POLITECNICO DI MILANO
2016-2017

SOFTWARE ENGINEERING 2: POWERENJOY

REQUIREMENTS ANALYSIS AND SPECIFICATIONS DOCUMENT
VERSION 1.0

PEVERELLI FRANCESCO
REPPUCCI FEDERICO

CONTENTS

1. INTRODUCTION
    1.1. DESCRIPTION OF THE PROBLEM 4
    1.2. GOALS 4
        1.2.1. PERSON 4
        1.2.2. USER 5
        1.2.3. EMPLOYEE 6
    1.3. DOMAIN PROPERTIES 6
    1.4. GLOSSARY 7
    1.5. ASSUMPTIONS OF THIS DOCUMENT 8
    1.6. STAKEHOLDERS IDENTIFICATION 8
2. ACTORS IDENTIFICATION
3. REQUIREMENTS
    3.1. FUNCTIONAL REQUIREMENTS 9
        3.1.1. PERSON 9
        3.1.2. USER 10
        3.1.3. EMPLOYEE 17
    3.2. NON-FUNCTIONAL REQUIREMENTS 18
        3.2.1. USER INTERFACES 19
        3.2.2. PERFORMANCE REQUIREMENTS 20
        3.2.3. SOFTWARE SYSTEM ATTRIBUTES 20
        3.2.4 LIMITATIONS 21
4. SCENARIO IDENTIFICATION 21
    4.1. TRIP TO THE SUPERMARKET 21
    4.2. TRANSFER STUDENT 21
    4.3. CINEMA WITH FRIENDS 22
    4.4. JOB INTERVIEW 22
    4.5. ABANDONED CAR 23
    4.6. TRIP TO THE COMPANY 23
    4.7. USUAL WEDNESDAY OF WORK 23
    4.8. BACK TO MILAN 24
5. UML MODELS

5.1. USE CASE DIAGRAM 24
5.1.2. USE CASE DESCRIPTION 25
5.2. CLASS DIAGRAM 36
5.3. SEQUENCE DIAGRAMS 36
5.4. STATECHART DIAGRAM 40
6. ALLOY MODELLING
6.1. MODEL 1 41
6.1.2. RESULTS 45
6.1.3. WORLD GENERATED 45
6.2. MODEL 2 46
6.2.2. RESULTS 51
6.2.3. WORLD GENERATED 51

1. INTRODUCTION

1.1 DESCRIPTION OF THE PROBLEM

The PowerEnJoy project aims to develop a car-sharing service run exclusively employing electric cars.

The system will provide a mobile application by means of which

the users, once registered, will be able to use the car sharing services.

The registration requires the user to provide a valid e-mail address, telephone number and a valid driving license, as well as some personal information and valid credit card data.

When a user completes the registration procedure, an e-mail is sent to him/her containing a password that will be used to access the platform.

Any registered user will be able to see on a map of the city the available cars within a certain radius either from their current position or from a selected address. By clicking on an available car on the map the user will be able to reserve it. The reserved car can be then unlocked via the appropriate function on the mobile app.

A pre-defined set of safe areas will be available for parking the cars. A number of these areas will also have power grid stations, providing a mean to recharge the cars.

The main goals of the service are to provide a sustainable and environmentally-friendly car sharing service as well as to promote virtuous behaviors from its users: a number of discounts will be applied to users who carry more than two passengers, leave the car with more than half of if its battery full or park it near to a power grid and take care of plugging it in. On the other hand, users who leave the car more than three kilometers away from the nearest power grid or leave it with less than 20% of its battery will be charged for an additional amount.

1.2 GOALS

1.2.1. PERSON

[G1.0]: Any person is able to register to the service by providing his/her credentials and valid payment info.

[G1.1]: He/she receives back a password with which he/she is able to access the system.

Additional goals emerged through scenario analysis

[GA1.0]: Any person registered to the system is able to log in the application

[GA1.1]: Only a person registered to the system is able to log in the application

[GA2]: Any person registered to the system is able to retrieve his/her password

[GA3]: Any person is able to access the PowerEnJoy website

1.2.2 USER

[G2.0]: The user is able to find the location of all available cars within a certain range from their current location.

[G2.1]: The user is able to find the location of all available cars within a certain range from a specified address.

[G3.0]: The user is able to pick a car among the available ones and reserve it.

[G3.1]: A reserved car is not available for renting until one hour has passed from the moment a user reserved it.

~~[G3.2]: After one hour from its reservation, a car becomes available again~~ (NO LONGER IN USE)

> ***RATIONALE***: G3.2 is formulated in a way which does not completely reflect the desired outcome, since a car is not meant to be available for renting even if an hour has passed from the moment it was reserved if it is currently in use, it has been left with less than 20% battery or it is more than 3km away from the nearest power grid, and therefore it has been substituted with the following goal.

[G3.2] A car becomes available again after one hour has passed from its reservation and it is parked in a safe area less than 3 km away from a power grid having more than 20% of its battery.

[G4]: The user pays 1 EUR if he/she doesn't reach the car he/she rent within 1 hour from the reservation.

[G5]: The user is able to unlock and open the car he/she rent when he/she is nearby the car.

[G6]: From the moment of ignition, the user is charged for a constant amount of money per minute

[G7.0]: The charging of the user stops as soon as the driver parks the car in a safe area and exits from it.

[G7.1]: The car is automatically locked as soon as the driver parks the car in a safe area and exits from it.

[G8]: A discount of 10% is applied on the last ride if the driver took at least two passengers onto the car and no higher discount or any extra fee can be applied.

[G9]: If a car is left with more than 50% of its maximum battery available, a discount of 20% is applied on the last ride and no higher discount or any extra fee can be applied.

[G10]: A discount of 30% is applied on the last ride if a car is left at special parking areas where they can be recharged and the driver takes care of plugging the car into the power grid and no higher discount or any extra fee can be applied.

[G11.0]: If a car is left at more than 3 kilometers from the nearest power grid station, the user is charged for an extra corresponding to 30% of the amount charged for the last ride.

[G11.1]: If a car is left with less than 20% of its maximum battery available, the user is charged for an extra corresponding to 30% of the amount charged for the last ride.

Additional goals emerged through scenario analysis

[GA4]: If the user has not paid for his/her last ride, the user cannot reserve a car

[GA5]: If the user's identity card or credit card has expired, he/she cannot reserve a car

[GA6]: The user is able to update expired credit card or identity card data

[GA7]: The user can reserve just one car at a time

[GA8]: If the user parks the car outside of a safe area he/she is charged for a **set** extra fee

1.2.3. EMPLOYEES

[G12]: If a car is parked in a non-safe area, an employee will retrieve it

[G13]: No one is charged for the retrieval of a car

1.3 DOMAIN PROPERTIES

[D1.0]: The credentials provided by the person at the moment of his/her registration are always correct, and always belong to the person carrying out the procedure.

[D1.1]: The credentials provided by the user while editing his/her profile are always correct.

[D1.2]: The validity check for the payment info delegated to the external payment service is always correct.

[D1.3]: The external payment service always charges the user for the exact amount of charging requested by the system.

~~[D2]: The user has always enough money to pay for the ride~~ (DEPRECATED)

[D2]: The user is always able to receive the e-mail after a finite number of tries

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

[D3.1]: The GPS coordinates of a user received by the system always correspond to the actual position of the user.

~~[D4]: If a person enters a car in the driver seat, also ignites it~~ (DEPRECATED)

[D5]: Every road traffic offense received by the company is forwarded to the user who last rent the car before the offense occurred.

~~[D6]: An employee is always able to retrieve a dislocated car within 24 hours~~ (DEPRECATED)

[D7.0]: If the sensors in the car detect other passengers, human passengers are on board

[D7.1]: If the sensors in the car detect no one inside, there are no people in the car

[D8]: Power grid stations are always operational

[D9]: The company's cars used by the employees are equipped with a navigation system

[D10]: After a finite number of times that a retrieval notification is sent for the same car, that car is retrieved

1.4. GLOSSARY

person: as far as our system is concerned, we consider only those people who possess a smartphone with GPS functionalities.

user: a logged-in person who uses the application and the services provided by PowerEnJoy.

driver: person who enters a car in the driver seat.

system: the server side software providing the core functionalities of the application.

application or app: the client side of the software present on the user's phone.

registration: iter through which the user can create a personal account in order to access the services of the application.

credentials: set of information provided by the user during the registration. These include: the user's first name, family name, gender, "Comune" of birth, "Provincia" of birth, his/her "Codice Fiscale", his/her identity card number, date of release and date of expiration, a valid driving license (B or higher or equivalent) a valid e-mail address and a mobile phone number.

payment info: a valid credit card number, verification value (CVV), expiration date and the holder's full name.

external payment service: a software system which allows the company to charge the users.

car: electric powered vehicles owned by PowerEnJoy.

range: distance value selected by the user.

available car: a car that can be reserved by a user for a future ride.

reserved car: a car that cannot be reserved and can only be used by the one who performed its reservation.

'in use' car: a car is in this state from the moment it is turned on by the driver until the moment he/she exits from it, which causes it to automatically close.

dislocated car: a car that is left in a non-safe area by the driver.

retrieve a car: action performed by an employee that can be described as follows:

-the employee is notified that a car has been left outside of a safe area

-the employee reaches the car, possibly manually recharges it and drives it back to a safe area.

safe area: A set of locations within a certain geographical area where the users are allowed to park their car, decided by the system administrator. These may include public car parks, specific areas of the city and private PowerEnJoy car parks

'nearby' the car: a user is 'nearby' the car when he/she is distant from the car less than 10 meters.

special parking area: part of safe area equipped with a power grid, in which user can recharge the electrical car.

power grid station: little tower that provides electrical current situated in a special parking area that allows the user to recharge a car.

1.5. ASSUMPTIONS OF THIS DOCUMENT

[T1]: We have in mind that our target customer has a smartphone which supports GPS related services

[T2]: We assume that the actual transaction with the payment service provider is performed only once, at a later time after the end of each the ride

[T3.0]: We assume only the highest-percentage discount to be applied on a ride.

[T3.1]: Any extra charge prevents any discount from being applied on a ride.

[T3.2]: All the extra charges are applied cumulatively; if two extra charges expressed as a percentage of the cost of last ride need to be applied, the resulting extra charge amounts to the sum of the two percentages.

1.6. STAKEHOLDERS IDENTIFICATION

The stakeholders identified for the PowerEnJoy project are the following:

- The PowerEnJoy stockholders, whose main interest is the margin of profit that the project hopes to generate and the public reception of the brand

- The PowerEnJoy users, who are expected above all to value the quality of the service and the pricing relative to the competitors

- The car leasing company, who is interested in the profit made through the partnership with our service, mainly dependent on its popularity

2. ACTORS IDENTIFICATION

The actors interacting with our system are the following:

Person: a person who is browsing PowerEnJoy's website looking for information on the service or a person who is using the app but is not yet registered.

User: a person who is logged into our system and can use all the services provided

by the application and the car sharing service.

PowerEnJoy employee: a person who works for the company and, among other things, is responsible for the retrieval of cars parked outside of the safe areas.

3. REQUIREMENTS

3.1. FUNCTIONAL REQUIREMENTS

3.1.1. PERSON

Goals

[G1.0]: Any person is able to register to the service by providing his/her credentials and valid payment info.

[G1.1]: He/she receives back a password with which he/she is able to access the system.

