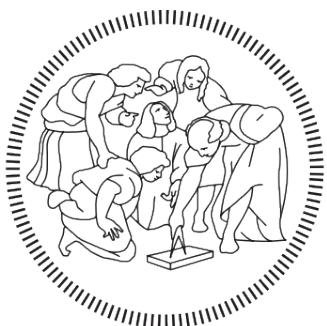


RASD - SOFTWARE ENGINEERING 2



POLITECNICO MILANO 1863

PowerEnjoy

Marini Alberto

862838

alberto2.marini@mail.polimi.it

Marrone Matteo

810840

matteo.marrone@mail.polimi.it

Sabatelli Antonella

875666

antonella.sabatelli@mail.polimi.it

November 13th, 2016

Politecnico di Milano

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Actors	4
1.4	Goals	5
1.5	Definitions, Acronyms, Abbreviations	6
1.5.1	Definitions	6
1.5.2	Acronyms	6
1.5.3	Abbreviations	7
1.6	Reference Documents	7
1.7	Document overview	7
2	Overall Description	8
2.1	Product perspective	8
2.2	User characteristics	8
2.3	Constraints	8
2.3.1	Regulatory policies	8
2.3.2	Hardware limitations	8
2.3.3	Interfaces to other applications	8
2.3.4	Parallel operation	8
2.4	Assumptions and Dependencies	9
2.4.0.1	Assumptions concerning the user	9
2.4.0.2	Assumptions concerning the cars	9
2.4.0.3	Assumptions concerning the operating space	9
2.4.0.4	Assumptions concerning payments and discounts policy	10
2.4.0.5	Assumptions concerning the rides	10
2.5	Future possible implementations	10
3	Specific Requirements	11
3.1	External Interface Requirements	11
3.1.1	User Interfaces	11
3.1.1.1	Download app	11
3.1.1.2	Homepage	13
3.1.1.3	Registration Form	14
3.1.1.4	Login	15
3.1.1.5	Personal Page	16
3.1.1.6	History	17
3.1.1.7	Info	18
3.1.1.8	Find a car	19
3.1.1.9	Marked map	21
3.1.1.10	Reserved Car	22

3.1.1.11 Reached Car	23
3.1.1.12 End ride	25
3.1.1.13 Navigation system	26
3.1.1.14 Money saving option	27
3.1.1.15 Battery warning	28
3.1.1.16 Battery empty	29
3.1.1.17 Boundaries warning	30
3.1.2 API Interfaces	31
3.1.3 Hardware Interfaces	31
3.1.4 Software Interfaces	31
3.1.5 Communication Interfaces	31
3.1.6 Memory	31
3.2 Functional Requirements	31
3.2.1 [G0] Allow a guest to visualize information regarding the service and become a registered user.	31
3.2.2 [G1] Allow a registered user to log in.	31
3.2.3 [G2] Allow logged in user to check the location of nearby cars.	32
3.2.4 [G3] Allow logged in user to reserve a car.	32
3.2.5 [G4] Allow logged in user to cancel its reservation until 15 minutes from the reservation time	32
3.2.6 [G5] Allow logged in user to open the car reserved . .	32
3.2.7 [G6] Carry out the transaction at the end of the ride, after the calculation of discounts and penalties.	32
3.2.8 [G7] Allow logged in user to ride the car.	33
3.2.9 [G8] Allow logged in user to enable money saving option when in the car	33
3.3 The World & the Machine	34
3.4 Scenarios	35
3.5 UML Models	36
3.5.1 Use Case: Guest registration	36
3.5.2 Use Case: Login	39
3.5.3 Use Case: Find available car	41
3.5.4 Use Case: Reserving a car	44
3.5.5 Use Case: Deleting a reservation	46
3.5.6 Use Case: Picking up a car	48
3.5.7 Use Case: Managing the car being driven	51
3.5.8 Class Diagram	54
3.5.9 State Machine Diagram	54
3.6 Non Functional Requirements	54
3.6.0.1 Maintainability	54
3.6.0.2 Portability	54
4 Appendix	55

Chapter 1

1 Introduction

1.1 Purpose

This document contains the **RASD** (*Requirement Analysis and Specification Document*), aimed to describe a car sharing system named PowerEnjoy which only employs electric cars, characterized by an AI developed by the car producer. No other previous software infrastructure exists. Conforming to the *IEEE-STD-830-1993* standard for RASD documentation this will include an overview of the functional and the non functional requirements of the system, showing the constraints and the limits of the software and simulating typical post-development scenarios through corresponding use cases.

This document is intended for all developers and programmers who have to implement the requirements, to system analysts who want to integrate other systems with this one, as well as for stakeholders interested in learning how the developed solution matches their request.

1.2 Scope

The project is aimed at creating a product that will allow end users to rent electric cars for any amount of time. Generic users should be able to register to the system by providing their credentials and payment information, receiving at the end of the process a password usable to access the system. Once accessed, the system should point to the users the locations of nearby cars available for reservation, which they'll be able to access within one hour after the reservation time. Once arrived on the site where the car is parked within the allotted time, a user should be able to prove its identity as the booker of the car, which would then be unlocked. The user should be charged while driving, in proportion to the time spent driving, until the car is parked in a safe area, whose set is pre-defined by the system. Once the user is out of the car, the system would lock the car and charge the user which would then be ready to be used by another user.

In case of expiration of the reservation, the user is charged of a pre-determined fee. Economical incentives and penalties should be awarded in proportion to how eco friendly the user's behaviour has been.

1.3 Actors

- **Guest:** a guest is a user who has yet to join the service; as such, it can only consult the information regarding the service available on

the homepage and navigate between it and the registration and the login pages. To become a registered user the guest will have to fill in a registration form and provide payment method data as well as a photo of its driving license; the system will then analyze the data and confirm the success of the registration process within 3 days.

- **Registered user:** a registered user is able to log in.
- **Logged-In User:** a user allowed to search for nearby cars and make and (within 15 minutes from its making) cancel a reservation for one of them at a time. After the reservation it will be enabled to open the reserved car.
- **Car AI:** through its sensors, the car detects information regarding the ride that will be sent to the system, unlocks and locks itself and displays to the user infos about the surrounding area.
- **Power Enjoy System:** the Power Enjoy System primarily keeps a record of the users, the cars and the reservations updating the status of each of them according to the input from the users and the cars AIs. It also calculates the price for each ride and coordinates itself with payment systems interfaces to finalize money transfers, and warns the staff about exceptional situations such as ones where a car has been damaged, or abandoned in a discharged state.

1.4 Goals

List of the goals of PowerEnjoy application:

- [G0] Allow a guest to visualize information regarding the service and become a registered user.
- [G1] Allow a registered user to log in.
- [G2] Allow logged in user to check the location of nearby cars.
- [G3] Allow logged in user to reserve a car.
- [G4] Allow logged in user to cancel its reservation until 15 minutes from the reservation time.
- [G5] Allow logged in user to open the car reserved.
- [G6] Carry out the transaction at the end of the ride, after the calculation of discounts and penalties.
- [G7] Allow logged in user to ride the car.
- [G8] Allow logged in user to enable money saving option when in the car.

1.5 Definitions, Acronyms, Abbreviations

1.5.1 Definitions

- **Pairing Code:** A code provided to the portable device by the system after the reservation, to be used to open the car exploiting bluetooth technology. When the car is reached by the user its portable device and the car join what is called a trusted pair through a discovery and authentication process where the matching of the code in possession of car and device is verified, and at the end of the process the car opens.
- **Blocked User:** a registered user who is able to access the service yet unable to reserve a car since the money transfer related to the last transaction failed.
- **Safe Area:** An area where the car can be parked without being subject to further time-based charging. It corresponds to a collections of parking spots in the city of Milan reserved by the service for the users.
- **Grid Station:** A safe area where it is possible to plug an electric car.

