PowerEnJoy

## Requirement Analysis and Specification Document

Cattaneo Davide Frontino Francesco El Hariry Matteo

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. Scenarios
   5. 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.

•[G5+] Once a user is inside the vehicle (s)he must be able to indicate eventual damages in the car.

•[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.
* **Operator:** is a flexible actor in our system. He’s part of a set of people operating under the administrator directions. Their normal tasks are to bring to charging stations cars left with less than 20% battery level, interact with users which call for help during a ride, intervene when necessary (e.g. a wheel brakes during a ride). Their exceptional task can be the cases in which …..
* **Administrator:** the administrator of the system is the person allowed to manage eventual unexpected cases (like incidents and damaging situations). He is the person notified every time a problem occurs, and once analyzed the situation he’ll decide how to handle it (call for support, send operators, call the police etc.).
  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.

Once the car is unlocked the system gives the user 1 minute to enter it before locking the car again. Once user is inside he must insert a PIN on the screen and signal eventual damages on the car using the touch display. If the PIN is inserted 3 times unsuccessfully the system will invite the user to exit the car and try another reservation. From 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 permanently exit the vehicle (car lock enabled).

User may also interact with the system by handling his own profile with different provided operations (check previous rides, check personal stats 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 make a reservation, 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.
* One user can reserve only one car at a time.
* User can only make a new reservation if there isn’t another one pending on him.
* After a reservation expires the user is allowed to make a new one.
* 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.
* Available cars can only have esthetical damages, that don’t compromise in anyways the usage of the vehicle according to our car-sharing system.
* Mechanic damages are considered all those problems the car may have which will negatively affect the driving and car usage in general.
* Mechanic damages can be electronically detected by our cars and though signalled to the main system in real time.
* The only way to unlock the car 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 is in use 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.
* The PIN code users must insert is generated with every reservation and is provided to users with the booking information on the app.
* When the start button is pressed the car must turn on the engine.
* The charging fee unit is money per minute.
* 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 and rationals***

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

* The car is unlocked when the user, after reserving the vehicle, uses the app function “unlock car”. This action will grant access to the associated car for up to 1 minute, time when the system locks the car again if nobody is detected inside.
* Once the user is inside the car, (s)he has to insert the PIN code provided through the app view in order to prove (s)he was the person who reserved that vehicle and opened it. This authentication method is intended to grant maximum security for our vehicles. In fact, if the PIN is wrongly inserted 3 times, the car will disable all functionalities (so that car won’t be moved) and invite the person to exit and try with another reservation.
* Users have the responsibility of esthetical damages, so it is their duty to signal if the car they reserved has any, otherwise the guilt of those eventual problems will automatically be charged on them.
* Once authenticated to the car system (PIN code correctly inserted), users might want to signal some damages they found, so they have to simply interact with the apposite interface. If no damage is reported but actually something is wrong, the cost of repairs will be charged to the last user who didn’t care about informing the system about the damage.
* During the period between the car opening and the car locking, in which the car has the person inside but the engine is not, 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.
* If during the driving some problem that concerns the car or its usage, users can call for help using the apposite button provided by the car screen. Once confirmed the help request, an operator will be put in charge of the situation and he’ll manage the situation.
* The system is provided with a list of safe parking areas.
* 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 the vehicle. 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 plugged the car in charge within 40 seconds from the selection of the option “end ride and recharge” through the car display, the system is discarding that option, locking the car and applying the “end ride” action only.

***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).
* The car, for security reasons, turns on and remains on as long as all the doors are properly closed.
* 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.
* 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.
* Areas are mapped as circles with Center and diameter
* One car can be disconnected from the charging column by an operator or by a user who has booked it and unlocked it.
* 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. ***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***

The car sharing service is made available only to those users which are in possess of a modern smartphone or tablet and satisfy the constraints expressed in chapter 3.1.3. No other physical device is permitted to unlock a car.

Each car has an embedded micro-computer aiming to assist the user and receive commands throughout a touch display.

Moreover, all vehicles are provided with a set of functioning sensors (seat weight sensors, GPS, door sensors and charge-plug sensor) useful to grant an efficient check of car statuses and conditions.

Each car is provided with a mobile communication technology platform allowing it to be permanently connected and reachable through the Internet. This connection is used to exchange data between the car and the system.

* + 1. ***Software interfaces***

The user must be in possess of the official PowerEnJoy app. The app is available to download on Google Play, Apple Store and Windows Store for every mobile device having respectively:

* Android 4.4 or above releases
* iOS 7.0 or above releases
* Windows Mobile 10

Accepting usage condition is essential to use the app. No other 3rd party software is officially supported nor allowed.

To access the PowerEnJoy site, a web browser supporting HTML 5/CSS 3 is needed.

* + 1. ***Communications interfaces***

Devices used to unlock and reserve a car must be connected to a telecommunication network. Thus, at least one of the following technologies must be supported:

* 3G
* HSDPA
* 4G

Wi-fi networks can also be used to unlock a car but a mobile communication modem is needed to download the app.