> ***NOTE***: G1.0 and G1.1 are interpreted as: providing one's credentials and some valid payment condition is sufficient and necessary to register to the system

Requirements

[R1.1]: The app is available for any person to download and run on his/her phone

[R1.2]: From the home page of the app any person can carry out the registration procedure

[R1.3]: The registration procedure requires a person's credentials and payment info to be carried out

[R1.4]: The registration procedure uses the external payment service to verify the validity of the provided payment info

[R1.5]: At the end of the registration procedure the person whose credentials were used is registered in the system

[R1.6]: At the end of the registration procedure the person receives an e-mail containing a password which he/she can use to access the system

[R1.7]: At the end of the registration procedure the person can ask the system to send another mail

Domain assumptions that ensure the desired outcome

[D1.0]: The credentials provided by the person at the moment of his/her registration are always correct, and always belong to the person carrying out the procedure.

[D1.2]: The validity check for the payment info delegated to the external payment service is always correct.

[D2]: A person is always able to receive an e-mail sent by the system after a finite number of tries

Additional goals emerged through scenario analysis

[GA1.0]: Any person registered to the system is able to log in the application.

[GA1.1]: Only a person registered to the system is able to log in the application

Requirements

[RA1.1]: The app allows any person to log in by providing a valid e-mail and password

[RA1.2]: The app does not allow any person who does not provide a valid e-mail and password to log in

Domain assumption that ensures the desired outcome

[DA1]: Only a registered person is able to provide a valid e-mail and password

Goals

[GA2]: Any person registered to the system is able to retrieve his/her password

Requirements

[RA2.1]: From the home page of the app the password retrieval procedure can be initiated by any person

[RA2.2]: If a person provides a valid e-mail address during the password retrieval procedure the system sends an e-mail to that address containing the associated password

Domain assumption that ensures the desired outcome

[DA2]: If a password is sent by e-mail to a person's address that password is retrieved by the person

[D2]: A person is always able to receive an e-mail sent by the system after a finite number of tries

Goals

[GA3]: Any person is able to access the PowerEnJoy website

Requirements

[RA3]: Access to the PowerEnJoy's website (a static page) is granted upon request by the system (no login required)

Domain assumption that ensures the desired outcome

[DA3]: Any person is able to request any webpage (use a browser)

3.1.2 USER

Goals

[G2.0]: The user is able to find the location of all available cars within a certain range from their current location.

[G2.1]: The user is able to find the location of all available cars within a certain range from a specified address.

Requirements

[R2.1]: The "Reserve a car" function can be accessed by the user from the home page of the app

[R2.2]: The "Reserve a car" function allows the user to select a range (distance)

[R2.3]: The system acquires the user's current position through the GPS coordinates of the user's phone

[R2.4]: The system tracks all available cars' current position through their GPS coordinates

[R2.4.1]: The cars must possess a device which can be tracked via GPS

[R2.5]: The "Reserve a car" function allows the user to select a starting position for the search, which can be either their current location or a given address

[R2.6]: When the user confirms the inserted parameters the search is carried out and the "Reserve a car" function displays to the user the data of the search acquired from the system in a Google provided map

> ***NOTE***: here we chose not to detail the interaction between the app and the system, to avoid talking about any architectural detail.

Domain assumptions that ensure the desired outcome

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

[D3.1]: The GPS coordinates of a user received by the system always correspond to the actual position of the user.

Goals

[G3.0]: The user is able to pick a car among the available ones and reserve it.

> ***NOTE***: G3.0 is interpreted as: if there are available cars the user can pick any one among them and reserve it, so it does not guarantee anything regarding the presence of available cars. Moreover, the availability of the selected car needs to persist until the moment the user confirms the reservation of a car or interrupts the procedure for it to be considered an available car in this instance.

Requirements

[R3.1]: The app allows the user to tap on any available car on the map displayed as the result of a search conducted through the "Reserve a car" function.

[R3.2]: When a user taps on a car the app generates a pop-up asking the user if he/she wants to confirm the reservation.

[R3.3]: As long as the car was not reserved by another user in the meantime, when the user confirms the car is marked as reserved by the system and the user can see the "Reservation successful!" message on the app.

Goals

[G3.1]: A reserved car is not available for renting until one hour has passed from the moment a user reserved it.

~~[G3.2]: After one hour from its reservation, a car becomes available again~~ (NO LONGER IN USE)

[G3.2] A car becomes available again after one hour has passed from its reservation and it is parked in a safe area less than 3 km away from a power grid having more than 20% of its battery

Requirements

[R3.1]: When the system marks a car as reserved any reservation request from any user is rejected by the system while the car is in the reserved state.

[R3.2]: A car is in reserved state for one hour from the moment it was marked as reserved.

[R3.3]: A car in reserved state is not signaled by the system during the "Reserve a car" procedure.

[R3.4]: After one hour from its reservation a car is no longer in reserved state.

[R3.5]: A car not in reserved state is considered available by the system only if it is parked in a safe area less than 3 km away from a power grid station and has more than 20% of its battery.

Goals

[G4]: The user pays 1 EUR if he/she doesn't reach the car he/she rent within 1 hour from the reservation.

> ***NOTE***: "reach" in this case means that the user has ignited the car. This is to prevents the user from avoiding the fee by just opening the car, discouraging negative behavior.

Requirements

[R4.1]: One hour after a car has been reserved if it was never ignited the system charges for 1 EUR the user who reserved it.

Goals

[G5]: The user is able to unlock and open the car he/she rent when he/she is nearby the car

Requirements

[R5.1]: From the home page of the app the user can access the "My reservations" section

[R5.2]: In the "My reservations" section if the user has reserved a car less than an hour ago an active reservation is displayed with an "Unlock" button

[R5.3]: If the user is less than 10 meters away from the car and presses the unlock button the car unlocks

> ***NOTE***: a user is a logged in person, therefore he/she is identified by the system through an account and a GPS position. Whether the physical person who opens the car is the same who reserved it is not our concern.

Goals

[G6.0]: From the moment of ignition, the user is charged for a constant amount of money per minute.

[G6.1]: The driver is notified of the current charges through a screen on the car.

[G7.0]: The charging of the user stops as soon as the driver parks the car in a safe area and exits from it.

> ***NOTE***: The request to the external payment service is made only once, and after the end of each ride; here by charged we mean that the system records and updates the amount of money to charge the user for at a later time.
>
> ***NOTE***: G7 is interpreted as: if the car is parked in a safe area and the driver and all the passengers exit from it, the user stops being charged (as a simple implication)

Requirements

[R6.1]: When a car is ignited the system starts charging the last user who reserved the car

[R6.2]: When the charging starts, the display on the car shows a "Current charge" field with a number representing the current total charge, which starts from 0

[R6.3]: Once a minute the "Current charge" value is incremented by a **set** amount

[R7.1]: When a car is stopped and the sensors in the car detect no one inside, if a user was being charged for the car the system stops charging him/her.

> ***NOTE***: in R7.1 we mean as soon as both conditions are met, the user stops being charged.

Goals

[G7.1]: The car is automatically locked as soon as the driver parks the car in a safe area and exits from it.

Requirements

[R7.2]: One minute after a car has been stopped and the sensors in the car detect no one inside, the system locks the car.

Domain assumption that ensures the desired outcome

[D7.1]: If the sensors in the car detect no one inside, there are no people in the car

Goals

[G8]: A discount of 10% is applied on the last ride if the driver took at least two passengers onto the car and no higher discount or any extra fee can be applied.

Requirements

[R8]: The moment the car is stopped, if the sensor in the car detected two passengers the system records it as a possible discount of 10%

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride

Domain assumptions that ensure the desired outcome

[D7.0]: If the sensors in the car detect other passengers, human passengers are on board.

[D7.1]: If the sensors in the car detect no one inside, there are no people in the car.

Goals

[G9]: If a car is left with more than 50% of its maximum battery available, a discount of 20% is applied on the last ride and no higher discount or any extra fee can be applied.

Requirements

[R9]: The moment the car is stopped and the sensors in the car detect no one inside, if the car has more than 50% of its maximum battery the system records it as a possible discount of 20%.

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride.

Goals

[G10]: A discount of 30% is applied on the last ride if a car is left at special parking areas where they can be recharged and the driver takes care of plugging the car into the power grid and no higher discount or any extra fee can be applied.

Requirements

[R10]: If before 2 minutes since the moment the car has been stopped and its sensors detected no one was inside the car is plugged in a power grid and its position is within a special parking area, the system records it as a possible discount of 30%.

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride

Domain assumption that ensures the desired outcome

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

[D8]: Power grid stations are always operational

Goals

[G11.0]: If a car is left at more than 3 kilometers from the nearest power grid station, the user is charged for an extra corresponding to 30% of the amount charged for the last ride.

Requirements

[R11.0]: The moment the car is stopped and the sensors in the car detect no one inside, if the safe area nearest to the car is more than 3km away from it, the system records an extra fee of 30%

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride

Domain assumptions that ensure the desired outcome

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

Goals

[G11.1]: If a car is left with less than 20% of its maximum battery available, the user is charged for an extra corresponding to 30% of the amount charged for the last ride.

Requirements

[R11.1]: The moment the car is stopped and the sensors in the car detect no one inside, if the car has less than 20% of its maximum battery, the system records an extra fee of 30%.

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride.

Additional goals emerged through scenario analysis

[GA4]: If the user has not paid for his/her last ride, the user cannot reserve a car.

Requirements

[RA4]: If a user with a pending payment procedure tries to reserve a car, a pop-up lets the user know that he/she needs to pay for his/her last ride to be able to reserve a car and the app does not allow the user complete the reservation procedure

Goals

[GA5]: If the user's identity card or credit card has expired, he/she cannot reserve a car

Requirements

[RA4]: If in the user's profile either the credit card or identity card expiration date has already passed, when the user tries to reserve a car a pop-up lets him/her know that the data in the user's profile need to be updated and the app prevents the reservation procedure from being completed

Goals

[GA6]: The user is able to update expired credit card or identity card data

Requirements

[RA6.1]: From the home page of the app the user can access the "My profile" section

[RA6.2]: From the "My profile" section the user can use the "Edit profile" button to modify his/her credential and payment info

[RA6.3]: The system can check via the external payment service whether the payment info inserted are valid

[RA6.4]: If the inserted payment info is valid the user can save the changes by tapping the "Confirm" button.

Domain assumptions that ensure the desired outcome

[D1.1]: The credentials provided by the user while editing his/her profile are always correct.

[D1.2]: The validity check for the payment info delegated to the external payment service is always correct.

Goals

[GA7]: The user can reserve just one car at a time

Requirements

[RA7]: If the user has already a reservation which is not expired yet when he/she tries to reserve a car, a pop-up lets the user know that he/she needs to pay for his/her last ride to be able to reserve a car and the app does not allow the user complete the reservation procedure

Goals

[GA8]: If the user parks the car outside of a safe area he/she is charged for a set extra fee

Requirements

[RA8]: When a car is locked the system checks its GPS coordinates, and if they correspond to those of a non-safe area the last user who reserved the car is charged for a set extra fee.