1.5.2 Acronyms

- **RASD:** Requirements Analysis and Specication Document
- **DB:** DataBase
- **DBMS:** DataBase management system
- **API:** Application Programming Interface
- **OS:** Operating System
- **JVM:** Java Virtual Machine
- **JEE:** Java Enterprise Edition
- **AI:** Artificial Intelligence
- **GPS:**Global Positioning System
- **MB:** Megabyte
- **MSO:** Money Saving Option

1.5.3 Abbreviations

1.6 Reference Documents

- Assignment document: AA 2016-2017 Software Engineering 2 -Project goal, schedule, and rules
- ISO/IEC/ IEEE Std 29148:2011(E) Systems and software engineering - Life cycle processes - Requirements engineering

1.7 Document overview

This document is divided in four sections:

- Section 1: Introduction, it consists of a generic introduction of the document aimed at providing the basic info necessary for the comprehension of the other sections contents.
- Section 2: Overall Description, gives general information about the software product with a special focus on constraints and assumptions.
- Section 3: Specific requirements, this part mainly includes different representations of functional requirements and also nonfunctional requirements.
- Section 4: Appendix, it contains a modelization of the requirements created by using alloy language.

Chapter 2

2 Overall Description

2.1 Product perspective

The chosen solution is a web-based application meant for portable devices, which will take up to 20MB of free space on the device memory and will make use of Bluetooth technology for the car unlock mechanism as well as of GPS technology to detect the car closest to the user. Only interfaces to external systems of financial institutions and country authorities meant to verify the existence of the person associated to the document provided will be developed, since, as stated before, no other previous software infrastructure exists (bar the car AIs) and the the application will be self contained.

2.2 User characteristics

We expect our user pool to be comprised of rather young individuals (age varying between 18 and 35), owning a driving license but still not in possession of a car and/or with a preference for electric cars due to environmental reasons. The user is expected to be used to exploiting mobile apps and should not require a particularly user-friendly interface.

2.3 Constraints

2.3.1 Regulatory policies

- As previously explained, to utilize the service a user must provide proof of its driving capability. No other legislation binding car sharing services exists.

2.3.2 Hardware limitations

No hardware limitations exist.

2.3.3 Interfaces to other applications

Interfaces to the systems of financial institutions responsible for the money transfers connected to the transactions are to be developed.

2.3.4 Parallel operation

PowerEnjoy must support parallelism since it is bound to face reservation and access to the service requests from multiple users at once on a 24 hour basis.

2.4 Assumptions and Dependencies

2.4.0.1 Assumptions concerning the user

- Users do not seek to trick the sensors of the car to trigger the application of a discount (for example carrying around weight to mimic the presence of passengers).
- No user who reserves a car and actually reaches it does not inform the system about damages seemingly previously incurred by the car, even when though damaged it is still possible to drive it.
- Every user is held responsible for the damages incurred by the car driven and accountable for every fine received while driving it, being informed of this policy within the terms of use.
- The user never approaches another car that he didn't have reserve.
- Users always provide an address corresponding to their actual location.
- The user either plugs the car within 5 minutes from getting out of the car or he doesn't plug it.

2.4.0.2 Assumptions concerning the cars

- All the cars used are identical.
- Cars AI source of power is not the same as the engines.
- Cars' doors are locked while the engine is turned on.
- If the engine is turned on, the locking mechanism turns itself off only in case of accident.
- The car doors close automatically if the user, after having gotten out of the car, does not do it within 10 seconds.
- A car undergoing charging is not displayed whenever a search is carried out by the user, if its battery level is below 75%.

2.4.0.3 Assumptions concerning the operating space

- The service is presently active in Milan and nowhere else. Attempts to lead the car beyond the city boundaries will succeed, yet no safe areas nor grid stations are defined outside of Milan.
- Safe areas are evenly distributed around the city.
- Every grid station is a safe area.
- In Milan there is at least a grid station per 5 km area.

2.4.0.4 Assumptions concerning payments and discounts policy

- The system checks the validity of the payment method provided by the user only once during the registration process.
- If a transaction is not carried out successfully, the user is blocked and becomes unable to reserve other cars until the debt to the service is repaid.
- The service calculates the price of a ride and charges the user only after 5 minutes it has left the car, so to allow him to plug the car and exploit the potential related discount.
- The user can visualize from the map how much battery is left in each car found before reserving one.

2.4.0.5 Assumptions concerning the rides

- Every user has 30 seconds to enter the car it has reserved once the lock has been lifted.
- A ride is deemed ended when the user parks in a safe area, the engine is turned off and the user has got out of the car. If the engine is turned on again before the user has got out of the car when that is parked in a safe area the counter restarts using the ride price previously calculated (that is the amount previously shown on the car display up to a second before parking) as zero value.
- It is thus not possible to stop the car, exiting, closing the car doors and reentering the car and then keep driving under any circumstance.
- When the car battery fall to 5% a warning.
- If a car battery falls to 0% while the car is being driven and the car stops outside of a safe area as a consequence, the ride will be deemed concluded, but the driver will be held responsible for any accident or disturbance caused not having properly recharged the battery before the beginning of the ride and will be expected to move the car to a suitable location through any means, failure in doing so resulting in a violation of the terms of use.

2.5 Future possible implementations

Chapter 3

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 Download app This page represents the official page inside the app store, from where an user can download our PowerEnjoy app.



Figure 1: Download app

Before the installation procedure starts, PowerEnjoy app needs the users permission to access to some functionalities, like *GPS position*, *Bluetooth*, *Camera* and *Camera Roll*.

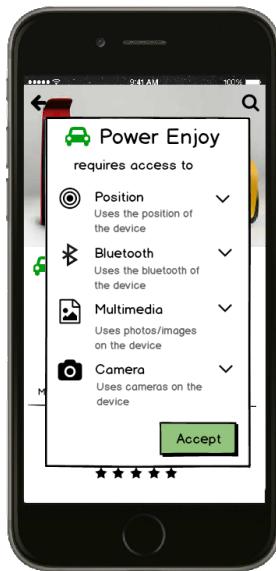


Figure 2: Download app permissions

3.1.1.2 Homepage Starting from the homepage, the user can easily register to the PowerEnjoy service or login.



Figure 3: Homepage

3.1.1.3 Registration Form In the registration form, the user must complete all the mandatory fields, accept the use term and upload a driving license picture.



Figure 4: Registration page 1

If one of the previous conditions won't be satisfied, the user must repeat the registration procedure.



Figure 5: Registration page 2

3.1.1.4 Login Inside the login page, the user can easily login using a username and a password.



Figure 6: Login page

3.1.1.5 Personal Page This page can be seen after the login. The user can find statistics, a button for checking previous rides and a button to look for a new car.

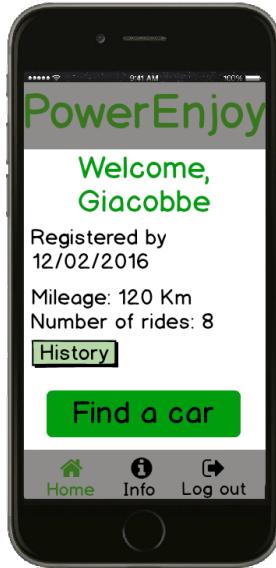


Figure 7: Personal page

3.1.1.6 History Inside the history page, a user can check informations about previous rides, in chronological order.



Figure 8: History 1

By selecting one element from the list, the app will provide many informations about the ride, including date, starting time, arriving time, final destination address, time, distance, total cost, discounts and penalties.



Figure 9: History 2

3.1.1.7 Info By pressing the 'Info' button placed on the task bar, the user can check the how discounts and penalties are applied.

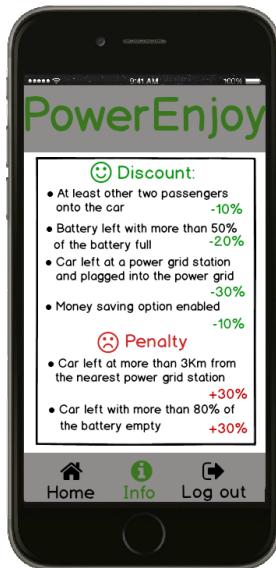


Figure 10: Info button shows penalties and discounts

3.1.1.8 Find a car From the Personal Page, by clicking on the 'Find a car' button, PowerEnjoy app will show all the cars available depending on user's position or inserted address (when a user doesn't give any permission for using the localization system).



