. Scenarios
   6. UML Models
      1. Use cases
      2. Class diagram
      3. State diagrams
4. ***Overall description***
   1. Alloy
      1. Signatures
      2. Facts
      3. Assertions
      4. Predicates
      5. Alloy model
      6. Generated world
   2. Software and tools used
   3. Team work
5. ***Revision***
   1. Changed assumptions
   2. Modified functional requirements
   3. Modified scenarios
   4. Modified use cases
   5. Modified diagrams
6. ***Introduction***
   1. ***Purpose***

With this project we will design an electric-car sharing software system.

Car Sharing is a very cost-effective and useful service for anyone who needs a car occasionally. It allows people to use and pay for the car according to their personal use, without the hassle and costs of owning their own vehicle (parking, purchase costs, maintenance, insurance etc.).

The system that we will develop is meant for cities which are provided with an efficient amount of parking lots and a wide distribution of electric car-charging platforms throughout the urban areas.

The application must allow the users which are registered to perform several easy and effective operations. Once logged in, the user can find available cars around him/her or in specified locations of the city, and chose the one to reserve.

Afterwards the user, who needs to reach the car before a given time slot expiration, will be able, by unlocking the car using the app, to easily enter the vehicle and drive to his/her destination.

* 1. ***Goals***

To create a throughout solution for the problem of providing a good and effective service to our users the application must allow the following:

•[G1] Guest users must be able to register to the system by providing their credentials and their payment information. They receive back a password that can be used to access the system.

•[G1] Users must be able to register and access to the system.

•[G2] Registered users must be able to find the locations of available cars within a certain distance.

•[G3] Users must be able to reserve a single car with a one-hour time limit.

•[G4] If a car is not picked‐up within one hour from the reservation, the system tags the car as available again, and the reservation expires; user will be charged with a fee of 1 €.

•[G5] A user that has reserved successfully a car must be able to unlock it using the app.

•[G6] As soon as the engine ignites, the system starts charging the user for a given amount of money per minute; the user is notified of the current charges through a screen on the car.

•[G7] The system stops charging the user as soon as the car is parked in a safe area and the user exits the car; at this point, the system locks the car automatically.

•[G8] The set of safe areas for parking cars is pre‐defined by the management system.

•[G9] system must know parking location, the battery level, status (in charge or not) and if there were two passengers onboard every time a ride is over in order to calculate the right discount.

* 1. ***Definitions, acronyms, abbreviations***

Here is a brief description of the most important actors and words used in our system:

* **User:** by user we mean a person already registered in the system, so that has a profile, uses the features provided by the system and perform actions accordingly. He can use all the functionalities described below (see Functional Requirements).
* **Guest:** a guest is a person that probably for the first time accesses the system or that hasn’t already signed up. Guest has less power in the system than a user, his functionalities are limited to access an introduction view and to register.
* **System:** is the application core. The software system which will perform all the operations and monitor interactions between users and cars.
* **Reservation:** the allocation of a car to a user, which starts when the booking request arrives and ends either when the expiration time ends or when the car is unlocked.
* **Car:** the vehicle used by the users, which contains different sensors and an embedded computer. It has seat sensors to detect if there are passengers, battery level and charging sensors. The computer of course has as main functionality to provide navigation through GPS system and to send all the relevant data to the main system server.
* **Ride:** conceptually is the use of the car, and it can be identified by the time duration of the user’s journey, from unlocking the vehicle until the final parking (having user selecting “end ride” or “end ride & charge” on the car screen) with the car locked.
* **Travel:** is considered as the ride segment from starting engine and car stop without ending definitely the ride. More travels can be part of a single ride.
* **Administrator:** the administrator of the system is the person allowed to manage eventual unexpected cases (like incidents and damaging situations).
  1. ***Reference documents***

Specification Document: Assignments 1 and 2 (RASD and DD).pdf

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

Examples documents:

RASD sample from Oct. 20 lecture

* 1. ***Document overview***

1. ***Overall description***
   1. ***Product perspective***

We propose a mobile application platform that will provide the public with the services described below.

Users will be able to reserve the car which better suites their location preference and enter it as soon as they unlock it using the app and reach it. Once the car is unlocked the system gives the user 1 minute to enter it before locking the car again. When the user is inside the vehicle the system start charging him/her with the halt-rate until he starts the engine (then the normal rate starts). The rate will keep changing according to the engine mode (on with normal rate/off with halt rate) until the user reaches his destination and exit the vehicle (car lock enabled).