Domain assumptions to ensure the desired outcome

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

3.1.3 EMPLOYEE

During requirements elicitation, we identified the need to specify some additional goals for the part of the system used by the employees, the resulting set of additional functional requirements for the system

Goals

[G12]: If a car is parked in a non-safe area, an employee will retrieve it

Requirements

[R12.1]: Each employee has access to an application, AdminPowerEnJoy, on their phone

[R12.2]: When a car is locked the system checks its GPS coordinates, and if they correspond to those of a non-safe area all employees are notified through AdminPowerEnJoy that the car needs to be retrieved

[R12.3]: AdminPowerEnJoy allows an employee to accept a retrieval request through the "Retrieval procedure" function

[R12.4]: If an employee has already accepted a retrieval request, the retrieval request can no longer be accepted

[R12.5]: After 12 hours, if an employee has accepted a retrieval request but has not retrieved the car, the request is issued again by the system and another employee can accept it

[R12.6]: When an employee is notified of a car to retrieve, the notification contains the information necessary to set up the navigator of the company's cars to find the position of the car to retrieve

[R12.7]: AdminPowerEnJoy allows an employee to unlock any car for which he/she has accepted a retrieval request

Domain assumptions to ensure the desired outcome

[D3.0]: The GPS coordinates of the cars received by the system always correspond to the actual positions of the cars.

[D9]: The company's cars used by the employees are equipped with a navigation system

[D10]: After a finite number of times that a retrieval notification is sent for the same car, that car is retrieved

Goals

[G13]: No one is charged for the retrieval of a car

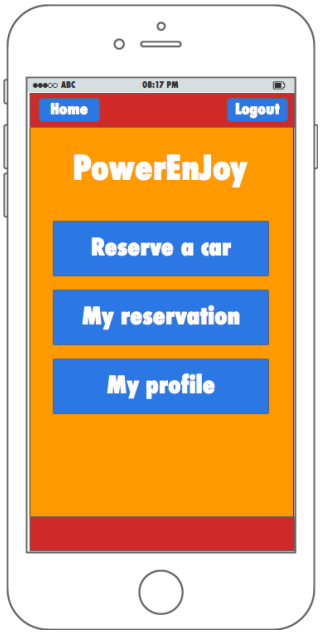
Requirements

[R13.1]: When the employee ignites a car which was opened through AdminPowerEnJoy the system does not initiate any charging procedure

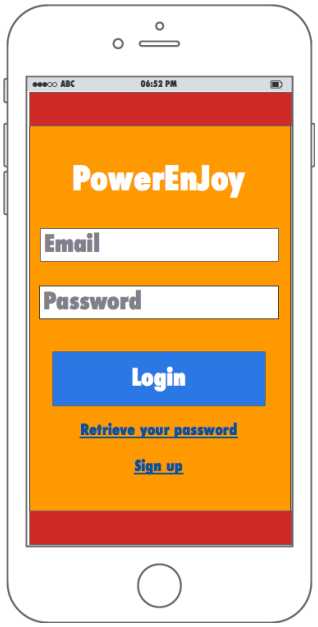
3.2. NON-FUNCTIONAL REQUIREMENTS

3.2.1. USER INTERFACES

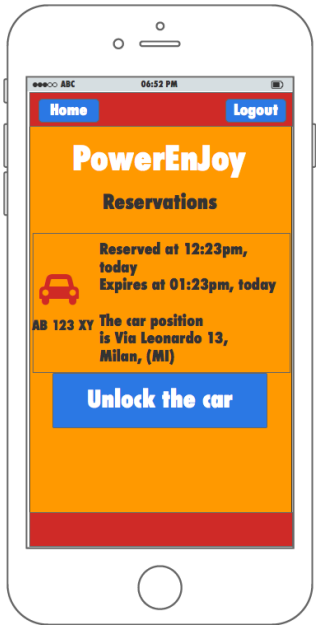
Registration layout



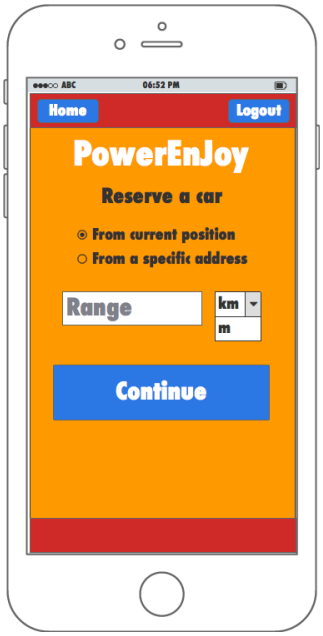
Home

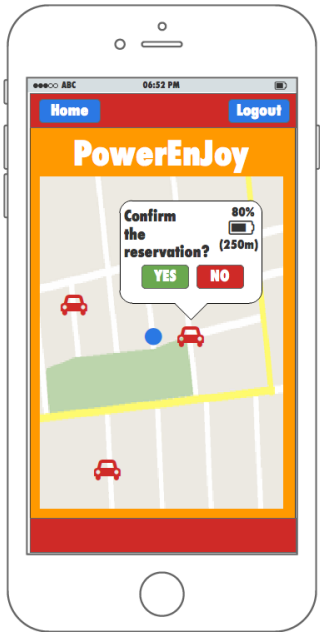


Login

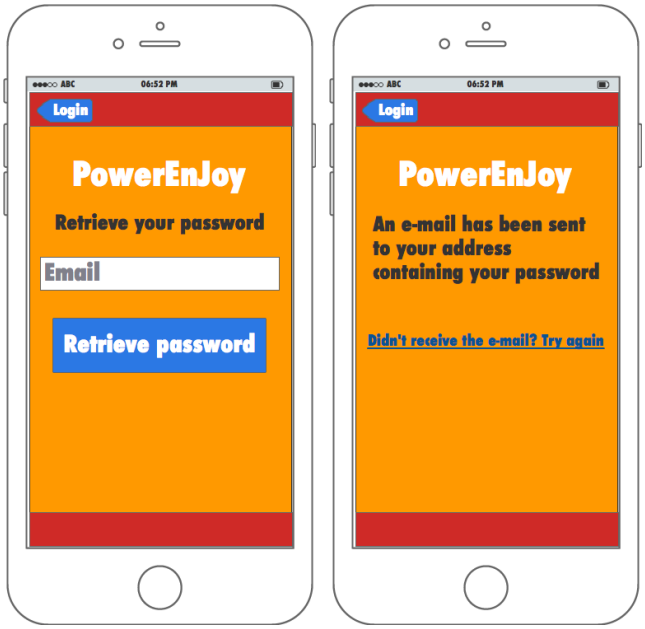


Reservation





Reserve a car



Retrieve password

3.2.2 PERFORMANCE REQUIREMENTS

[NFR1]: The system is able to handle the interaction with the expected number of users and employees guaranteeing a maximum response time of 5 seconds

3.2.3 SOFTWARE SYSTEM ATTRIBUTES

Scalability:

[NFR2]: The system is highly scalable to adapt to the possible increasing number of users

Capacity:

[NFR3]: The system is able to store all the data regarding the users, the cars and the employees up to the expected number of users

Availability:

[NFR4]: The system needs to be operational 24/7

Robustness:

[NFR5]: The system's database uses proper methods to guarantee fault tolerance

Interoperability:

[NFR6]: Both the PowerEnJoy app and the AdminPowerEnJoy app are compatible with all the major mobile OS's (iOS, Android, Windows)

Integrity:

[NFR7]: The system has the proper data storage hardware to guarantee the integrity of the data

Security:

[NFR8]: The system has an encryption protocol which guarantees the safety of the users' passwords, identity card data, payment info data and driving license data

3.2.4 LIMITATIONS

Interfaces to other applications:

[NFR9]: The system is able to interact with the external payment service of choice

Safety and security considerations:

[NFR]: A system failure does not impair the ability of the users to drive the cars and exit from them

4. SCENARIO IDENTIFICATION

4.1 TRIP TO THE SUPERMARKET

Yesterday Betty brought her car to the mechanic to change the front tires. Today she needs to go to the supermarket, which is approximately 5km away. Since she notices that today there is a strike of public transport she chooses to use the new electric car-sharing system provided by PowerEnJoy. So she takes her smartphone and opens the PowerEnJoy application. From the menu, she chooses "Reserve a car", checking the option "From my position",

and sets the maximum distance within which she would like to find a car to 500 meters. There are 2 cars within 500 meters and Betty reserves the nearest one.

After a while, she arrives at the car and she is able to unlock it through the proper button "Unlock the car". After getting in the car, she ignites it pushing the "Start" button next to the steering wheel.

Betty arrives at the supermarket 10 minutes later. She parks the car in the supermarket parking lot and 30 seconds after she exits the car, the system locks automatically the car tagging it as "available" again.

4.2 TRANSFER STUDENT

Giulia has come from her hometown Pescara to Milano to attend university and her new friends tell her that she should consider registering to a car sharing service, and so after a bit of search on the web she reads about PowerEnJoy. The curiosity towards electric powered cars and the potential benefits that the service could provide to her daily life lead her to decide to register to the service, so she downloads the app to her smartphone and installs it.

After starting the app, a sign in/sign up screen appears, so she selects the sign up option and starts filling in the required information.

After double-checking that her identity card's number and her Codice Fiscale are typed correctly, as well as the data of her driving license, she proceeds to the next step of the registration process.

At this point the system asks Giulia to provide a valid credit card to associate to her account. Luckily her parents just recently gave her a visa credit card for her living expenses, so she promptly inserts the credit card number and CVV. The system replies confirming that the payment information she provided are indeed valid, and so her registration is almost complete: upon agreeing to the terms of services confirmation e-mail is sent to her, she opens her mail box and she writes down the password that the system has sent her, and then tries to log in the application.

The e-mail and password provided are correct and so Giulia can start using PowerEnJoy.

4.3 CINEMA WITH FRIENDS

Stefano Chiara and Marco have decided to go see a movie tonight, and since is Marco's turn to drive today, he has decided to reserve a PowerEnJoy for himself and his friends.

To do so, Marco selects the "Reserve a car" option on the app and checks the option "From address", which allows him to reserve a car near his and his friends' usual meeting point. Even though Stefano was a bit late, they all manage to arrive where the car is in time. As soon as his friends are all present, Marco uses the "Unlock the car" option on the app on his smartphone and they all get in the car before the reservation expires. At this point Marco ignites the car through the button near the steering wheel and he heads towards his destination. It is quite a long trip to the movie theatre, and unfortunately the battery of the car was already at 60% so by the time they reach their destination the battery of the car is at 35%. At this point Marco parks the car in a proper parking spot at the movie theatre and everyone exits the car. Marco checks on his smartphone to see how much the ride has cost him, and finds out that the presence of his friends has netted him a 10% discount.

4.4 JOB INTERVIEW

Giovanni has a job interview today, and since he has just recently registered to PowerEnJoy he decides to reserve a car near his house, to drive to the place where the interview will be held. He opens the PowerEnJoy app and looks for a car from his current position up to 1 km away and finds one nearby, so he gets there and unlocks the car using the option provided by the application. After igniting the car, he starts driving and realizes the mistake he has made: there are a lot of people on the road today and it is getting late. He finally reaches his destination, but an even worse problem presents itself, that is to say there are no parking spots to leave the car. Knowing full well that he is in the wrong, he decides to park the car illegally, hoping to get away with it. After he exits the car a notification informs him that an extra fee has been applied to his ride, since he decided to park in a non-safe area.