Figure 11: Find a car button

In this map, the user can see his position and all the available cars in 1km radius.



Figure 12: Find a car 2

By selecting a car, the user gets all the info and can reserve it. After pressing the 'Reserve' button, a popup will be shown to remember that the user has 1 hour for reaching the car and 15 minutes for canceling the reservation.

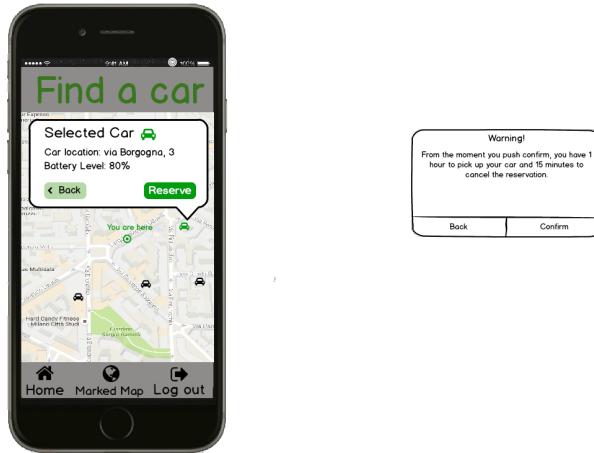


Figure 13: Book a car and popup

3.1.1.9 Marked map From the "Find a car" page, by pressing on "Marked Map", Power Enjoy app shows a map where there is the safe area and all the power grid stations. The user can also insert an address and take a look if it's inside or not a safe area.



Figure 14: Marked map

3.1.1.10 Reserved Car The Reserved Car page is shown when a user has booked a car. The "Cancel" button is activated only for 15 minutes, after the booking.



Figure 15: Reserved car

Here is represented the case when a user doesn't reach the car in one hour, starting from the booking time.

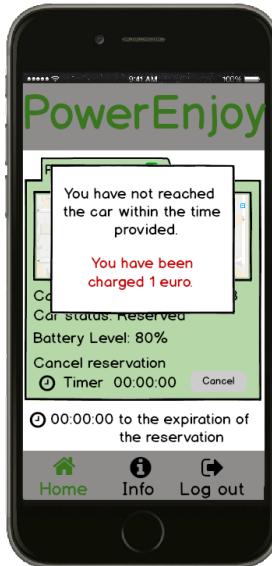


Figure 16: Reserved car not reached by the user in one hour

3.1.1.11 Reached Car The user has reached the booked car and the pairing process is going to start.



Figure 17: Reached car

The pairing procedure is concluded and the user can unlock the car or send a damage notification to the system.

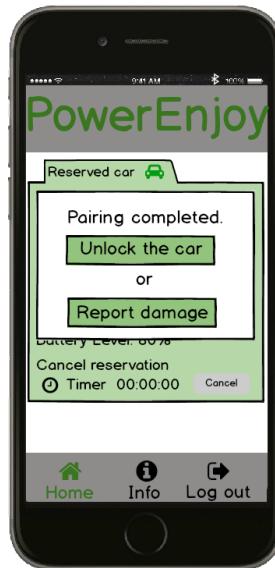


Figure 18: Reached car and pairing process concluded

After pressing "Report damage" button, the user can take photos of the

car. When the damage notification procedure is completed. the user will be redirect to the "Find a car" page, where he can see the nearest available car.

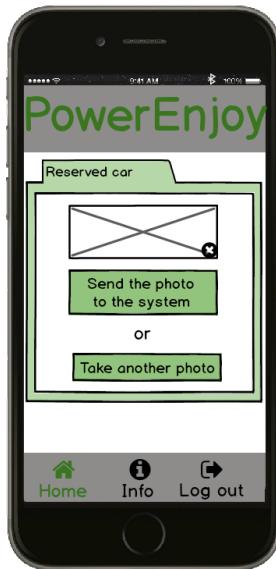


Figure 19: Send photos about damage

3.1.1.12 End ride Page shown to a user when the destination has been reached.

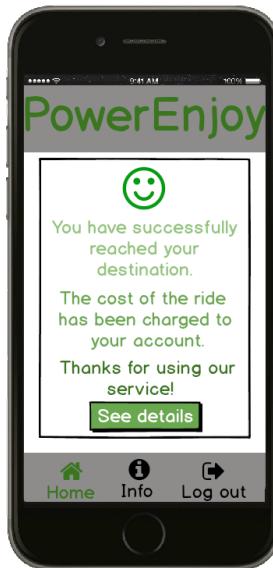


Figure 20: End ride

Once the user comes out from the car, by pressing on "View details" button, PowerEnjoy app will show all infos about discounts, penalties, date, stating time, ending time and price.

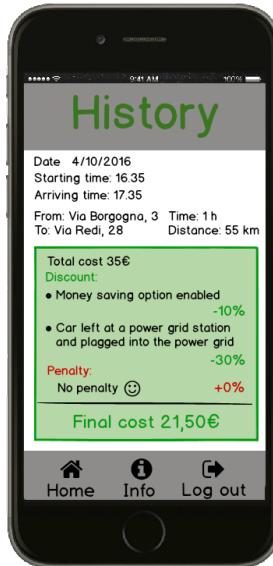
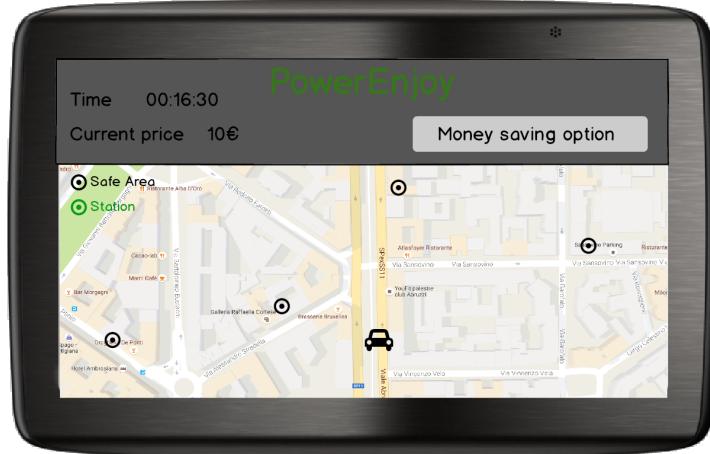


Figure 21: Send photos about damage

3.1.1.13 Navigation system This interface can be seen by the user while he is driving. Time, map and price are updated in real time. The map also shows where is the car and where is a safe area.

The "Saving money option" button is disabled, because it can be activated only before starting a ride.



3.1.1.14 Money saving option Before starting the ride, the user can activate the MSO from the navigation device. If MSO button will be selected, the user should insert the final destination address.



The navigation system will show to the user a map with arrows pointing to the closest power grid station, to a given address.



