PowerEnJoy

PEVERELLI FRANCESCO
REPPUCCI FEDERICO

ACTOR IDENTIFICATION

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

PARKING AREAS

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

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

CAR RETRIEVAL

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.

IDENTIFICATION AND PAYMENT

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.

TEXT ASSUMPTIONS

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

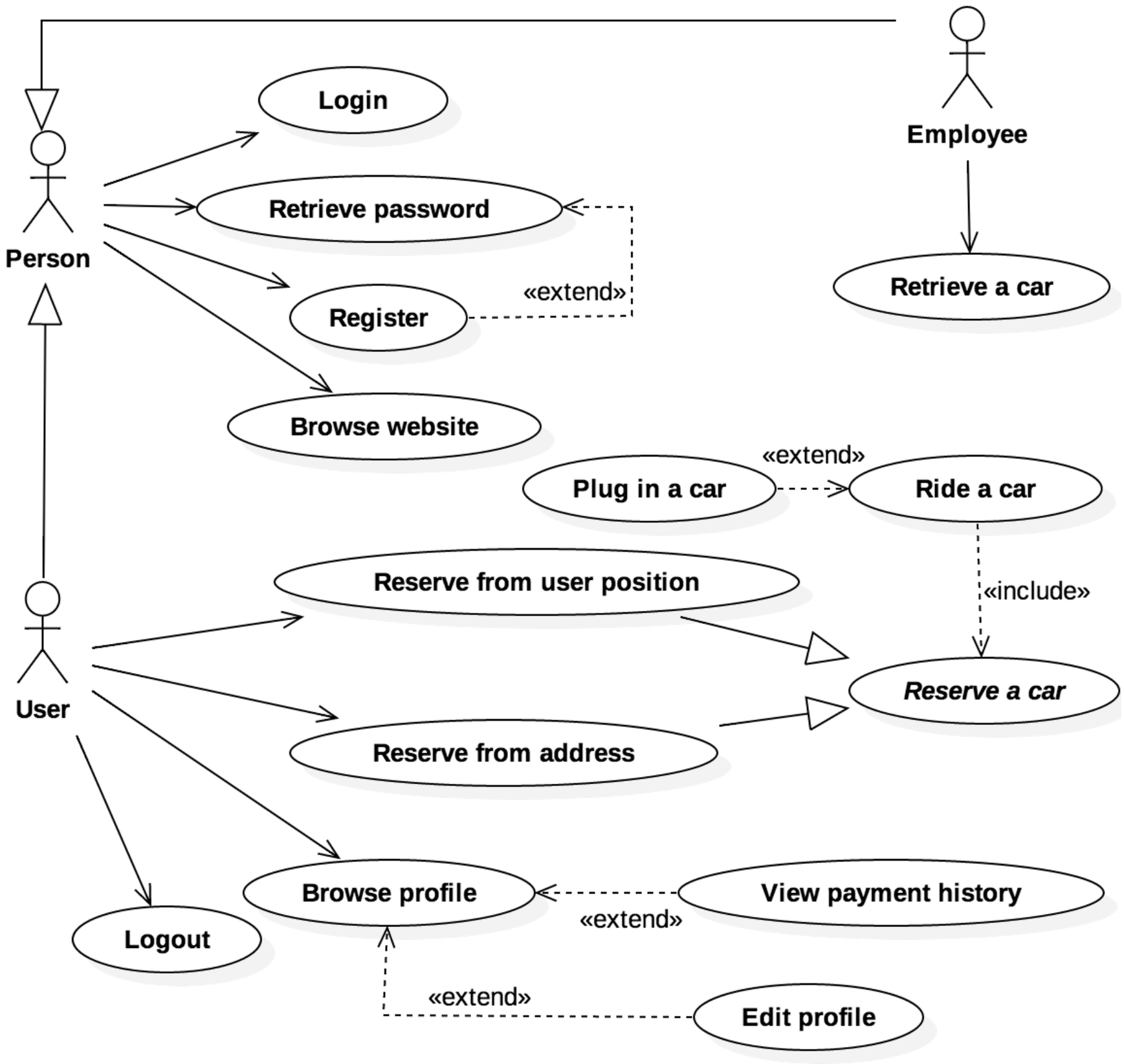
[T2]: We assume that the actual transaction with the payment service provider is performed only once, at a later time after the end of each the ride