4.5 ABANDONED CAR

Giorgio's shift starts as he meets up with his colleague Mario, and he receives a notification telling him that a car has been left in a non-safe area. Seeing how there are no other urgent matters that require their attention, Giorgio and Mario decide to head there immediately. Giorgio uses the app provided by the company to signal that he intends to head there to avoid the deployment of more cars than needed. After entering the company's car, they insert the code representing the car to retrieve in order to obtain the necessary information about the car's location through their car's navigator. They bring an extra battery with them to recharge the car and drive to the location. As soon as they arrive there they recharge the car and Giorgio unlocks it with the app and drives it back to a safe area where Mario picks him up once again.

4.6 TRIP TO THE COMPANY

Clare needs to be at her company's conference room within two hours in order to attend the monthly company meeting. Today the mayor of Milan has announced a car ban in order to reduce the amount of pollution in the air. Given that the nearest bus stop is ten minutes away on foot from her house, she decides to rent a PowerEnJoy electric car to get to the company, since during the car ban electric cars can circulate, in virtue of being considered non-polluting machines.

Opening the PowerEnJoy app, through the proper "Reserve a car" functionality, she finds a couple of cars just next to her home, one or two minutes away on foot.

After reserving the nearest one she waits about twenty minutes before reaching the car to drive towards her company.

Unfortunately, ten minutes after her reservation, Daniel - her boss - texts her that the meeting has been postponed by a week. Therefore, Clare has booked a car that she will never use. As stated the PowerEnJoy's terms of service, if a user's reservation expires and he/she doesn't pick up the car within one hour from the reservation, he/she will pay a fee of 1 EUR. As expected, one hour after the reservation of the car, the system deducts 1 EUR from Clare's credit card.

4.7 USUAL WEDNESDAY OF WORK

Usually Franco uses public transportation to go to work, but since on Wednesdays he needs to go to a different building which is not well served by the subway and he does not have a car, he registered to PowerEnJoy to be able to go there by car.

So as he does every week Franco reserves a car through the appropriate function on the app, checking the "From my position" option. Then he reaches the car and unlocks it through the "Unlock the car" option, ignites it and drives to work. The place where he needs to go is near a PowerEnJoy power grid station, so he hopes that there are currently available power grids there. Seeing that there are indeed available parking spots at the

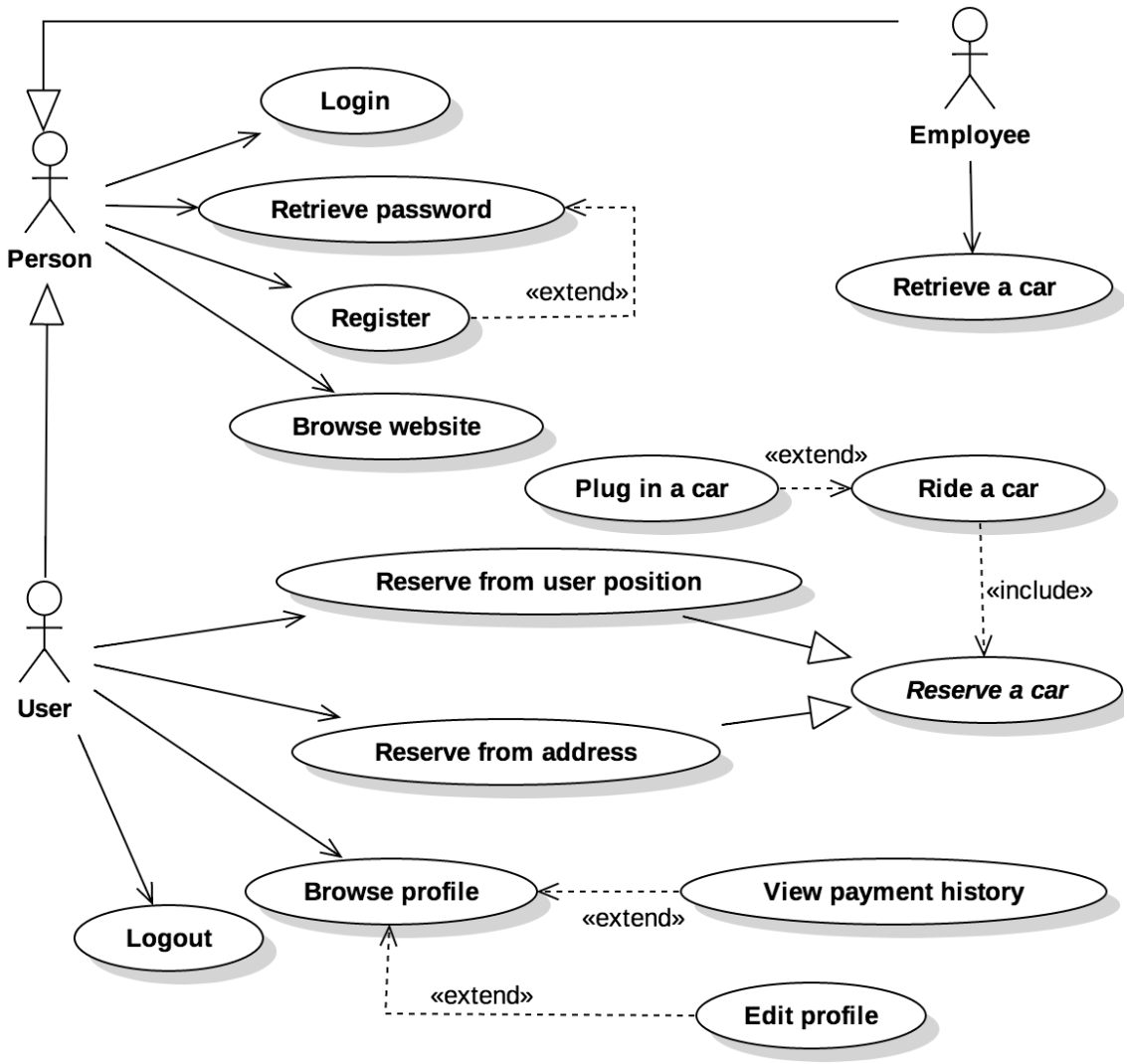
usual place, and parks there, also taking care of plugging the car in, which nets him a 30% discount on the ride.

4.8 BACK TO MILAN

Pietro moved back to Milan recently, so he wants to return to use PowerEnJoy, to which he is already registered. Realizing that he has forgotten his password, he decides to initiate the password retrieval procedure via the “Retrieve your password” link in the app’s login screen. After inserting his e-mail address, he confirms and quickly receives an e-mail containing his password. Upon logging in he realizes that both his credit card and id card data are outdated, so he uses the “Edit profile” function in the “My profile” section to insert his new credit card and identity card expiration date. After checking the correctness of the data he typed he confirms the modification. At this point Pietro is able to start using PowerEnJoy again.

5. UML MODELS

5.1. USE CASE DIAGRAM



5.1.2. USE CASE DESCRIPTION

Name*:* **Register**

Participating actors

Non logged person: someone who is not logged in the system

Entry condition

The person downloads the app, starts it and clicks on "Sign up".

Flow of events

1. A new screen appears containing a number of blank required fields and a "Continue" button.

2. The non-registered person fills the required fields with his/her personal data which include: a valid e-mail address which is not already present in the system's database, a valid telephone number, the non-registered person's full name, gender, "Comune", "Provincia" and date of birth, his/her "Codice Fiscale", his/her driving license data, his/her identity card number, release date and expiration date.

3. The non-registered person clicks the "Continue" button.

4. A new screen appears containing three required fields to fill: the credit card circuit (Visa, MasterCard, American Express...), the credit card number and the credit card's Card Verification Value (CVV) as well as a tick-box for the "Terms of Service" agreement and a "Register" button, both inactive.

5. The non-registered person fills the required fields with the data of a valid credit card.

6. The system sends a verification request to the external payment service for the credit card information.

7. The external payment service confirms to the system the validity of the data and the tick-box and "Register" button become active.

8. The non-registered person clicks on the "Register" button.

9. A screen appears with the following sentence: "Registration successful! Please check your e-mail, we sent you the password to access your account" and also a "Didn't receive the e-mail? Try again!" link.

Exit condition

The use case terminates when the e-mail sent from the system is received by the non-registered person

Exceptions

- System/connection failure: if either the system ceases to function or the connection with the person's phone is lost at any time between step 2 and 8 before step 8 is completed, the procedure needs to be performed again from the entry condition. If the connection is lost after step 8 is completed and the user does not receive the e-mail nor is able to

request another e-mail to be sent, the user will need to initiate the Retrieving Password use case.

- Invalid data: if the non-registered person is not able to provide valid data to fill any of the required fields, the procedure cannot be carried out any further, and the non-registered person can only stop the procedure by exiting the app.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Retrieve password**

Participating actors

Non-logged person: someone who is not logged in the system.

Entry condition

The person starts the app and taps on "Retrieve your password".

Flow of events

1. A new screen appears containing a blank text box named "E-mail" and a button named "Retrieve password".

2. The person writes his/her e-mail in the text box and taps on the "Retrieve password" button.

3. A new screen appears with the following sentence: "An e-mail has been sent to your address containing your password" and a link named "Didn't receive the e-mail? Try again".

Exit condition

The person receives the e-mail containing his/her password.

Exceptions

-E-mail not received: the user can tap on the "Didn't receive the e-mail? Try again" button and he/she will eventually receive the e-mail.

-Invalid data: if the non-registered person is not able to provide an e-mail address present in the system's database, the procedure cannot be carried out any further and the person can only stop the procedure by exiting the app.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Browse website**

Participating actors

Non-logged person: someone who is not logged in the system.

Entry condition

The person uses his/her browser to navigate to the PowerEnJoy website.

Flow of events

1. A static page appears containing all relevant information about the service and the terms and conditions for the users.

Exit condition

The person closes the browser or requests another web page.

Special requirements

- The non-registered person's phone/PC is required to be connected to the internet either via Wi-Fi or 3G (or similar) or LAN connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Login**

Participating actors

Non-logged person: someone who is not logged in the system.

Entry condition

The person starts the PowerEnJoy app.

Flow of events

1. A screen appears containing two blank textboxes named "E-mail" and "Password" respectively and a "Login" button.

2. The person fills in the two textboxes with his/her e-mail address and the password to access his/her account.

3. The user taps on the "Login" button.

Exit condition

A new screen displays the user's profile. The person can now access all the functionalities available to a user.

Exceptions

-Invalid data: if the non-registered person is not able to provide valid data to fill the required fields, the procedure cannot be carried out any further, and the person can only stop the procedure by exiting the app.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Reserve a car** (abstract)

Participating actors

User: a person who is logged in the system.

Entry condition

The user opens the PowerEnJoy app, logs in and taps "Reserve a car".

Flow of events

1. A new screen appears containing a "Continue" button and a form where the user is able to select:
    - The range of the search
    - The unit of measurement of such range
    - The starting point of the search
2. The user inserts the range in which he/she wants to find a car.
3. The user selects the desired unit of measurement (km is preselected)
4. The user selects a starting point for the search.
5. The user taps the "Continue" button.
6. The application displays a Google provided map with the found cars.
7. The user taps on the object representing the car he/she wants to reserve.
8. A pop-up appears asking the user whether he/she wants to confirm the reservation.
9. The user taps on the "Yes" button.
10. The app confirms to the user that the car has been reserved correctly by displaying the following sentence: °Reservation successful!”.

Exit condition

A notification from the app tells the user that the car has been reserved.