Figure 22: Money saving option

3.1.1.15 Battery warning The battery warning is shown on the car's GPS display when the car's battery is under 5%.

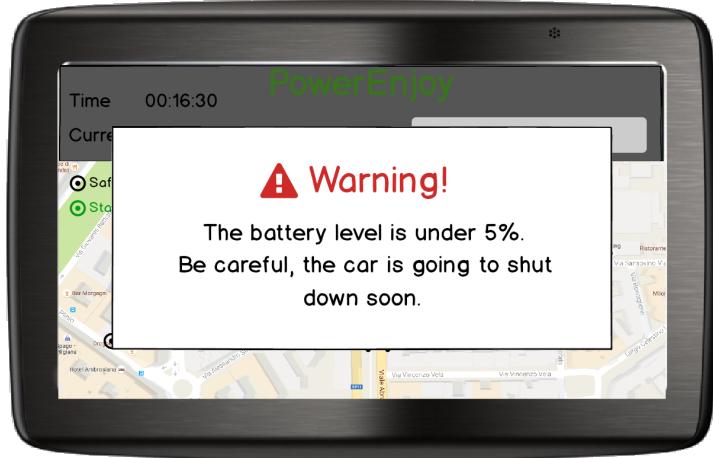


Figure 23: Battery warning

3.1.1.16 Battery empty The battery empty is shown when the car's battery is totally empty.

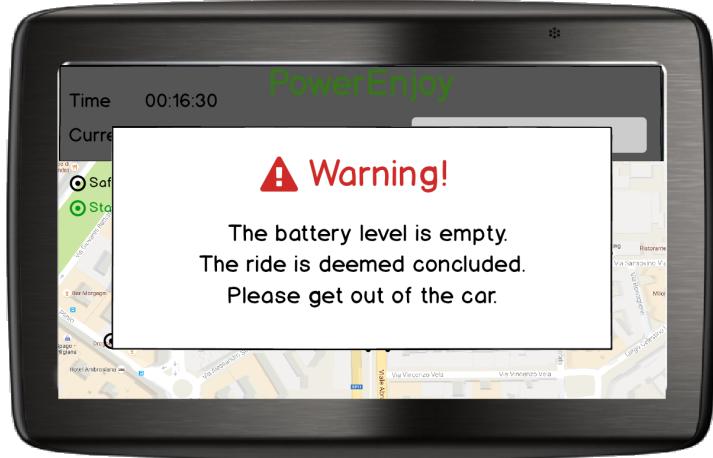


Figure 24: Battery empty

3.1.1.17 Boundaries warning The boundaries warning is shown, when the car's GPS understands that the user is driving outside the PowerEnjoy service area.

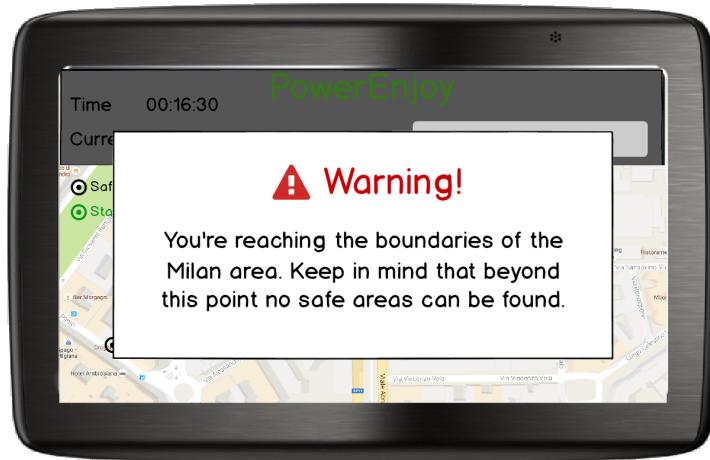


Figure 25: Boundaries warning

3.1.2 API Interfaces**3.1.3 Hardware Interfaces****3.1.4 Software Interfaces****3.1.5 Communication Interfaces****3.1.6 Memory****3.2 Functional Requirements****3.2.1 [G0] Allow a guest to visualize information regarding the service and become a registered user.**

- [R0] Until he has properly joined the service a guest can but access the registration and log in interfaces, and visualise important info related to the service.
- [R1] A registered user cannot join a second time.
- [R2] To carry out successfully the registration, the guest cannot choose a username already chosen by another user.
- [R3] To carry out successfully the registration, the guest has to provide a valid credit card number.
- [R4] To carry out successfully the registration, the guest has to provide a CVV matching the credit card number inserted.
- [R5] To carry out successfully the registration, the guest has to insert driving license data matching the one inferred from the driving license photo as inferred by the image analysis software.
- [R6] To carry out successfully the registration, the guest has to insert data related to a driving license that a proper authority acknowledges as actually existing.

3.2.2 [G1] Allow a registered user to log in.

- [R0] The user must be already registered to log in.
- [R1] The user must insert a correct username-password couple.

3.2.3 [G2] Allow logged in user to check the location of nearby cars.

- [R0] At least one car must be located within a 1 km radius from the user location.
- [D0] Users always provide an address corresponding to their actual location.

3.2.4 [G3] Allow logged in user to reserve a car.

- [R0] The user has to have carried out a search for nearby cars.
- [R1] The user must not be blocked.

3.2.5 [G4] Allow logged in user to cancel its reservation until 15 minutes from the reservation time

- [R0] The user has to have reserved a car.
- [R1] No more than 15 minutes can have passed since the reservation was made.

3.2.6 [G5] Allow logged in user to open the car reserved

- [R0] The user must have reserved a car and received the pairing code.
- [R1] No more than an hour can have passed since the reservation was made.
- [R2] The car has to not have been damaged.
- [D0] A user who realizes the car reserved has been damaged always reports the damage, even if the car is driveable.
- [D1] Every user has 30 seconds to enter the car it has reserved once the lock has been lifted.

3.2.7 [G6] Carry out the transaction at the end of the ride, after the calculation of discounts and penalties.

- [R0] The user must have exited the car.
- [R1] Five minutes have to have gone by since the user did.
- [D0] Users do not seek to trick the sensors of the car to trigger the application of a discount (for example carrying around weight to mimic the presence of passengers).

3.2.8 [G7] Allow logged in user to ride the car.

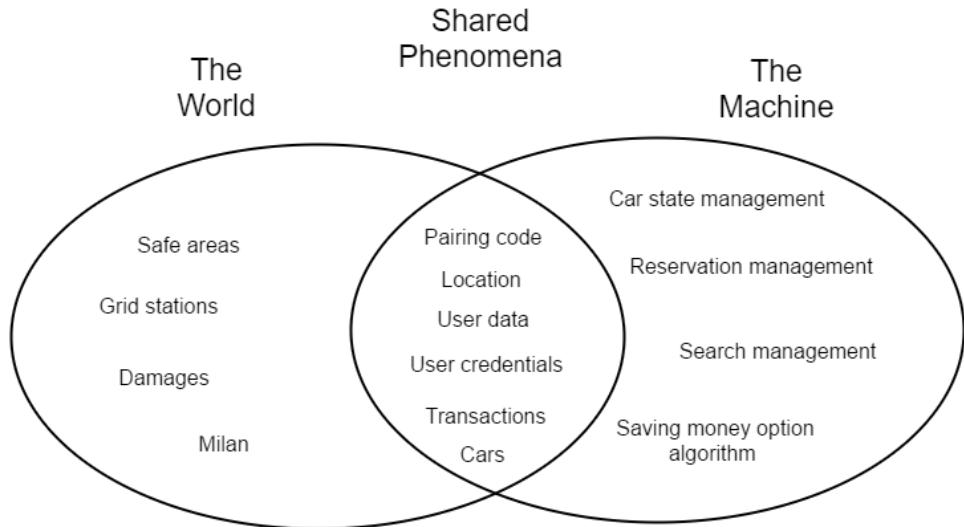
- [R0] The user has to have opened the car.
- [R1] the user has to open the car door within 30 seconds from having unlocked the car.
- [R2] The user has to ignite the engine.
- [R3] The user has to park in a safe area, otherwise the ride will not be deemed concluded.
- [D0] A ride is deemed ended when the user parks in a safe area, the engine is turned off and the user has got out of the car. If the engine is turned on again before the user has got out of the car when that is parked in a safe area the counter restarts using the ride price previously calculated (that is the amount previously shown on the car display up to a second before parking) as zero value.
- [D1] It is thus not possible to stop the car, exiting and reentering the car and then keep driving under any circumstance.

3.2.9 [G8] Allow logged in user to enable money saving option when in the car

- [R0] The user has to have entered the car
- [R1] The user has to have closed the car doors.
- [R2] The user has to insert an existing destination.

3.3 The World & the Machine

In this section we present a schematic domain analysis of the application using the *The World & the Machine* model by M.Jackson & P.Zave. The world contains the physical elements to be considered within the development process, whereas the machine includes the software ones, the shared phenomena being the intersection between the two and containing consequently entities such as the cars, physical yet provided with software that enables them, for example, to calculate the shortest path to the destination within the MSO context.



3.4 Scenarios

3.5 UML Models

We decided to represent the actor "System" in the UCDs in spite of it being unnecessary to allow for an easier comprehension of the content.