User may also interact with the system by handling his own profile with different provided operations (top up wallet, manage coupons, send gifts etc.).

The administrator will be given a special module to manage specific and special situations.

* 1. ***Domain properties***

We suppose that the following conditions hold in the analyzed world.

* system must be able to store user information.
* GPS always give the right position.
* Cars shown to the users trying to book one are always and only the available ones.
* Available cars have always an active GPS.
* No two reservations for a single car occur on the exact same time.
* Car reservations which are completed can be cancelled within 10 minutes.
* One user can reserve only one car at a time.
* Reservation processes are addressed as time limited transactions.
* a booked car will always be found in the location provided at the reservation time within one hour.
* system must start the time counter when a booking confirmation is received.
* a booked car will always be found in the location provided at the reservation time.
* The only way to connect with the car to authenticate is by means of the smartphone app.
* The user has 1 minute of time to enter the car before it locks again.
* All parked and not in use cars are locked.
* A car must be unlocked only if it’s being used or if an authentication request is processed as successful.
* the car system is able, through the seat sensors, to know when a person is inside the vehicle.
* When a user enters a car the engine must be off.
* When a user enters a car the screen must be working.
* When the start button is pressed the car must turn on the engine.
* The charging fee unit is money per minute.
* In the system there is no scenario without safe areas for parking.
* the car system is always able to send information through mobile network to the main system.
* mobile networks are always providing a fair signal to all devices and computers of our system.
  1. ***Assumptions***

There below we specify how we assume the world works in ambiguous situations:

* The car is unlocked when the user inserts the correct PIN in the lock device, which is able to match the user profile with the one who reserved the car.
* During the period between the car opening and the car locking, in which the car engine is not running, the user will be charged with a halting rate, which will be smaller than the normal charge but useful in avoiding people from occupying the vehicle for free without using it.
* When dealing with safe parking evaluation, we assume that the onboard computer can identify each time if the parking has a match among the safe ones on its list.
* The information needed to estimate the discount for each ride, will be available to the system for each car. The system can get the battery level/status, car location and passengers number through sensors mounted in it. Then, knowing those parameters, it will calculate the eventual discount for the user.
* The discount will be applied to the user’s account only once the ride is finished and the car locked. This because the system needs to know whether the car after the usage has been plugged to the power grid or not and the exact location where it has been left.
* If the user has not put in charge the car in 40 seconds after the selection of the option “end ride and recharge”, the system is discarding that option, locking the car and applying the “end ride” action only.
* Fair vehicle distribution is not depending on PowerEnJoy and is left to users. PowerEnJoy just take care of having all available cars fully charged.
  1. ***Possible future implementations***
* The service could be extended to disabled people by adding special vehicles to the company’s set of cars.
* The service could be extended to electric motor-bikes.
* The service could allow the exchange of vehicles between cities in which PowerEnJoy operates.

1. ***Specific requirements***
   1. ***External interface requirements***
      1. ***User interfaces***
      2. ***Hardware interfaces***
      3. ***Software interfaces***
      4. ***Communications interfaces***
   2. ***Functional requirements***

-R1: system must be able to provide a registration form, in which the user must enter his/her personal and payment information in order to successfully register.

-R1: The correctness of debit or credit card information is evaluated by a specific external bank modules linked to our system.

-R1: system must verify the fiscal code and the email validity automatically and eventually send the user a password, which (s)he can use to log in.

-R5: system must be able to find all available cars within the distance defined by the user. This functionality is provided by the application itself.

-R5: user can choose the searching position from his own location (using the phone GPS) or from an address he manually inserts.

-R6: system must be able to indicate the vehicles in the perimeter range on the map.

-R7: for each indicated car on the map system will show its battery autonomy.

-R8: system will show on the map only cars which are available in the searching region defined by the user inputs.

-R8: system must hide a reserved car from the map until its release.

-R: a car release must occur when the one-hour period of time from the reservation of a car expires.

-R: a car release implies its reinsertion among the available ones.

-R: user must be able to reserve an available car shown by the system, by selecting it directly from the map or a list provided by the application.

-R9: system must hide a reserved car from the map until its booking expires.