Exceptions

- System/connection failure: if either the system ceases to function or the connection with the user's phone is lost at any time between step 2 and 9 before step 9 is completed, the procedure needs to be performed again from the entry condition.

- Invalid data: in the text box in the first screen the user can insert only a positive integer number. If the user inserts something else, the button "Continue" becomes inactive.

- Pending payment: if the user has yet to pay for his/her last ride, when the user taps the continue button in step 5 another pop-up appears which reads: "Pending ride, cannot confirm reservation". The user must interrupt the procedure.

- Expired credit card/identity card: if the user's credit card or identity card has expired, when the user taps the continue button in step 5 another pop-up appears which reads: "Expired document, please edit profile". The user must interrupt the procedure.

- Double reservation: if the user has already a reservation which is not expired yet, when the user taps the continue button in step 5 another pop-up appears which reads: "Another reservation is active".

Special requirements

- The user's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The user's phone needs an activated GPS connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Reserve a car from address**

Participating actors

User: a person who is logged in the system.

Entry condition

The user opens the PowerEnJoy app on his/her smartphone and taps on the "Reserve a car" option.

Flow of events

1. A new screen appears containing:

  • two mutually exclusive checkboxes used to decide whether the user wants to search for a car from his/her current position or from a specific address

  • a blank text box used to specify the desired range of the search

  • a drop-down list containing the unit of measurement (m or km)

  • a “Submit” button

2. The user inserts the range in which he/she wants to find a car.

3. The user selects the desired unit of measurement (km is preselected)

4. The user selects the "From Address" checkbox.

5. The user taps the "Submit" button.

6. The application displays a Google provided map with the found cars

7. The user taps on the object representing the car he/she wants to reserve.

8. A pop-up appears asking the user whether he/she wants to confirm the reservation.

9. The user taps on the "Yes" button.

10. The app confirms to the user that the car has been reserved correctly by displaying the following sentence: “Reservation successful!”.

Exit condition

A notification from the app tells the user that the car has been reserved.

Exceptions

- System/connection failure: if either the system ceases to function or the connection with the user's phone is lost at any time between step 2 and 9 before step 9 is completed, the procedure needs to be performed again from the entry condition.

- Invalid data: in the text box in the first screen the user can insert only a positive integer number. If the user inserts something else, the button “submit” becomes inactive.

-Pending payment: if the user has yet to pay for his/her last ride, when the user taps the continue button in step 5 another pop-up appears which reads: "Pending ride, cannot confirm reservation". And the user must interrupt the procedure.

-Expired credit card/identity card: if the user's credit card or identity card has expired, when the user taps the continue button in step 5 another pop-up appears which reads: "Expired document, please edit profile". The user must interrupt the procedure

-Double reservation: if the user has already a reservation which is not expired yet, when the user taps the continue button in step 5 another pop-up appears which reads: "Another reservation is active".

Special requirements

- The user’s phone is required to be connected to the internet

either via Wi-Fi or 3G (or similar) connection.

- The user’s phone needs an activated GPS connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Reserve a car from user position**

Participating actors

User: a person who is logged in the system.

Entry condition

The user opens the PowerEnJoy app on his/her smartphone and taps on the "Reserve a car" option.

Flow of events

1. A new screen appears containing:

  • two mutually exclusive checkboxes (radio buttons) used to decide whether the user wants to search for a car from his/her current position or from a specific address

  • a blank text box used to specify the desired range of the search

  • a drop-down list containing the unit of measurement (m or km)

  • a “Continue” button

2. The user inserts the range in which he/she wants to find a car.

3. The user selects the desired unit of measurement (km is preselected)

4. The user selects the "From current Position" checkbox.

5. The user taps the "Continue" button.

6. The application displays a Google provided map with the found cars

7. The user taps on the object representing the car he/she wants to reserve.

8. A pop-up appears asking the user whether he/she wants to confirm the reservation.

9. The user taps on the "Yes" button.

10. The app confirms to the user that the car has been reserved correctly by displaying the following sentence: “Reservation successful!”.

Exit condition

A notification from the app tells the user that the car has been reserved.

Exceptions

- System/connection failure: if either the system ceases to function or the connection with the user's phone is lost at any time between step 2 and 9 before step 9 is completed, the procedure needs to be performed again from the entry condition.

- Invalid data: in the text box in the first screen the user can insert only a positive integer number. If the user inserts something else, the button “Continue” becomes inactive.

- Pending payment: if the user has yet to pay for his/her last ride, when the user taps the continue button in step 5 another pop-up appears which reads: "Pending ride, cannot confirm reservation". And the user must interrupt the procedure.

-Expired credit card/identity card: if the user's credit card or identity card has expired, when the user taps the continue button in step 5 another pop-up appears which reads: "Expired document, please edit profile". The user must interrupt the procedure.

- Double reservation: if the user has already a reservation which is not expired yet, when the user taps the continue button in step 5 another pop-up appears which reads: "Another reservation is active".

Special requirements

- The user’s phone is required to be connected to the internet

either via Wi-Fi or 3G (or similar) connection.

- The user’s phone needs an activated GPS connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: Retrieve a car

Participating actors

Employee: a person hired by the company

Entry condition

Two employees are informed by a notification from their AdminPowerEnJoy that a car needs to be retrieved.

Flow of events

1. One of the employees opens the AdminPowerEnJoy application.

2. The employee taps on the "Retrieval procedure" button in the newly appeared screen.

3. A new screen appears containing a list of codes representing cars that need to be retrieved and a "Confirm" button and an "Unlock" button.

4. The employee accepts the assignment by tapping on the code representing the car which was specified in the notification and tapping "Confirm".

5. The pair of employees heads to a company car and enters it.

6. They proceed to insert the code of the car to retrieve in the navigator of the company car, which instructs them on how to reach the car's location.

7. Once they arrive to the car they use the equipment in the company's car to recharge the car to retrieve if necessary.

8. The employee who accepted the task uses his app to unlock the car through the "Unlock" button.

9. One of the employees ignites the car and drives to the nearest safe area, while the other follows him with the company car.

10. Arrived at the nearest safe area the employee parks and exits the car that had to be retrieved.

Exit condition

The car parked in a safe area becomes available and this is notified to the app of the employee who accepted the task

Exceptions

- The car cannot be recharged/moved: the employees request the help of a special unit that handles these situations

Special requirements

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Ride a car**

Participating actors

Non-logged person: someone who is not logged in the system

User: a person logged in the system

Entry condition

The Reserve a car use case

Flow of events

1. From the home page of the app the user selects "My reservations".

2. The user selects his/her reservation and clicks the "Unlock" button.

3. The user who unlocked the car or another person enters the unlocked car in the driver seat (he will be referred to as the driver).

4. Up to four other people or three other people and the user enter the car in the passenger seats.

5. The driver ignites the car, and the system starts charging the user.

6. The driver drives the car to his/her destination.

7. The driver stops the car and the system stops charging the user.

8. All the people who entered the car exit the car.

9. Insertion point for the Plug in a car use case.

Exit condition

8. The system automatically locks the car

Exceptions

- Car accident: the driver notifies the company that an accident happened, if possible he completes the ride, otherwise he leaves the car there.

- The employee fails to retrieve the car within 12 hours: the retrieval request is issued again by the system and another employee can accept it

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The user's phone needs an activated GPS connection.

Name: **Pug in a car**

Participating actors

Person: any human person

Entry condition

There is a car parked near a power grid and not plugged in

Flow of events

1. The person plugs the car in the power grid.

2. The system is informed that the car in question is now plugged in.

Exit condition

A light on the power grid displays that the car has been successfully plugged in.

Name: **Logout**

Participating actors

User: a person who is logged in the system.

Entry condition

The user is logged in his/her account.

Flow of events

1. The user taps the "Logout" button in the top right corner of any of the screens which appear when the user is logged in the application.

Exit condition

The application shows the Sign in/Sign up screen.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Browse profile**

Participating actors

User: a person who is logged in the system.

Entry condition

The user opens the application, logs in and taps on "My profile".

Flow of events

1. A screen appears containing a chart with all the user's personal data and an "Edit" button, as well as some statistics regarding the user's rides:

- Total number of rides, followed by a "medal" associated with a number (50+ bronze, 150+ silver, 250+ gold)

- Total distance traveled

- Behavior history: total of negative, positive and neutral behavior cases, associated with a color which ranges from green(good) to yellow(neutral) to red(bad).

2. Extension point for the Edit profile and View payment history use cases.

Exit condition

The user either logs out, closes the app or goes back to the home page.

Special requirements

- The user's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **View payment history**

Participating actors

User: a person who is logged in the system

Entry condition

Extension from the Browse profile use case

Flow of events

1. The user from his/her main profile page taps on the "View payment history" button.

2. A screen appears containing a "Back" button and a chart with five columns:

- Date: the date of the ride

- Start/Finish: the starting time and stopping time of the charging for the ride

- Extras: the extra fee percentage + flat amount if any

- Discounts: the discount percentage if any

- Payment state: OK / PENDING with a tuple for each ride the user has performed.

Exit condition

The user taps the "Back" button and returns to his/her profile.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

Name: **Edit profile**

Participating actors

User: a person who is logged in the system

Entry condition

Extension from the Browse profile use case

Flow of events

1. The user taps the "Edit Profile" button.

2. The chart containing the user's data can now be modified and the "Edit profile" button disappears and a "Confirm" and "Back" buttons appear.

3. The user modifies his/her data.

4. The user taps the "Confirm" button.

Exit condition

The data inserted is valid and the user goes back to the Browse profile use case

Exceptions

-Invalid data: if the user is not able to provide valid data to fill any of the required fields, the procedure cannot be carried out any further, and the user can only stop the procedure by tapping the "Back" button.

Special requirements

- The non-registered person's phone is required to be connected to the internet either via Wi-Fi or 3G (or similar) connection.

- The system's server needs to be operational and responsive enough to allow the procedure to be carried out in a reasonable amount of time.