3.5.1 Use Case: Guest registration

Table 1: Guest registration

Participating actors	Guest
Entry condition	The guest is not registered into the system
Flow of events	<ol style="list-style-type: none"> 1. The guest downloads the app from the app store on his device, gives permission to access Bluetooth, location info and camera and opens the app. 2. The guest clicks on the register button and enters the registration page. 3. The guest provides his credentials, a photo of his driving license, his payment information, accepts terms of use and confirms. 4. The system checks the driving license and payment information. 5. The system sends the password to the guest and informs the guest its registration has ended successfully.
Exit condition	The guest is now a registered user.

Exceptions	<ul style="list-style-type: none">• The guest doesn't give the permission to access Bluetooth, location info and camera.• The app is not downloaded.• The credentials are not correct.• The driving license is not accepted.• The payment information are not correct.• To be filled fields are not filled.• All the exceptions are handled requesting the guest to continue the registration process from the point where incorrect info were provided or critical fields were left blank.
------------	---

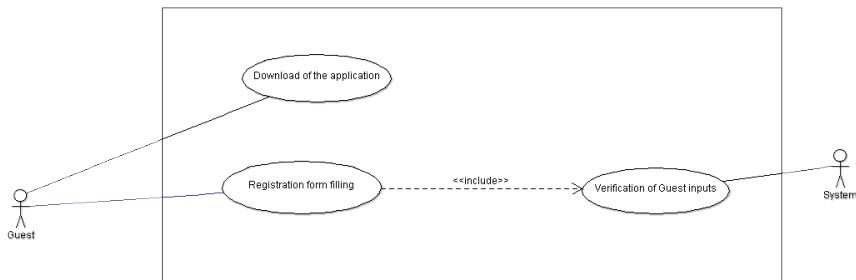


Figure 26: Guest registration use case diagram

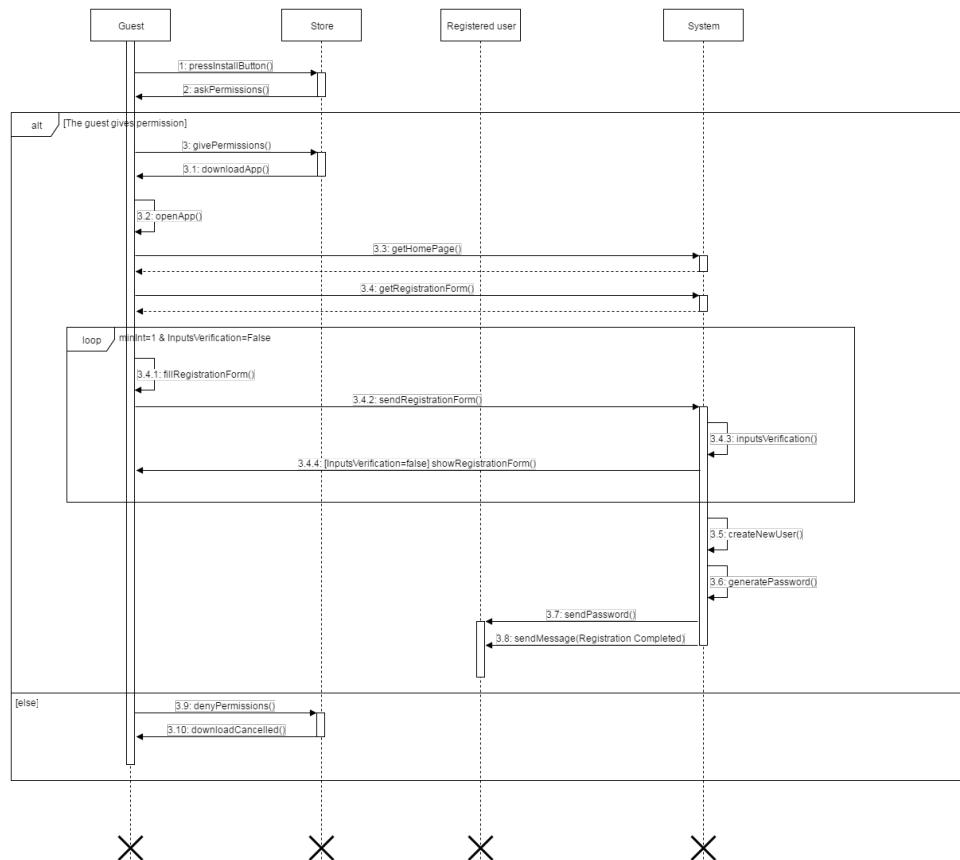


Figure 27: Guest registration sequence diagram

3.5.2 Use Case: Login

Table 2: Login

Participating actors	<ul style="list-style-type: none"> • Guest • Registered user • System
Entry condition	The visitor has already joined the community
Flow of events	<ol style="list-style-type: none"> 1. The guest inserts the username-password couple associated to him. 2. The system checks the couple username and password and allows the guest to log in as registered user if the username-password couple is found in the database.
Exit condition	The user is logged in and redirected to his personal page.
Exceptions	<ul style="list-style-type: none"> • The couple username and password is not found in the database and the login fails. • The app notifies the guest and redirects him to the login page.

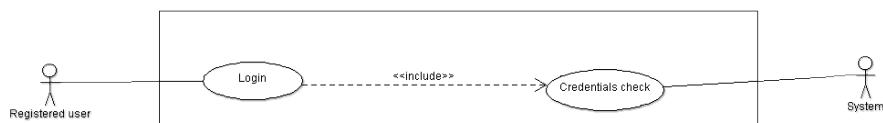


Figure 28: Login use case diagram

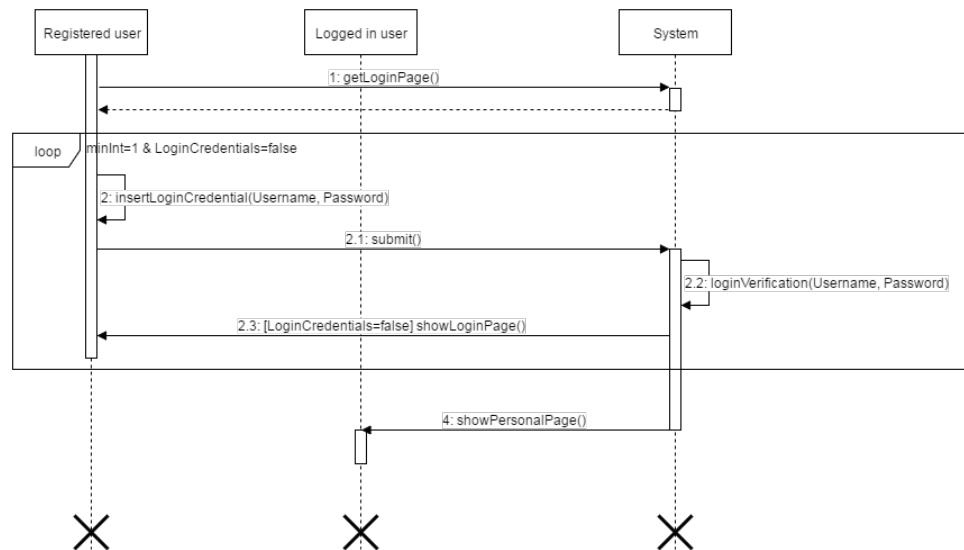


Figure 29: Login sequence diagram

3.5.3 Use Case: Find available car

Table 3: Find available car

Participating actors	<ul style="list-style-type: none"> • Logged in user • System
Entry condition	The user is logged in
Flow of events	<ol style="list-style-type: none"> 1. The user selects the option Find car. 2. The system searches for available cars within 1 km from the location derived by GPS localization or specified explicitly by the user.
Exit condition	The cars closest to the user are visualized on the map.
Exceptions	<ul style="list-style-type: none"> • The user has not yet given his consent to be automatically localized. • Both the options to give the address corresponding to his current location or be localized through GPS are displayed. • There are not available cars within 1km. • The system states that no available car is reasonably close. • The payment information are not correct.