Devices which are not provided with at least one of the listed communication technologies cannot be used in any way to reserve and unlock a car.

Each car is provided with a mobile communication technology. Connection signal is guaranteed to be available 24/7.

* 1. ***Functional requirements***

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

* 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.
* the correctness of debit or credit card information is evaluated by a specific external bank modules linked to our system.
* 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.

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

* system must be able to find all available cars within the distance defined by the user. This functionality is provided by the application itself.
* user can choose the searching position from his own location (using the phone GPS) or from an address he manually inserts.
* system must be able to indicate the vehicles in the perimeter range on the map.
* for each indicated car on the map system will show its battery autonomy.
* system will show on the map only cars which are available in the searching region defined by the user inputs.
* system must hide a reserved car from the map until its release.

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

* 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, and choosing “reserve car” option.
* Reservation lasts one hour.
* system must hide a reserved car from the map until its booking expires.

•[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 €.

* 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”.
* a car release implies its reinsertion among the available ones.
* system must charge the user who books a car and lets the reservation expire with a fee.

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

* The car system must be able to remotely unlock the car that’s been reserved when the user uses the app function “unlock car”.
* The car stays unlocked for up to 60 seconds.
* The system locks the car after 1 minute from the user “unlock request” if (s)he does not enter in that period of time.
* Once the user is inside the car he has to insert the PIN code provided by the app, in order to prove his identity.

•[G5+] Once a user is inside the vehicle (s)he must be able to indicate eventual damages in the car.

* The display, once the PIN is correctly inserted, must show the user a view useful to indicate esthetical damages to the car.
* If the user points out one or more damages, the system admin will be notified and he’ll set the vehicle as in maintenance right after this user’s ride.
* Maintenance operators will take care of the damaged cars and prior user to the damage indication will be charged the costs of repairs.

•[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.

* The electric car engine ignites when the user presses the “start” button in the car dashboard.
* The car system must start charging the user when he/she starts the engine.
* The charging fee must be shown on the car screen and kept updated throughout all the driving time.

•[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.

* The set of safe areas for parking must be defined before the system is being used for the first time.
* The car system, thanks to seat sensors, can determine when the user has left the car.

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

* 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.

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

* 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.

•[G10] System must inform users about the possibility of getting discounts and about how to obtain cost reductions.

* All the information about discounts and car usage are written in an appropriate module openable from the app menu.

***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.

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

* Seat sensors must be able to detect correctly a person on-board and notify the car system.
* 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.

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

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

* Special parking areas are predefined and stored in the database of the system.
* All special parking areas have plugs to charge the electric cars.
* Battery plug sensor always work properly on each available car of the system.
* Battery plug sensor must be able to notify the car system about the battery status (plugged or not) when the car is parked.
* System must be able to match if the parking where a car is located is part of the special parking area.
* Battery plug sensor must be able to notify the car system about the battery status (plugged or not) when the car is parked.
* 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.
* 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.

* 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.

D: Money saving option can only be activated if battery level is below 90% of charge.

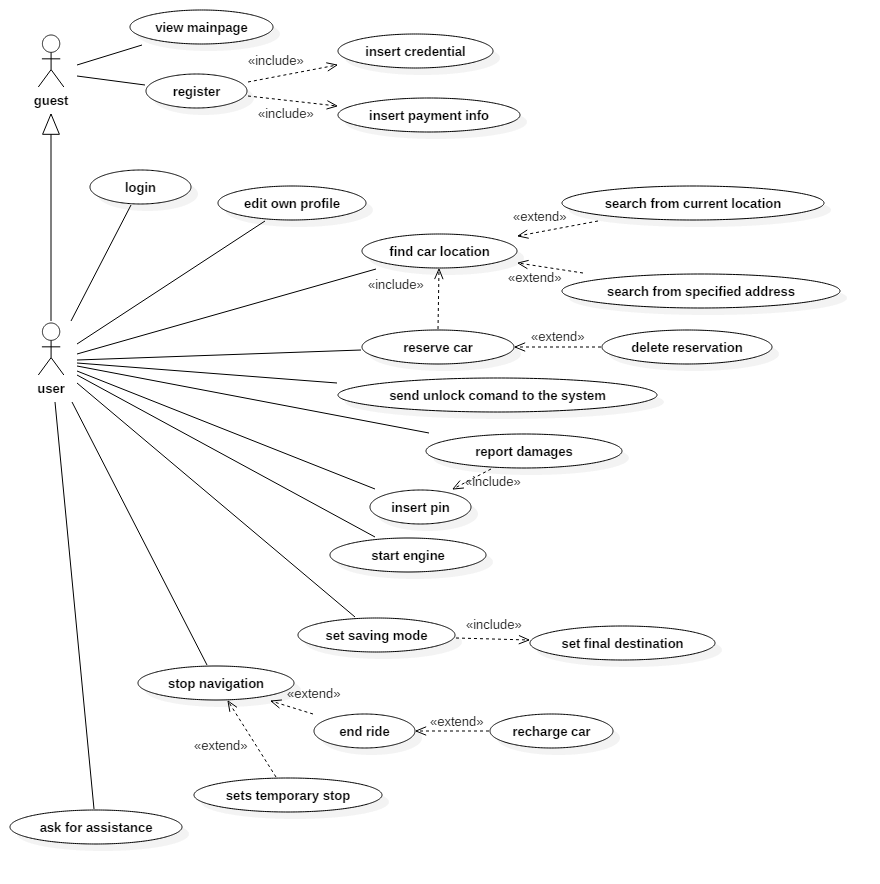
* The option will only be displayed on the car screen, so be selectable, if the battery level is less than 90%. This to ensure that cars which are quite full, so don’t actually need to be immediately charged, will not occupy charging plugs in the available stations.
* The system keep track along with the location of cars (distributed throughout the city) of their status and their battery status and level (in-charge).
* The cars provide a selectable ride option called “money saving”. This option is displayed by the car screen after the user enters the car. The user can start 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. The charge stations which best suits the searching parameters (location and distribution of other vehicles in charge) will be indicated in the navigation map on the screen.
* 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.
  1. ***Nonfunctional requirements***
  2. ***Scenarios***