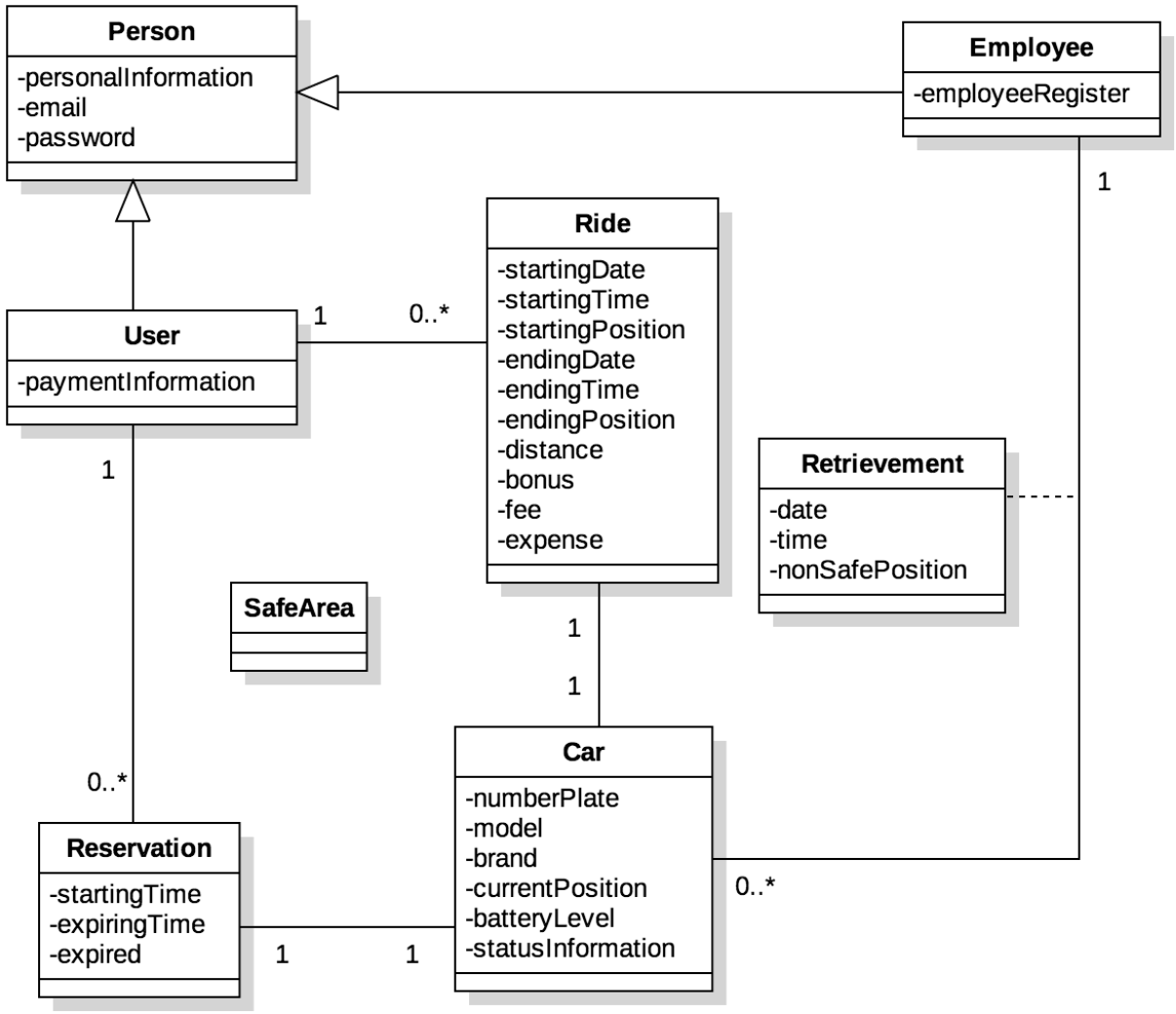
5.2. CLASS DIAGRAM

The following class diagram wants to show which are and how are connected the main information that the system tracks.

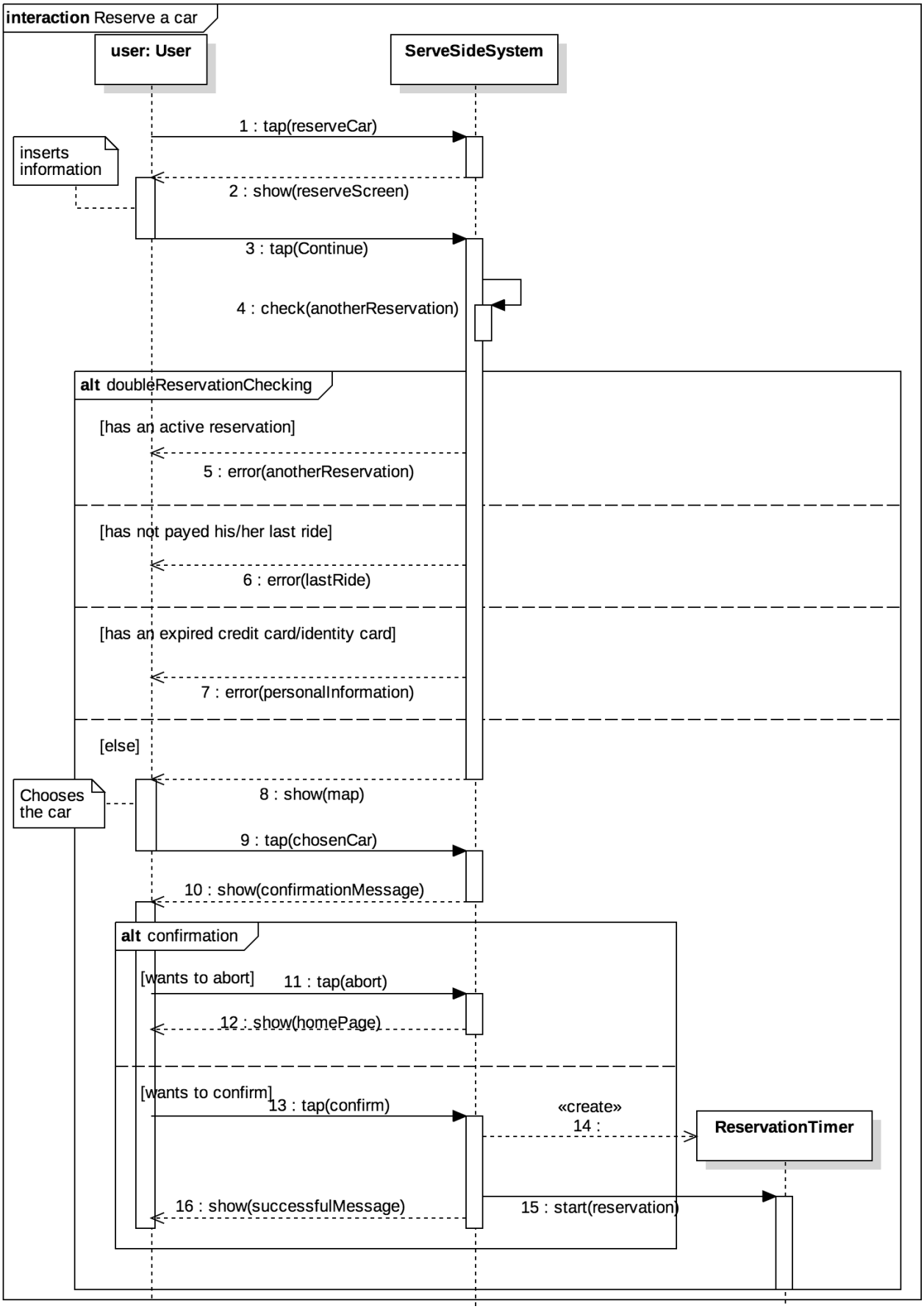
The system has saved records both for users and employees. For each user (in addition to his/her personal information) the system tracks his/her reservations and rides. For each ride, for example, the system keeps track the duration time and the start/ending position.

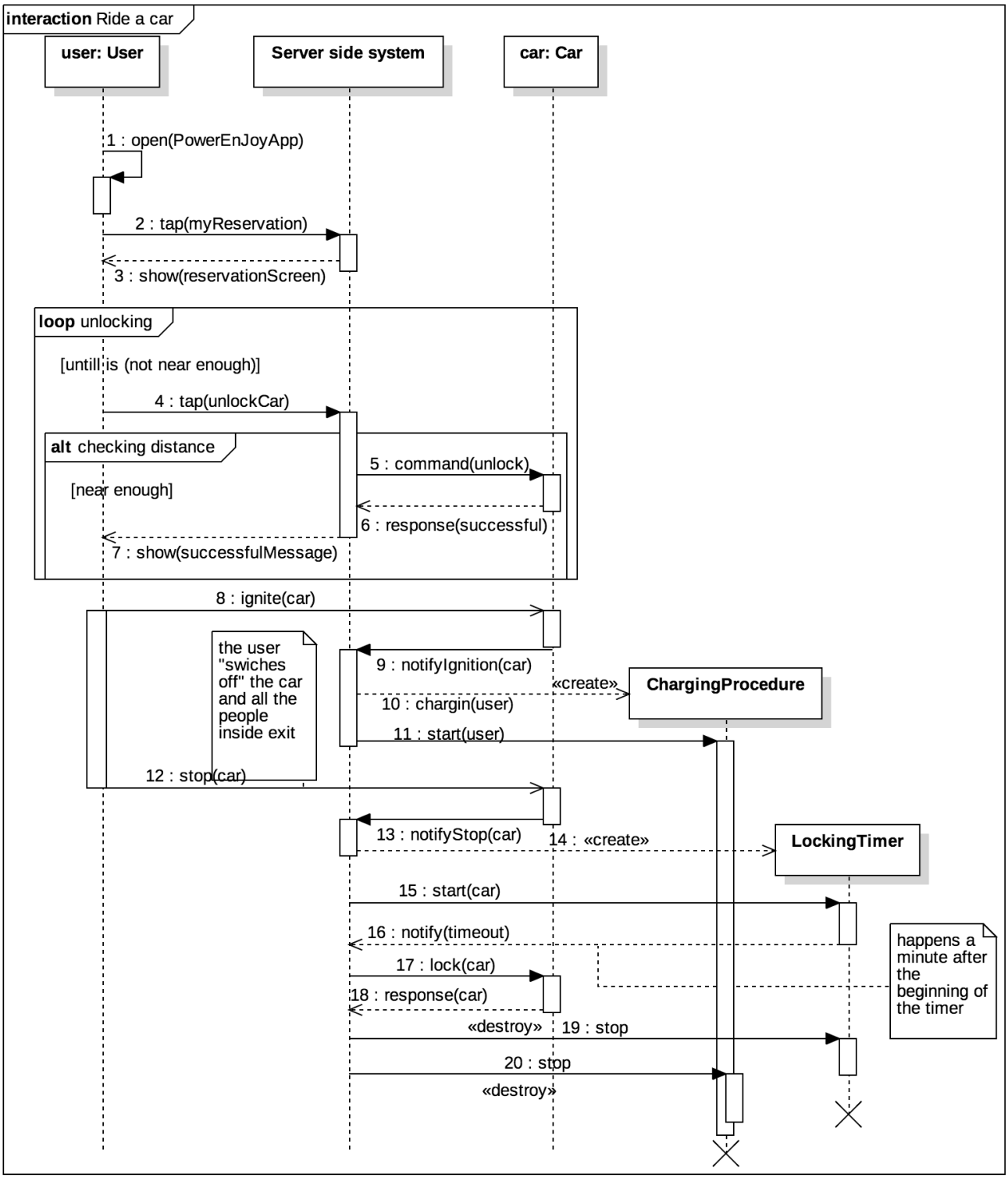
On the other hand, for the employee is important to know which cars he/she has retrieved, when and where.

The system knows also which are the safe areas, with which it can control if a car is well parked (in a safe area) or not.