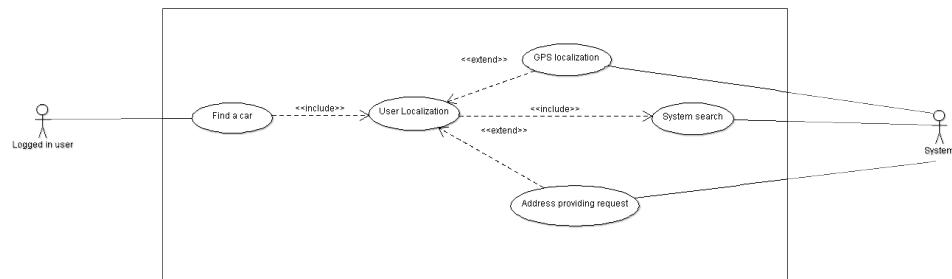


Figure 30: Find available car use case diagram

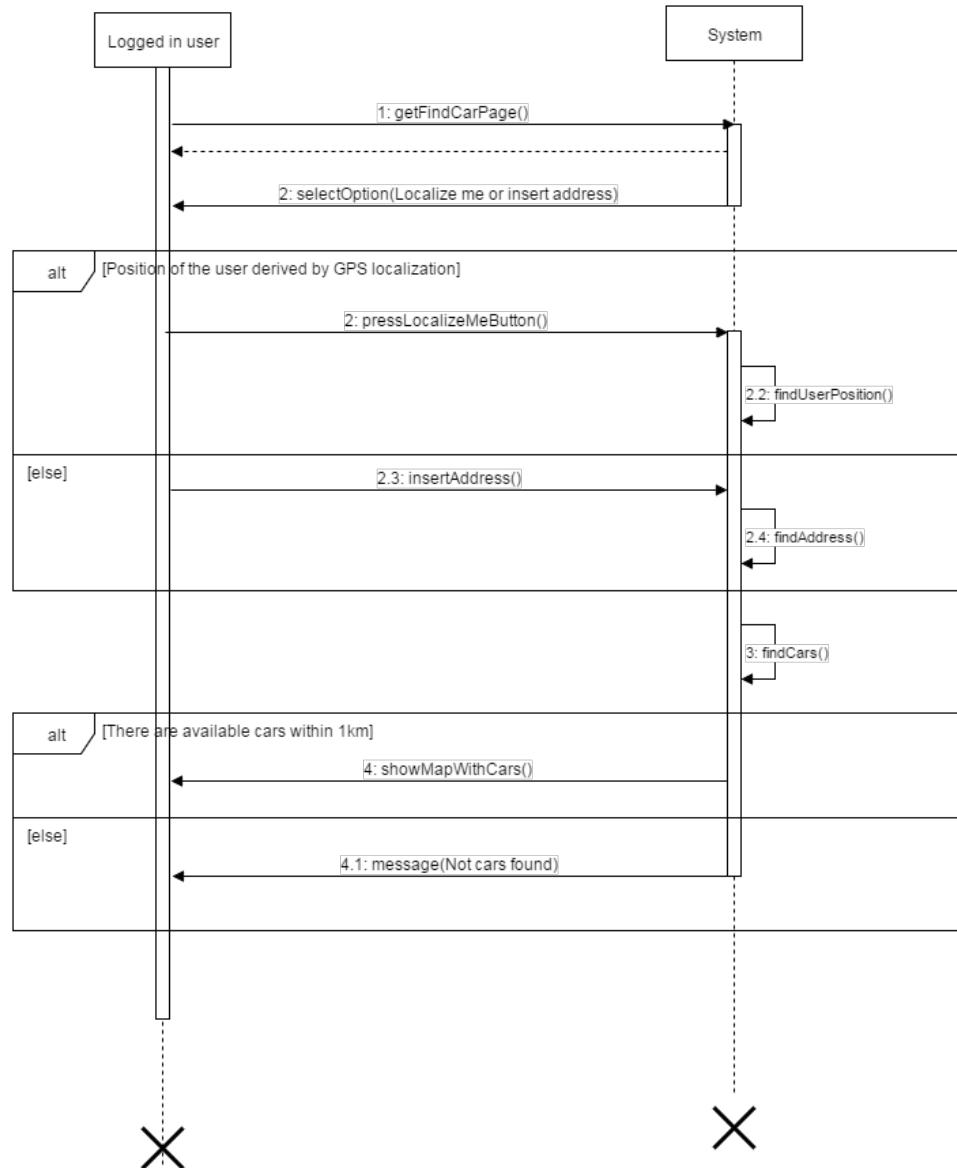


Figure 31: Find available car sequence diagram

3.5.4 Use Case: Reserving a car

Table 4: Reserving a car

Participating actors	<ul style="list-style-type: none"> • Logged in user • System
Entry condition	A search for available cars has just been conducted.
Flow of events	<ol style="list-style-type: none"> 1. The user chooses a car from the map. 2. The system reserves the car for the user, who is given a pairing code matching with the one set up at the same time for the car. 3. Two timers are initialized by the system: the user has now 15 minutes to cancel the reservation without suffering charges and 1 hour to pick up the car.
Exit condition	The user has now a reservation.
Exceptions	<ul style="list-style-type: none"> • The car selected is reserved by another user in the interval between the visualization and the selection. • The user is asked to select another car.