-R10: a car must be released from a reservation when the one-hour slot of time expires. This is done by changing its status from “reserved” to “available”.

-R11: system must charge the user who books a car and lets the reservation expire.

-R12: The car system must be able to remotely unlock the car that’s been reserved when the user uses the app function “unlock car”.

-R13: The system locks the car after 1 minute from the user “unlock request” if (s)he does not enter in that period of time.

-R13: The car system must start charging the user when he/she starts the engine.

-R12: The charging fee must be displayed and kept updated throughout the car driving time.

-R13: The set of safe areas for parking must be defined before the system is being used for the first time.

-R13: The set of safe areas for parking must be defined before the system is being used for the first time.

-R13: The system must be flexible in order to adapt to the city in which it is deployed. In fact, it allows the administrator to set the number and location of safe areas for parking and to modify them.

-R13: The set of safe areas for parking must be defined before the system is being used for the first time.

-R13: In order for the system to evaluate properly an eventual discount for the user’s ride, it has to know the parking location, the battery level, status (in charge or not) and if there were two passengers on-board.

***Advanced Functional requirements***

1. If the system detects the user took at least two other passengers onto the car, the system applies a discount of 10% on the last ride.

D15: the car is provided with always functioning seat sensors.

-R13: seat sensors must be able to detect correctly a person on-board and notify the car system.

-R13: the car system must correctly evaluate the seat sensors notifications and eventually report a 10% discount on the current ride.

1. If a car is left with no more than 50% of the battery empty, the system applies a discount of 20% on the last ride.

D15: battery level sensors always work properly on each available car of the system.

-R13: battery sensors must be able to notify the car system about the battery level (in a percentage scale) in real time.

-R13: the car system must correctly evaluate the battery level notifications when the drive is over and eventually report a 20% discount on the current ride.

1. If a car is left at special parking areas where they can be recharged and the user takes care of plugging the car into the power grid, the system applies a discount of 30% on the last ride.

D15: special parking areas are predefined and stored in the database of the system.

D15: all special parking areas have plugs to charge the electric cars.

D15: battery plug sensor always work properly on each available car of the system.

-R13: battery plug sensor must be able to notify the car system about the battery status (plugged or not) when the car is parked.

-R13: system must be able to match if the parking where a car is located is part of the special parking area.

-R13: battery plug sensor must be able to notify the car system about the battery status (plugged or not) when the car is parked.

-R13: from the time a user ends a ride the system will detect for 1 minute if the vehicle is being plugged in charge, and in case it is not, it will lock the car and apply no discount.

-R13: car system must apply a 30% discount to the user when the car is parked in a safe area and it is plugged in charge. When it is plugged to the power grid without being in a safe area no discount is applied, but if the user didn’t end the ride (so (s)he is in a temporary stop) the system will stop charging him/her any rate.

1. If a car is left at more than 3 KM from the nearest power grid station or with more than 80% of the battery empty, the system charges 30% more on the last ride to compensate for the cost required to re-­‐charge the car on-­‐site.

-R13: car system must find the nearest power station to the cars which right after a ride are parked and are not connected to a charge grid. When the distance found is greater than three kilometres, or when in the same case the parked car is not being plugged and the battery is less than 20% level, the user must be charged 30% more on the last ride.

1. If the user enables the money saving option, he/she can input his/her final destination and the system provides information about the station where to leave the car to get a discount. This station is determined to ensure a uniform distribution of cars in the city and depends both on the destination of the user and on the availability of power plugs at the selected station.

D15: the system keep track along with the location of cars (distributed throughout the city) their status and their battery status and level (in-charge).

-R13: The cars provide a selectable ride options called “money saving”. This option is displayed by the car screen after the user enters the car. The user can run the saving mode in anytime from the driving start till the stop. The option allows the user to find the nearest charge station from the current location or from his/her destination.

-R13: The money saving option uses a special algorithm able to take into account the availability of power plugs at the nearest station at the searching time, in order to provide an accurate suggestion. The algorithm also, using the information about other cars’ status (in charge or not) in the preselected area, find the best station to suggest ensuring a uniform distribution.

***More domain assumptions:***

-the bonus related to the battery savings is to be applied at the end of a ride, after users confirm the ride is over. The eventual 20% discount is to be applied to the total cost of the entire ride.