Tim is a student visiting some relatives in the city. He is late for dinner and he needs to get from the hotel to his uncles’ home. Tim doesn’t want to grab a taxi because it will be too expensive for him, so he decides to use PowerEnJoy to drive to his uncles. After downloading and installing the app, he successfully registers and login. He immediately tries searching for a car near his hotel setting a distance of 200m. The app notifies Tim that no car is found in the area around that input, so he extends the distance to 400m, but still he gets no result. Tim can finally find an available car at 800 meters away, so he makes a reservation.

***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 case***



* + - 1. View main page

|  |  |
| --- | --- |
| *Actors* | Guest |
| *Goal* | Allow the visitor to collect information about PowerEnJoy and its service |
| *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 |
| *Goal* | Allow the visitor to register to PowerEnJoy by submitting a form containing his/her personal data and payment information |
| *Input condition* | - |
| *Event flow* | 1. The guest clicks “Register” in the web page 2. The guest fills and submits his/her personal data and credit card number to the system 3. The system verifies user’s data and releases a password to access PowerEnJoy infrastructures |
| *Output condition* | Guest 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* | User |
| *Goal* | Allow the user to fill his credentials and access the system |
| *Input condition* | - |
| *Event flow* | 1. The user clicks “Login” in the web page 2. The user fills in his/her credentials 3. The user presses the “Login” button |
| *Output condition* | The user is logged in |
| *Exceptions* | * Username is invalid * Password is invalid |

* + - 1. Edit own profile

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to edit his profile data. |
| *Input condition* | The user is logged in |
| *Event flow* | 1. The user choses “Edit profile” from his/her account options 2. The user modifies his/her personal data and presses “Confirm” 3. New data are processed by the system and, if correct, accepted |
| *Output condition* | User’s data have been modified |
| *Exceptions* | * Incorrect data inserted |

* + - 1. Find car location

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to find a car location |
| *Input condition* | The user is logged in |
| *Event flow* | 1. The user accesses the PowerEnJoy app 2. The user selects “Find a car” among all possible actions 3. The user submits the system his/her 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” or “RECHARGING” and a battery level of at least 15%. |
| *Output condition* | User is provided with a map showing every usable 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 logged in and has already queried 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. Send unlock command to the system

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to unlock a reserved car |
| *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 PowerEnJoy by pressing the button “Unlock car” 3. The user has access to the car up to 1 minute 4. The system starts charging money to the user |
| *Output condition* | The car is unlocked |
| *Exceptions* | * The 1 minute after unlocking expires and the user has not entered the car yet. * The system locks again the car. |

* + - 1. Insert PIN

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to unlock a make usable the car |
| *Input condition* | The car has been unlocked using the PowerEnJoy app |
| *Event flow* | 1. The user is asked to evaluate the car and report damages 2. The user enters the PIN |
| *Output condition* | The car is unlocked and ready for a ride |
| *Exceptions* | * Invalid PIN inserted |

* + - 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 “ONROAD” |
| *Event flow* | 1. The user presses the button “Saving mode” 2. The user inserts his/her final destination 3. 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 |
| *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 |
| *Goal* | Allow to user to stop his/her navigation for a temporary of definitive stop |
| *Input condition* | The car is “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 do a temporary stop, the car 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 takes 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. Ask for assistance

|  |  |
| --- | --- |
| *Actors* | User |
| *Goal* | Allow the user to receive assistance in case of problems or technical issues |
| *Input condition* | - |
| *Event flow* | 1. The user presses the button “Ask for assistance” on the car screen 2. The user is show some FAQ 3. If no FAQ is useful, the user can be put in contact with a PowerEnJoy operator |
| *Output condition* | - |
| Exceptions | - |

* + 1. ***Class diagram***
    2. ***State diagrams***
    3. ***Sequence diagrams***
    4. ***Activity 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***