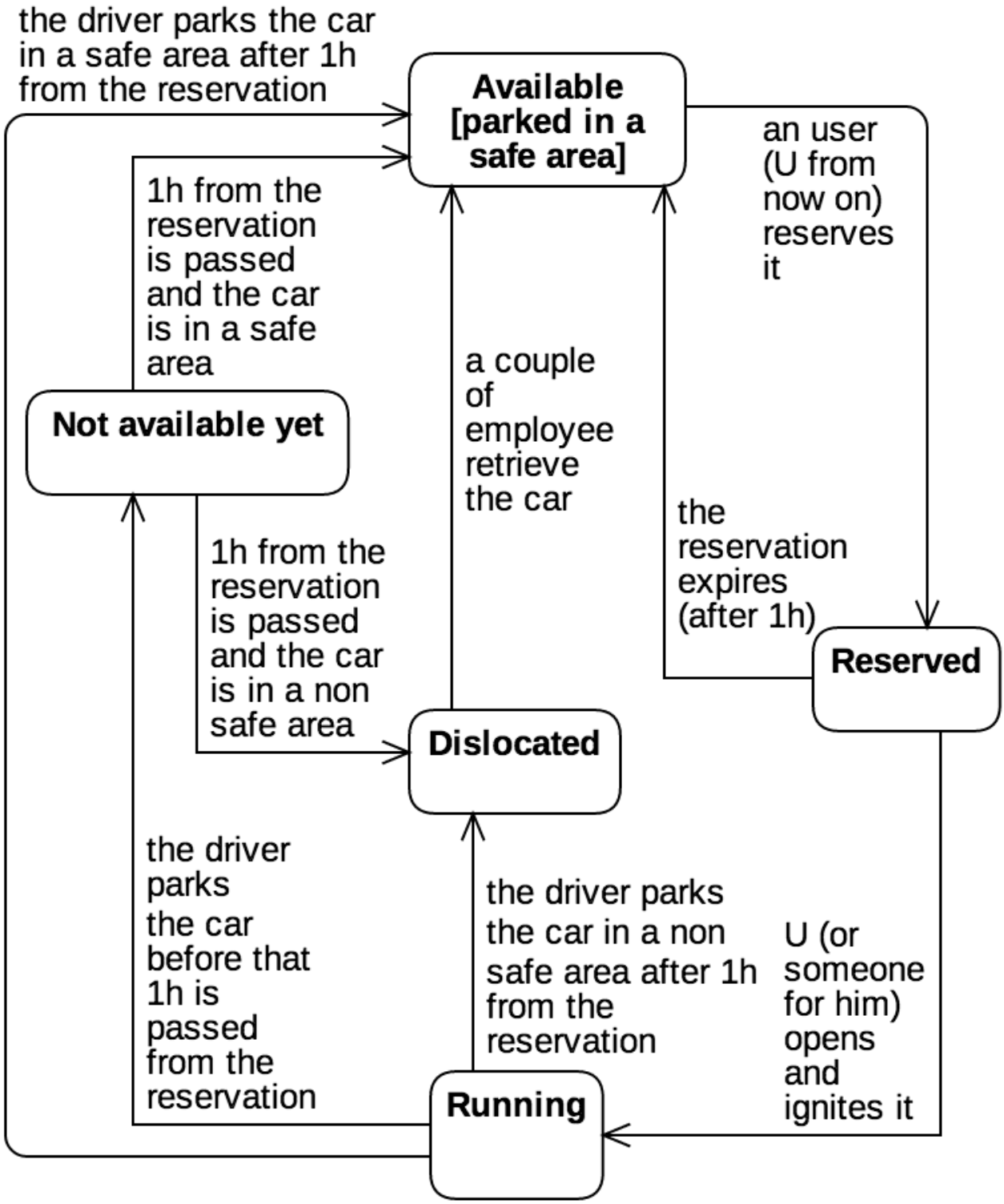


5.3. SEQUENCE DIAGRAMS




interaction Ride a car
user: User
Server side system
car: Car
1 : open(PowerEnJoyApp)
2 : tap(myReservation)
3 : show(reservationScreen)
loop unlocking
[untill is (not near enough)]
4 : tap(unlockCar)
alt checking distance
[near enough]
5 : command(unlock)
6 : response(successful)
7 : show(successfulMessage)
8 : ignite(car)
the user "swiches off" the car and all the people inside exit
9 : notifyIgnition(car)
«create»
ChargingProcedure
10 : chargin(user)
11 : start(user)
12 : stop(car)
13 : notifyStop(car)
14 : «create»
LockingTimer
15 : start(car)
16 : notify(timeout)
happens a minute after the beginning of the timer
17 : lock(car)
18 : response(car)
«destroy»
19 : stop
20 : stop
«destroy»

5.4. STATECHART DIAGRAM



6. ALLOY MODELLING

6.1. MODEL 1

```
//ALLOY MODEL FOR GOALS [G2.0], [G2.1] AND RELATIVE REQUIREMENTS
//----------------------THE WORLD ----------------------------------------------------
sig User{
        //goal-related user relations
        userIsAt : one Location,
        canFind : set Car,
        specifiedAddress : lone Address,

        //requirements-related user relations
        accesses : one PowerEnJoyApp,
        uses : set Function
}{
        //a user can find cars only using the app (in the sense of finding as stated by the
        goal )
        some canFind iff ReserveACar in uses

        //a user specifies an address only in the contest of the ReserveACar functionality ( in
        this context )
        one specifiedAddress implies ReserveACar in uses
}

//only the addresses specified by a user are shown in this model
fact  relevantAddress{ no a : Address | no u : User | a in u.specifiedAddress}
sig Car{
        carIsAt : one Location,
}
sig AvailableCar in Car{}
sig Location{
        inRange : set Location
}
```

```
sig Address{
	isAt : one Location
}
//properties of  a range
fact  RangeIsReflexive{ all l : Location | l in l.inRange }
fact  RangeIsSymmetric{ all l1,l2 : Location | l2 in l1.inRange implies l1 in l2.inRange }

//	[G2.0]: The user is able to find the location of all available cars within a certain range from their current location.
//	[G2.1]: The user is able to find the location of all available cars within a certain range from a specified address.
assert G2 { all u : User, c : AvailableCar | ReserveACar in u.uses and (c.carIsAt in u.userIsAt.inRange and no u.specifiedAddress) or (one u.specifiedAddress and c.carIsAt in u.specifiedAddress.isAt.inRange) iff c in u.canFind }

//---------------------- THE MACHINE ----------------------------------------------------------------
one sig PowerEnJoyApp{
	has : set Function
}{
	//the app has every one of its functions
	all f : Function | f in has
}
//[R2.1]: The "Reserve a car" function can be accessed by the user from the home page of the app
fact  userCanAccessAllFunctions{ all u : User | u.uses in u.accesses.has}

//[R2.2]: The "Reserve a car" function allows the user to select a range : in this model we assume the range considered to be the range of choice
//[R2.3]: The system acquires the user's current position through the GPS coordinates of the user's phone
fact  userCoordinatesAcquired{ all u : User | ReserveACar in u.uses iff (one r : ReserveACar | u in r.acquiresUser/*GPS*/.signalsForUser ) }
```

```
abstract sig Function{}
one sig ReserveACar extends Function{
                acquiresCars : set GPSCar,
                acquiresUser : one GPSUser,
                acquiresAddress : lone GPSAddress
}{
        acquiresAddress.signalsForAddress =  acquiresUser.signalsForUser.specifiedAddress

        //[R2.4]: The system tracks all available cars' current position through their GPS
        coordinates acquiresCars.signalsForCar = AvailableCar
        //[R2.5]: The "Reserve a car" function allows the user to select a starting position
        for the search, which can be either their current location or a given address
        no acquiresAddress and (all c : Car | c in acquiresUser.signalsForUser.canFind iff
                c.carIsAt.inRange = acquiresUser.signalsForUser.userIsAt.inRange ) or
        one acquiresAddress and (all c : Car | c in acquiresUser.signalsForUser.canFind iff
                c.carIsAt.inRange = acquiresAddress.signalsForAddress.isAt.inRange )
}

//[R2.6]: When the user confirms the inserted parameters the search is carried out and the
"Reserve a car" function displays to the user
// the data of the search acquired from the system in a Google provided map : this model
shows the instant after confirmation
//this fact  states that GPS coordinates are in general different for different objects
fact  uniqueGPS{ (all g1, g2 : GPSCar | g1 != g2 iff g1.signalsForCar != g2.signalsForCar) and
        (all g1, g2 : GPSUser | g1 != g2 iff g1.signalsForUser != g2.signalsForUser) and
        (all g1, g2 : GPSAddress | g1 != g2 iff g1.signalsForAddress != g2.signalsForAddress)}

// this fact  conveys the meaning of "the user can find only the cars tracked via GPS by the
system"
fact  canFindAcquiredCarsOnly{ all r : ReserveACar |
        r.acquiresCars.signalsForCar = r.acquiresUser.signalsForUser.canFind }
```

```
abstract sig GPS{}
sig GPSCar extends GPS{
        signals : one Location,
        signalsForCar : one Car
}{
        //GPS coordinates exist in the model as long as a "ReserveACar" functionality tracks
        them
        one signalsForCar iff (one r : ReserveACar | this in r.acquiresCars )
}
sig GPSAddress{
        signals : one Location,
        signalsForAddress : one Address
}{
        //GPS coordinates exist in the model as long as a "ReserveACar" functionality tracks
        them
        one signalsForAddress iff (one r : ReserveACar | this = r.acquiresAddress )
}
sig GPSUser{
        signals : one Location,
        signalsForUser : one User
}{
        one signalsForUser iff one signalsForUser.uses

        //GPS coordinates exist in the model as long as a "ReserveACar" functionality tracks
        them
        one signalsForUser iff (one r : ReserveACar | this = r.acquiresUser )
}
//[D3.0]: The GPS coordinates of the cars received by the system always correspond to the
actual positions of the cars.
//[D3.1]: The GPS coordinates of a user received by the system always correspond to the
actual position of the user.

fact  GPSUserCorrect{ all g : GPSUser | g.signals = g.signalsForUser.userIsAt}
fact  GPSCarCorrect{ all g : GPSCar | g.signals = g.signalsForCar.carIsAt}
```

fact GPSUserCorrect{ **all** g : GPSAddress | g.signals = g.signalsForAddress.isAt}