-the 50% of the battery is considered dynamically on the total battery level available at time of booking. If one takes a car with little battery is because (s)he expects to drive for a short time, so it is normal that it is more difficult for him to get the discount.

- the 30 % fee is to be applied considering that the battery has to be left with more than 80% empty level of charge with respect to the charge level that the car had at the reservation time.

-a car can only be taken if the battery level is higher than 15% of the total charge

-under 15%, in case the car is not connected to a charging station, the system will notify an employee to go to recharge the car, possibly moving it

- a ride can be composed by several travels.

- travels are considered as the ride segment from starting engine and car stop without ending definitely the ride.

-the 2 passengers related bonus is calculated for each travel according to every single travel charge for that ride.

- the 2 passengers related bonus is calculated on the cost of each travel and it is assigned at the moment when the doors are closed, engine is running, and the seat sensors detect at least three passengers onboard (one is the driver and at least two passengers).

D: the car, for security reasons, turns on and remains on as long as all the doors are properly closed.

D: the passenger capacity is calculated by weight sensors placed under the seats which detects a mass greater than 20 kg

- for every user the system stores a list of his travels: among the various information for a single trip, the system saves an itinerary composed by a set of positions. Thanks to this routes information, we are able to handle infractions, road violations, fines etc.  so the costs of the eventual fines and law breaking behaviors can be assigned to the right user.

- fines received by users will only be forwarded to them, including a copy of the itinerary information. It is then up to the user to decide whether to pay the fine or otherwise. PowerEnJoy only duty is to inform the law enforcement by providing user data to them at the time of the infringement.

-A: car charges are handled in this way: when the user finishes a ride, the car is able to detect, through the system, its location within a charging station. If the user chooses the option "park & charge", the car queries the system which in turn sends the vehicle the number of the column to be used, enabling it for use within 40 seconds. The columns not enabled does not deliver current.

-A: areas are mapped as circles with Center and diameter

-D: one car can be disconnected from the charging column by an operator or by a user who has booked it and unlocked it.

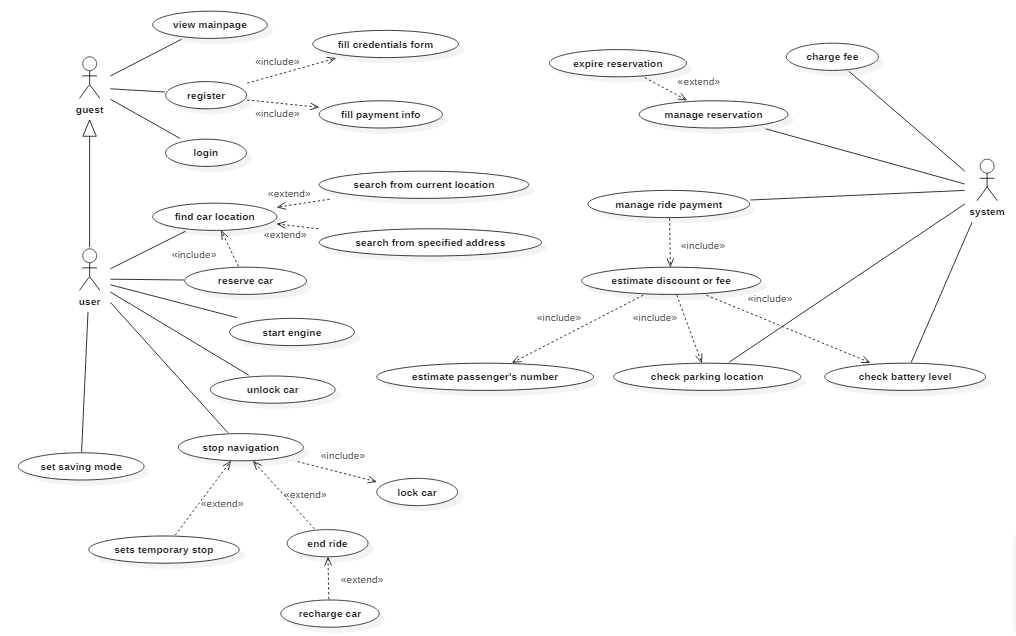
-D: users can park only in safe areas

-D: any damage that occurs the cars parked and plugged in charge are always related to the user who has placed in charge the vehicle. The recharge areas are under video surveillance to prevent vandalism acts from being charged as a user fault.