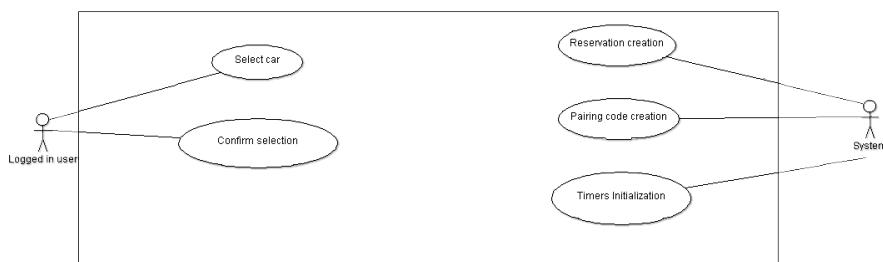


Figure 32: Reserving a car use case diagram

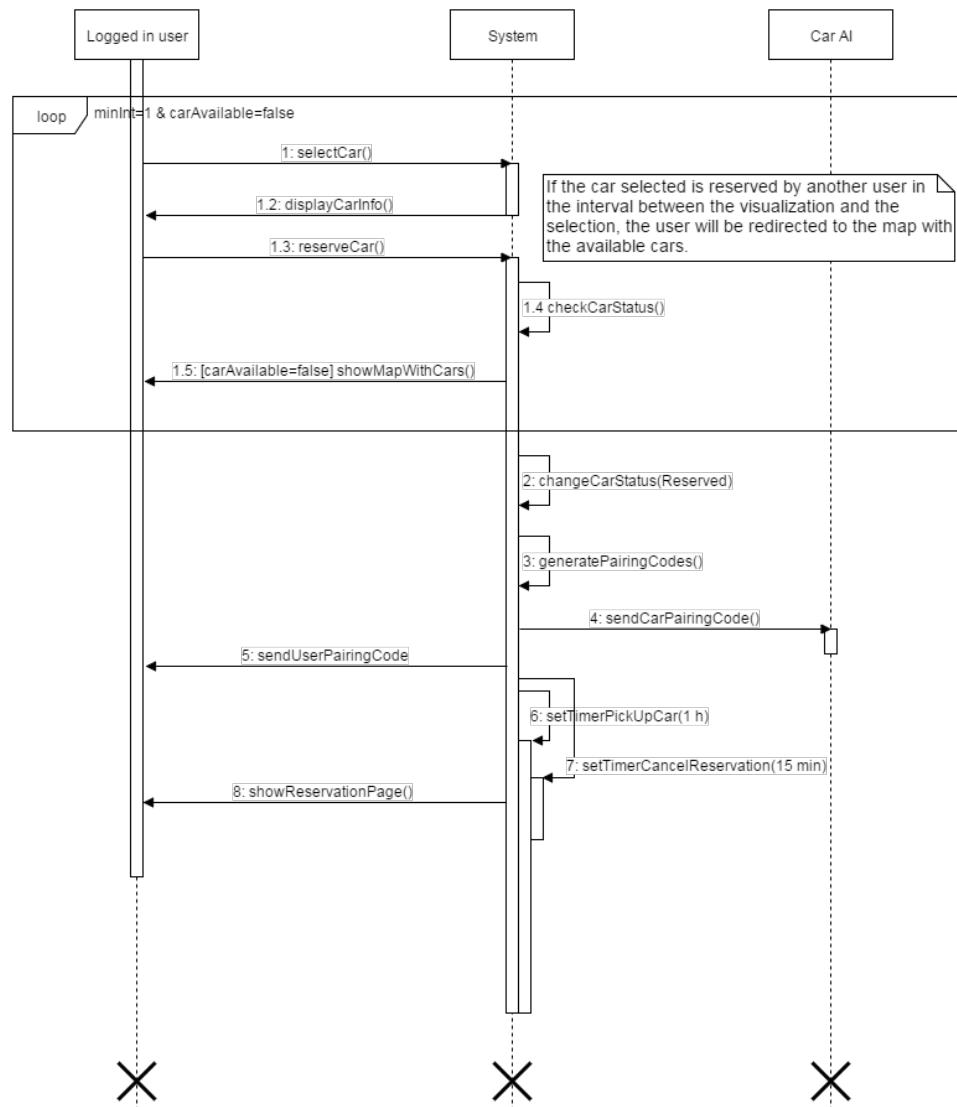


Figure 33: Reserving a car sequence diagram

3.5.5 Use Case: Deleting a reservation

Table 5: Deleting a reservation

Participating actors	<ul style="list-style-type: none"> • Logged in user • System
Entry condition	The user reserved a car not longer than 15 minutes ago.
Flow of events	<ol style="list-style-type: none"> 1. The user cancels the reservation. 2. The system redirects the user to the homepage and changes the car status as available again.
Exit condition	The cancellation is carried out successfully.
Exceptions	<ul style="list-style-type: none"> • The timer expires during the cancellation process. • The cancellation is carried out successfully.

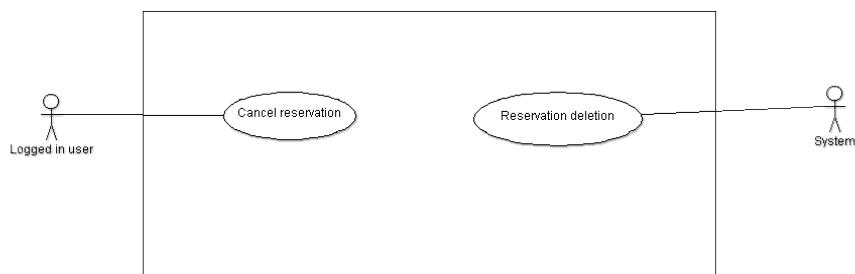


Figure 34: Deleting a reservation use case diagram

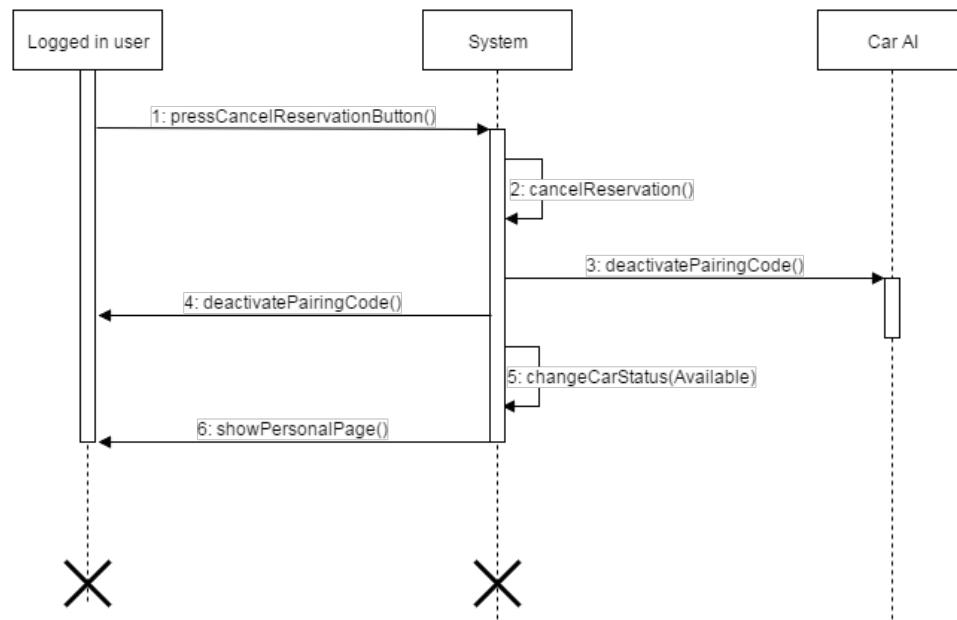


Figure 35: Deleting a reservation sequence diagram

3.5.6 Use Case: Picking up a car

Table 6: Picking up a car

Participating actors	<ul style="list-style-type: none">• Logged In User• Car AI
Entry condition	The user has reserved a car and has reached it within the allotted time.
Flow of events	<ol style="list-style-type: none">1. The car detects the presence of a smartphone connected to Power Enjoy whose Bluetooth function is activated, and starts the pairing procedure.2. If the pairing is successful the app asks to the user if the car is damaged. If there is no damage is reported, the user can unlock the car through the app.
Exit condition	The car is unlocked.

Exceptions	<ul style="list-style-type: none"> • The timer expires when the user does not reach the car. • The reservation expires, the system changes the car status to available again and charges 1 euro to the user. • The car is damaged. • The app asks the user to take a picture of the damages and the system changes the car status to not available. • The user will be redirected to the closest available car. • The user does not enter the car after he has requested its opening. • After 30 seconds from the end of the pairing procedure the car newly locks itself and the reservation is cancelled.
------------	--