check G2 **for** 4

6.1.2. RESULTS

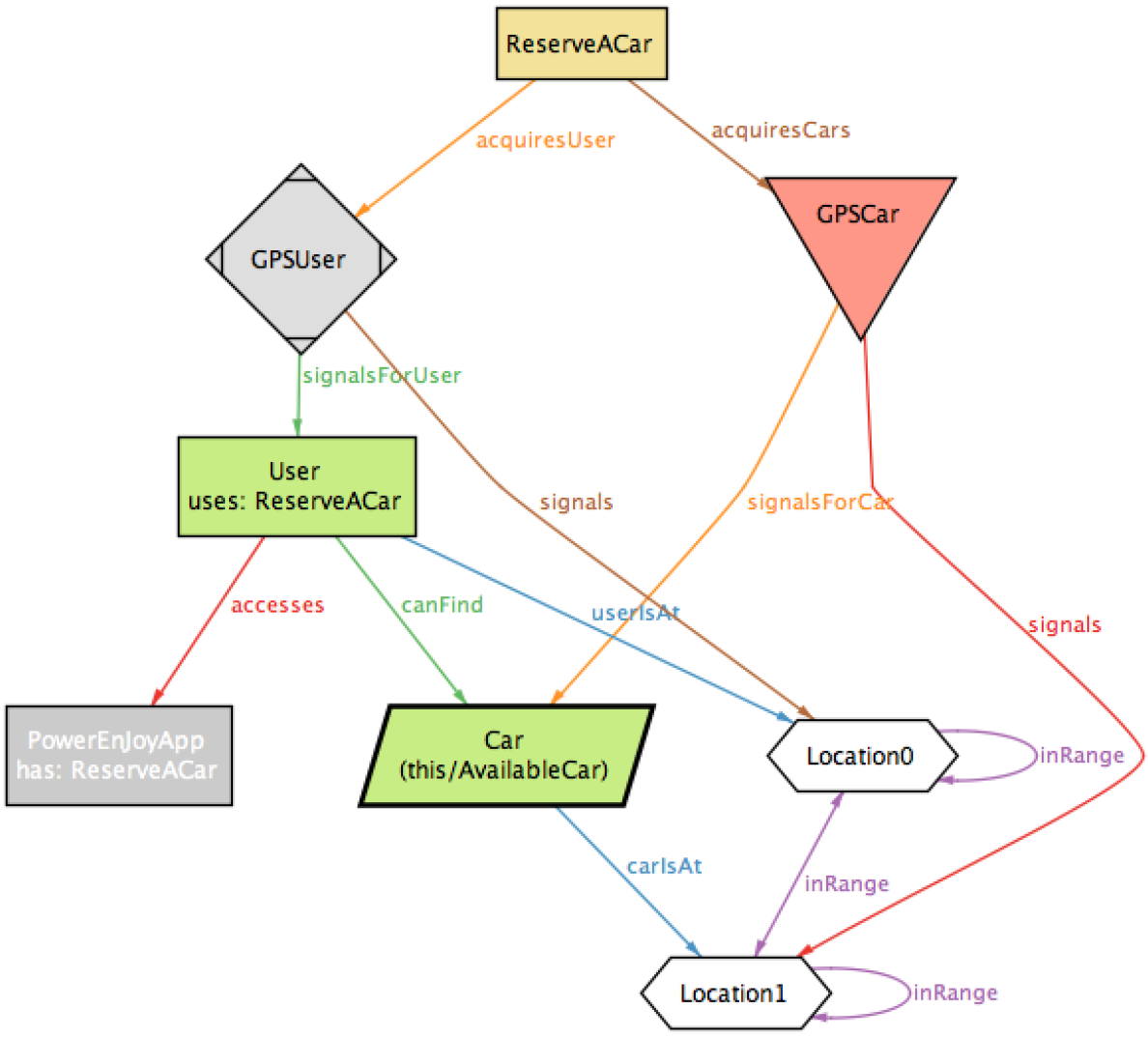
Executing "Check G2 for 4"

Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20

3763 vars. 269 primary vars. 7492 clauses. 351ms.

No counterexample found. Assertion may be valid. 150ms.

6.1.3. WORLD GENERATED (for 2)



6.2. MODEL 2

```
//[G6.0]: From the moment of ignition, the user is charged for a constant amount of money per minute

//[G6.1]: The [user] driver is notified of the current charges through a screen on the car.

//[G7.0]: The charging of the user stops as soon as the driver parks the car in a safe area and exits from it.

//[G7.1]: The car is automatically locked as soon as the driver parks the car in a safe area and exits from it.

//[D7.1]: If the sensors in the car detect no one inside, there are no people in the car

//ALLOY MODEL FOR GOALS *** AND RELATIVE REQUIREMENTS


//----------------------THE WORLD ----------------------------------------------------------------------------


sig Person{}


sig User in Person{
        isChargedFor : lone  Car,


}


sig Driver in Person{
        isNotified : set ConstantCharge,
        isDriving : lone  Car,
        hasParked : lone  Car
}{
        //a driver is driving if and only if has not parked a car
        one  isDriving iff no hasParked
        //a driven car is ignited
        isDriving  in IgnitedCar
        //a parked car is stopped
        hasParked  in StoppedCar
}
```

```
sig ConstantCharge{}

abstract sig Car{
        hasIn : set Person,
        detects : set Sensor
}{
        // a sensor is detected if and only if is in this car
        all s : Sensor | s in detects iff this in s.isIn
}

sig Sensor{
        detectedPerson : one  Person,
        isIn : one  Car
}

sig IgnitedCar extends Car{
        displays : set ConstantCharge
}

sig StoppedCar extends Car{
        parkedIn: one  Area
}

sig LockedCar in Car{}{
        this not in UnlockedCar
}
sig UnlockedCar in Car{}{
        this not in LockedCar{}
}

fact  { LockedCar + UnlockedCar = Car}
```

```
abstract sig Area{}
sig SafeArea extends Area{}

//no person can be in two cars at the same time
fact  { all c1, c2 : Car | c1 != c2 implies (no p : Person | p in c1.hasIn and p in c2. hasIn) }

//a driver must be in a car to drive it
fact  { all d : Driver, c : Car | c in d.isDriving implies d in c. hasIn }

//no two sensor detect the same person
fact { all s1, s2 : Sensor | s1 !=  s2 iff ( no p : Person | p in s1.detectedPerson and p in
s2.detectedPerson ) }

//a sensor can only detect a person inside the car
fact { all c : Car | c.detects.detectedPerson in c. hasIn }

//if someone has parked a car, that car is parked somewhere
fact { all c : StoppedCar, d : Driver | c in d.hasParked implies one  c. parkedIn }

//a parked car has been parked by only one person
fact { all d1, d2 : Driver | d1 != d2 implies ( no c : Car | c in d1.hasParked and c in
d2.hasParked ) }

//only the safe areas with some car parked in them are considered in the model
fact { all sa : SafeArea | one  c : Car | sa in c. parkedIn }

//all ignited cars have a driver
fact { all c : IgnitedCar | one  d : Driver | c in d.isDriving}

//all constant charges considered in the model are displayed by some car
fact { all ch : ConstantCharge | one  c : Car | ch in c.displays}

//a driver is notified of what the car he is driving displays
```

```
fact { all d : Driver | d.isNotified = d.isDriving.displays }

//for the sake of simplicity,  no driver who has parked a car can be in another car
fact { no c1, c2 : Car | ( one  d : Driver | d in c1. hasIn and c2 in d.hasParked ) }

//[D7.1]: If the sensors in the car detect no one inside, there are no people in the car
// in this model we consider the weakest possible sensor accuracy which still allows to satisfy the goal, in the form of D7.1
fact  sensorAccuracy{ all c : Car | no c.detects iff no c. hasIn }

assert G6_0 {  all c : Car | c in IgnitedCar iff ( one  u : User | c in u.isChargedFor)}
assert G6_1 { all d : Driver | some d.isNotified iff ( one  c : Car, u : User | c in u. isChargedFor and c in d.isDriving) }
/*see goal interpretation for [G7.0]*/
assert G7 { all u : User, c : Car | c. parkedIn in SafeArea and ( no c. hasIn) implies ( c not in u. isChargedFor and c in LockedCar ) }

check G6_0
check G6_1
check G7
pred p{}
run p for 6

//---------- THE MACHINE ----------
one sig System{
        lastReservation : set User -> one  Car
}

//[R6.1]: When a car is ignited the system starts charging the last user who reserved the car
fact { all u : User, s : System | one  u.isChargedFor iff (one  c : Car | s -> u -> c in lastReservation and c in u.isChargedFor)}
fact { all c : Car, s : System, u : User | ( c in IgnitedCar and ( s -> u -> c in lastReservation ) implies c in u. isChargedFor ) }
```

```
// a car can be ignited only if a user had reserved it
fact { all c : Car, s : System |  c in IgnitedCar iff ( one  u : User | s -> u -> c in lastReservation )}

//a car displays charges if ignited
fact { all c : Car | c in IgnitedCar  implies some c.displays}

// in the system each user has only one "last reservation"
fact { all c : Car, s : System | lone  u : User | s -> u -> c in lastReservation }

//[R6.2]: When the charging starts, the display on the car shows a "Current charge" field
//with a number representing the current total charge, which starts from 0
fact { all u : User | one  u.isChargedFor iff ( one  c : Car | c in u.isChargedFor and some c.displays ) }

//[R6.3]: Once a minute the "Current charge" value is incremented by a set amount : this requirement is not modeled in this model

//[R7.1]: When a car is stopped and the sensors in the car detect no one inside,
// if a user was being charged for the car the system stops charging him/her.
//[R7.2]: One minute after a car has been stopped and the sensors in the car detect no one inside, the system locks the car
fact { all c : StoppedCar | no c. hasIn iff ( no u : User | c in u. isChargedFor ) and c in LockedCar }
```

6.2.2. RESULTS

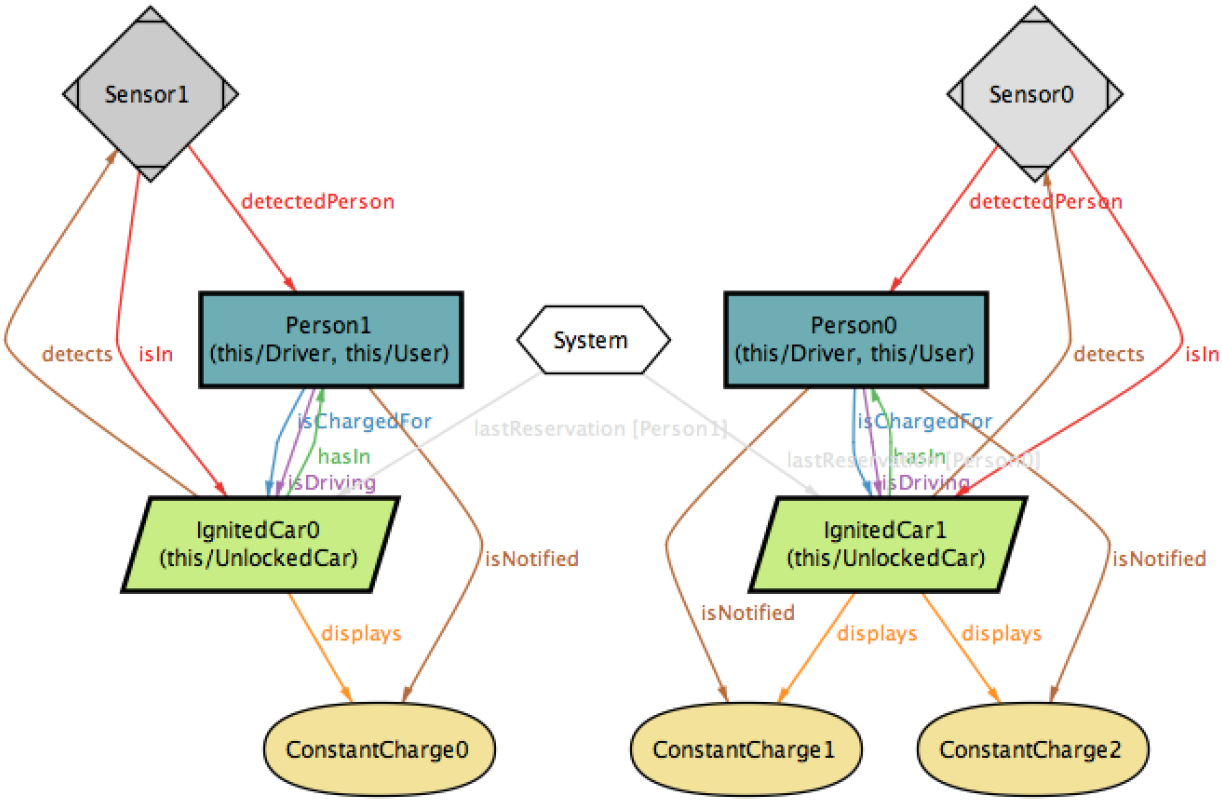
Executing "Check G6_0"

Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20

1837 vars. 132 primary vars. 3258 clauses. 280ms.

No counterexample found. Assertion may be valid. 107ms.

6.2.3. WORLD GENERATED (for 3)



7. HOURS OF WORK

Peverelli Francesco: 47h

Federico Reppucci: 35,5h

Together: 19,5h