* 1. ***Nonfunctional requirements***
  2. ***Performance requirements***
  3. ***Scenarios***
  4. ***UML models***

Here below are provided the major UML diagrams that allow a concrete comprehension of how the system is going to operate.

* + 1. ***Use cases***



* + - 1. View main page

|  |  |
| --- | --- |
| *Actors* | Guest |
| *Goal* | Allow the visitor to collect information about PowerEnJoy |
| *Input condition* | - |
| *Event flow* | 1. The guest accesses PowerEnJoy web site 2. The guest visits web pages and collects information about the service 3. The guest decides whether to register or leave the web site |
| *Output condition* | - |
| *Exceptions* | - |

* + - 1. Register

|  |  |
| --- | --- |
| *Actors* | Guest, System |
| *Goal* | Allow the visitor to register to PowerEnJoy by submitting a form containing (s)he’s personal data and payment information |
| *Input condition* | - |
| *Event flow* | 1. The guest clicks “Register” in the web page 2. The guest fills and submits (s)he’s personal data and credit card number to the system 3. The system verifies user’s data and releases a password to use PowerEnJoy infrastructures |
| *Output condition* | Guest user is now a user and the system has provided him with a password to access the service |
| *Exceptions* | * User data are not valid * User credit card is not valid |

* + - 1. Login

|  |  |
| --- | --- |
| *Actors* | Guest |
| *Goal* | Allow the guest to fill his credentials and access the system becoming a registered user |
| *Input condition* | - |
| *Event flow* | 1. The guest clicks “Login” in the web page 2. The guest fills in him/her credentials 3. The guest presses the “Login” button |
| *Output condition* | The system sees the guest as user |
| *Exceptions* | * Username is invalid * Password is invalid |

* + - 1. Find car location

|  |  |
| --- | --- |
| *Actors* | User, System |
| *Goal* | Allow the user to find a car location |
| *Input condition* | The user is registered to PowerEnJoy and logged in |
| *Event flow* | 1. The user accesses PowerEnJoy app or web site 2. The user selects “Find a car” among all possible actions 3. The user submits the system (s)he’s location. Location may be specified by using user’s current position or by entering a valid address 4. The system looks for cars in the nearby of the specified positions having the status set as “AVAILABLE” |
| *Output condition* | User is provided with a map showing every available car in the nearby of the specified position. Every found car is ensured to be available. |
| *Exceptions* | * Location entered doesn’t exist * Impossible to detect user’s position |

* + - 1. Reserve car

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to reserve a car for 1h time |
| *Input condition* | The user is registered to PowerEnJoy, logged in and has already asked the system to find a car |
| *Event flow* | 1. The user selects one of the cars proposed by the system 2. The user presses the button “Reserve” 3. The car status is changed to “BOOKED” |
| *Output condition* | The selected car is reserved for up to 1h and the user receives a PIN to access the car during that time |
| *Exceptions* | * The user is trying to reserve more cars for the same period |

* + - 1. Unlock car

|  |  |
| --- | --- |
| *Actors* | User, System |
| *Goal* | Allow the user to unlock a reserved car and allowing the car access |
| *Input condition* | The car has been reserved and the user took less than 1h to use it |
| *Event flow* | 1. The user reaches the car 2. The user unlocks the car using the given PIN (entered in the car unlocking system) 3. The system checks whether the PIN received in input is correct 4. The user has access to the car 5. The system starts charging money to the user |
| *Output condition* | The car is unlocked and ready to start a ride |
| *Exceptions* | * The inserted PIN is invalid |

* + - 1. Set saving mode

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to get tips for saving money by applying virtuous behaviors |
| *Input condition* | The car is unlocked |
| *Event flow* | 1. The user presses the button “Saving mode” 2. The system tells the user which could be useful tips to save money at the end of the ride. The user is not bound to that tips and doesn’t necessarily follow them. |
| *Output condition* | - |
| *Exceptions* | - |

* + - 1. Start engine

|  |  |
| --- | --- |
| *Actors* | User, System |
| *Goal* | Allow to user to start the engine and begin his/her ride |
| *Input condition* | The car is unlocked |
| *Event flow* | 1. The user turns the key of the car 2. The system changes the car status from “PARKING” to “ONROAD” |
| *Output condition* | The car is unlocked and the ride has started |
| Exceptions | * Mechanical fault |