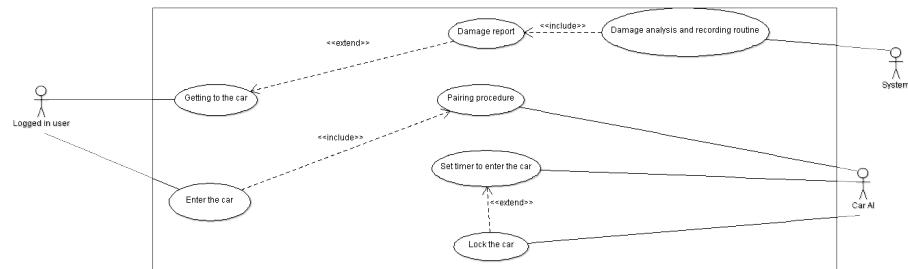


Figure 36: Picking up a car use case diagram

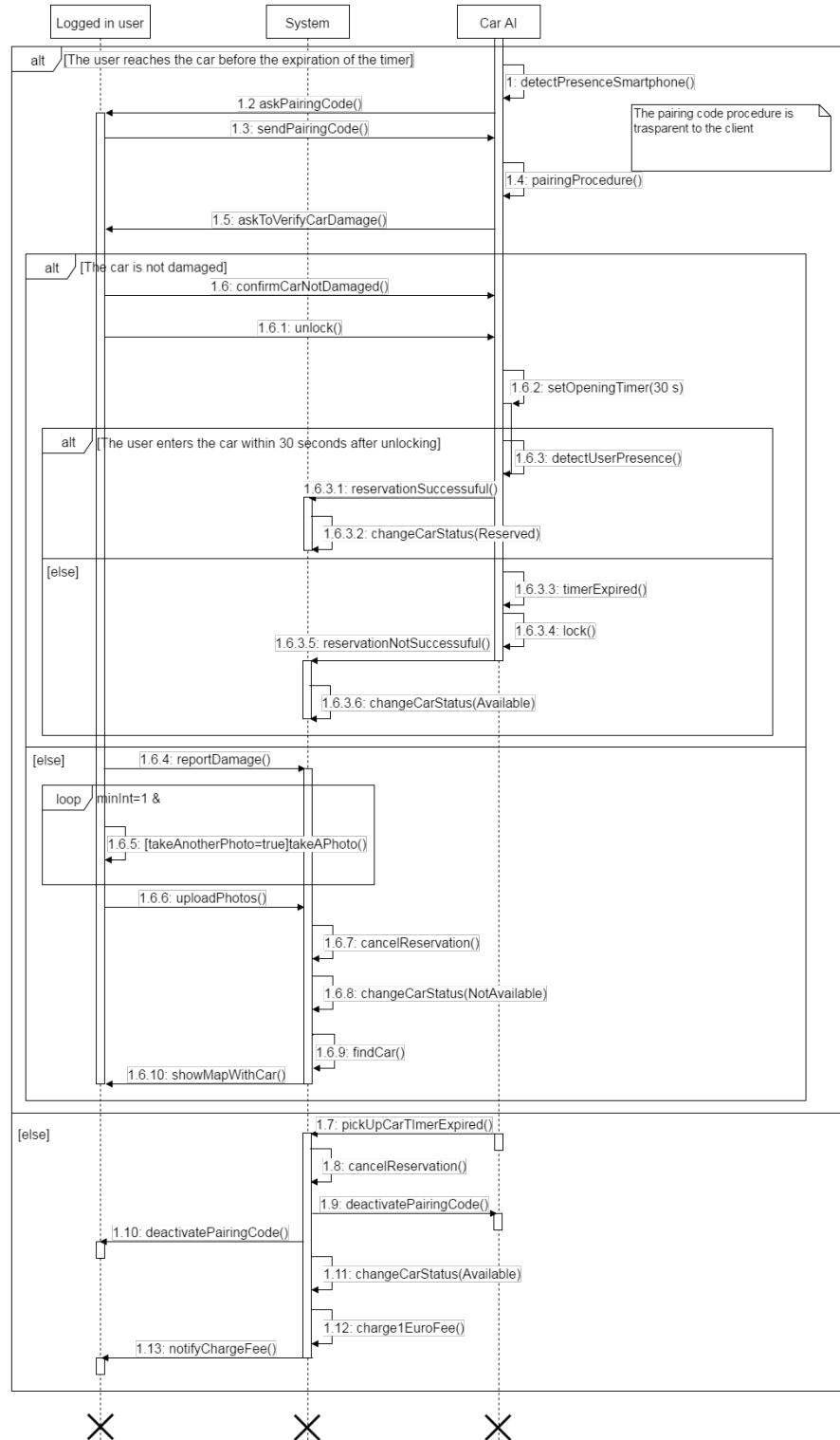


Figure 37: Picking up a car sequence diagram

3.5.7 Use Case: Managing the car being driven

Table 7: Managing the car being driven

Participating actors	<ul style="list-style-type: none">• Car AI• Logged In User
Entry condition	The user is in the car that he reserved.
Flow of events	<ol style="list-style-type: none">1. When the engine ignites, the system starts charging the user and does not stop until the car is parked in a safe area.2. The system displays through a screen the amount of money spent by the user, updating the said amount every minute, as well as the safe areas in the vicinity of the car in real time.3. When the user parks the car in a safe area and turns off the engine, the ride is deemed concluded for the moment.
Exit condition	The user parks the car in a safe area.

Exceptions	<ul style="list-style-type: none"> • The engine is ignited after the user has already stopped in a safe area. • The ride restarts, the base price is set as the price of the previous ride. • The car is damaged. • The user tries to get out of Milan. A warning is issued to the system and to the user shortly before reaching the boundaries. • If the car battery falls to 0% while the car is being driven and the car stops outside of a safe area. The ride is deemed concluded, but the user will be held responsible for any accident or disturbance caused.
Special requirements	<ul style="list-style-type: none"> • The system detects that the user takes at least two passengers onto the car. • The system is made aware by the car that the user will be entitled to a discount of 10% when the final price of the transaction will be calculated.

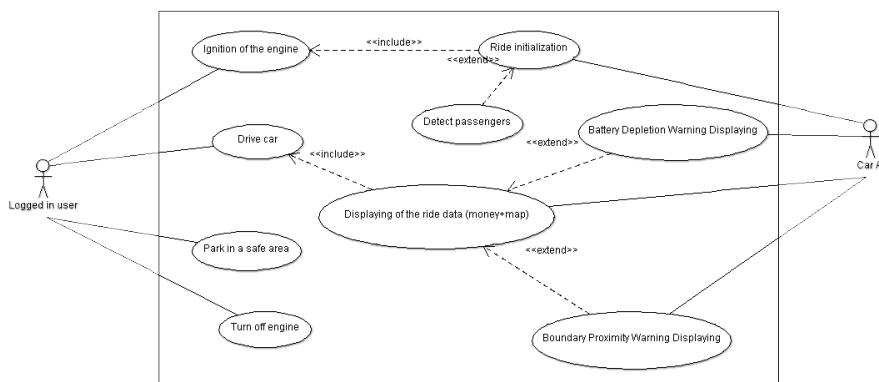


Figure 38: Managing the car being driven use case diagram

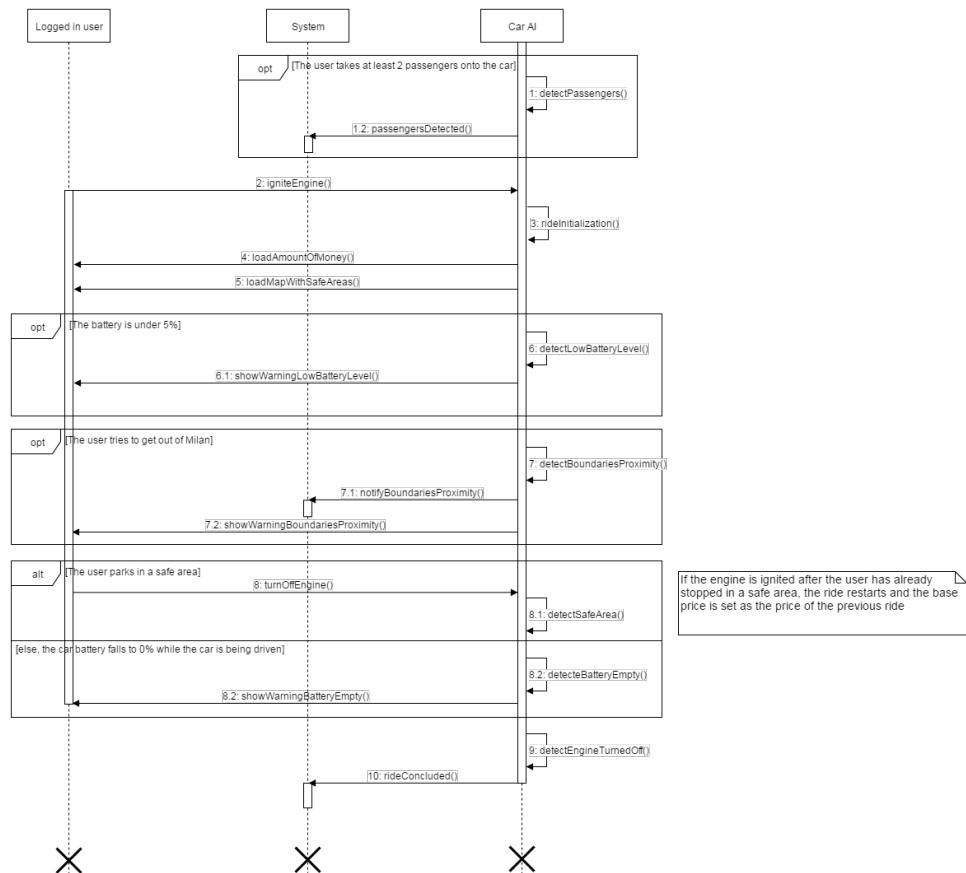


Figure 39: Managing the car being driven sequence diagram

3.5.8 Class Diagram

3.5.9 State Machine Diagram

3.6 Non Functional Requirements

3.6.0.1 Maintainability

3.6.0.2 Portability

Chapter 4

4 Appendix