* + - 1. Stop navigation

|  |  |
| --- | --- |
| *Actors* | User, System |
| *Goal* | Allow to user to stop his/her navigation for a temporary of definitive stop |
| *Input condition* | The car is unlocked and the car status is set to “ONROAD” |
| *Event flow* | 1. The user chooses whether to park the car or to end his/her ride by clicking the appropriate button shown on the car screen 2. **A)** If the user chooses to park the car, the status is changed to “PARKING” **B)** If the user chooses to end the ride, the status is changed to “AVAILABLE” **C)** If the user chooses to end the ride and recharge the car (s)he has 40 seconds, after the engine is turned off, to plug it in a recharge station. When a car is recharging the status is set to “BATTERYCHARGE”. If the user take more than 40 seconds to plug in the car, the recharge option is discarded and the status is set to “AVAILABLE” 3. In cases **B** and **C**, a payment request is submitted to the system 4. The car is locked by the system as soon as the user: - gets off (cases 2 and 3) - takes more than 40 seconds to plug in the car in the recharge station - plugs in the car in the recharge station |
| *Output condition* | The car is parked, recharging or available |
| *Exceptions* | - |

* + - 1. Charge fee

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Charge the customer with a fee due to unused reservation |
| *Input condition* | More than 1h has passed after a user’s reservation and the car is in “BOOKED” status |
| *Event flow* | 1. The system verifies that more than 1h has passed after a user’s reservation 2. The system charges the user with a 1€ fee on his/her credit card 3. The status of the car is changed to “AVAILABLE” |
| *Output condition* | The user has been charged with a 1€ fee and the car in made available for new reservations |
| *Exceptions* | - |

* + - 1. Manage reservation

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Keep control on reservations aiming at maintaining usable the largest set of car possible |
| *Input condition* | - |
| *Event flow* | 1. The system checks users’ reservations and look for those cars that haven’t been unlocked in 1h time from booking 2. In case the reservation has expired, the system charges the user with a 1€ fee and changes the car status to “AVAILABLE” |
| *Output condition* | No car results booked after 1h time from reservation |
| *Exceptions* | - |

* + - 1. Manage ride payments

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Charge the user with the import generated from the car usage |
| *Input condition* | The car is locked and the user has chosen to end his/her ride |
| *Event flow* | 1. The system receives data about the ride directly from the car 2. The system verifies whether it is possible to apply a discount or an extra fee due to virtuous or unvirtuous behaviors (passengers number, charge level and parking location) 3. The system charges the user directly on his/her credit card with the calculated import |
| *Output condition* | The user has been charged with the correct amount of money and the transaction was completed successfully |
| *Exceptions* | - |

* + - 1. Check battery level

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Check the car level of charge |
| *Input condition* | The user has finished his ride and hasn’t put in charge the car |
| *Event flow* | 1. The system receives data directly form the car 2. The system checks the charge level of the car 3. If the car has to be recharged, it’s status is set to “RECHARGING” and a notification of action is sent to the 3rd party company that takes care of the recharge. Once the recharge process is complete, the car notifies the system, that changes the car status to “AVAILABLE” |
| *Output condition* | - |
| *Exceptions* | - |

* + - 1. Check parking location

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Check the car location |
| *Input condition* | The user has finished his ride |
| *Event flow* | 1. The system receives data directly from the car 2. The system checks the car location and verifies if it’s parked in a safe area |
| *Output condition* | - |
| *Exceptions* | - |

* + - 1. Send action request

|  |  |
| --- | --- |
| *Actors* | System |
| *Goal* | Send notifications of action to the 3rd party company that manages a fair distribution of vehicles and keeps charged all the cars |
| *Input condition* | The system verifies that a car needs an intervention |
| *Event flow* | 1. The system automatically sends a notification of action to the 3rd party company containing the car position and its license plate |
| *Output condition* | The car will be put in charge in 30 minutes |
| *Exceptions* | - |

* + 1. ***Class diagram***
    2. ***State diagrams***

1. ***Overall description***
   1. ***Alloy***
      1. ***Signatures***
      2. ***Facts***
      3. ***Assertions***
      4. ***Predicates***
      5. ***Alloy model***
      6. ***Generated world***
   2. ***Software and tools used***
   3. ***Team work***