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

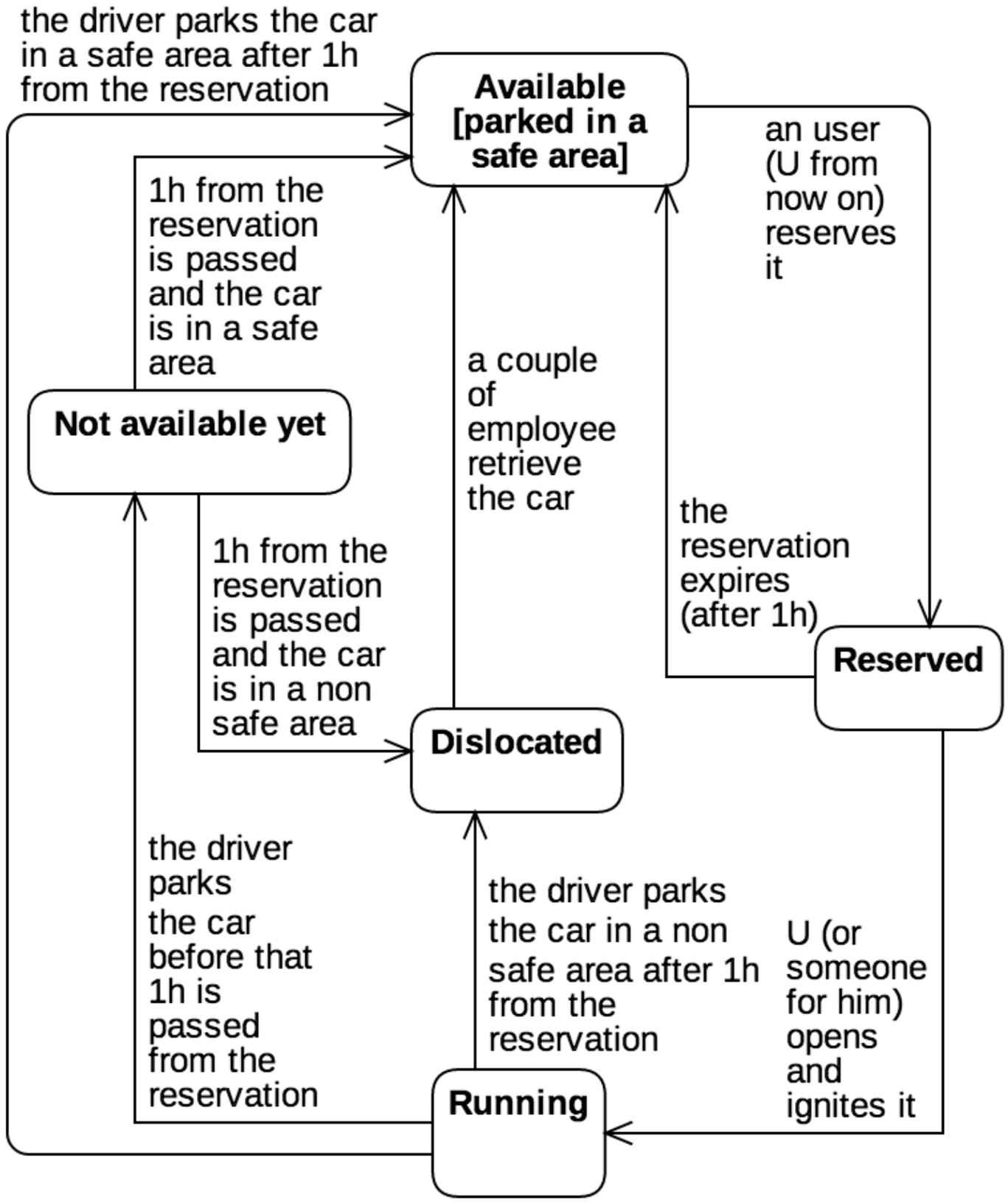
[T3.1]: Any extra charge prevents any discount from being applied on a ride.

[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

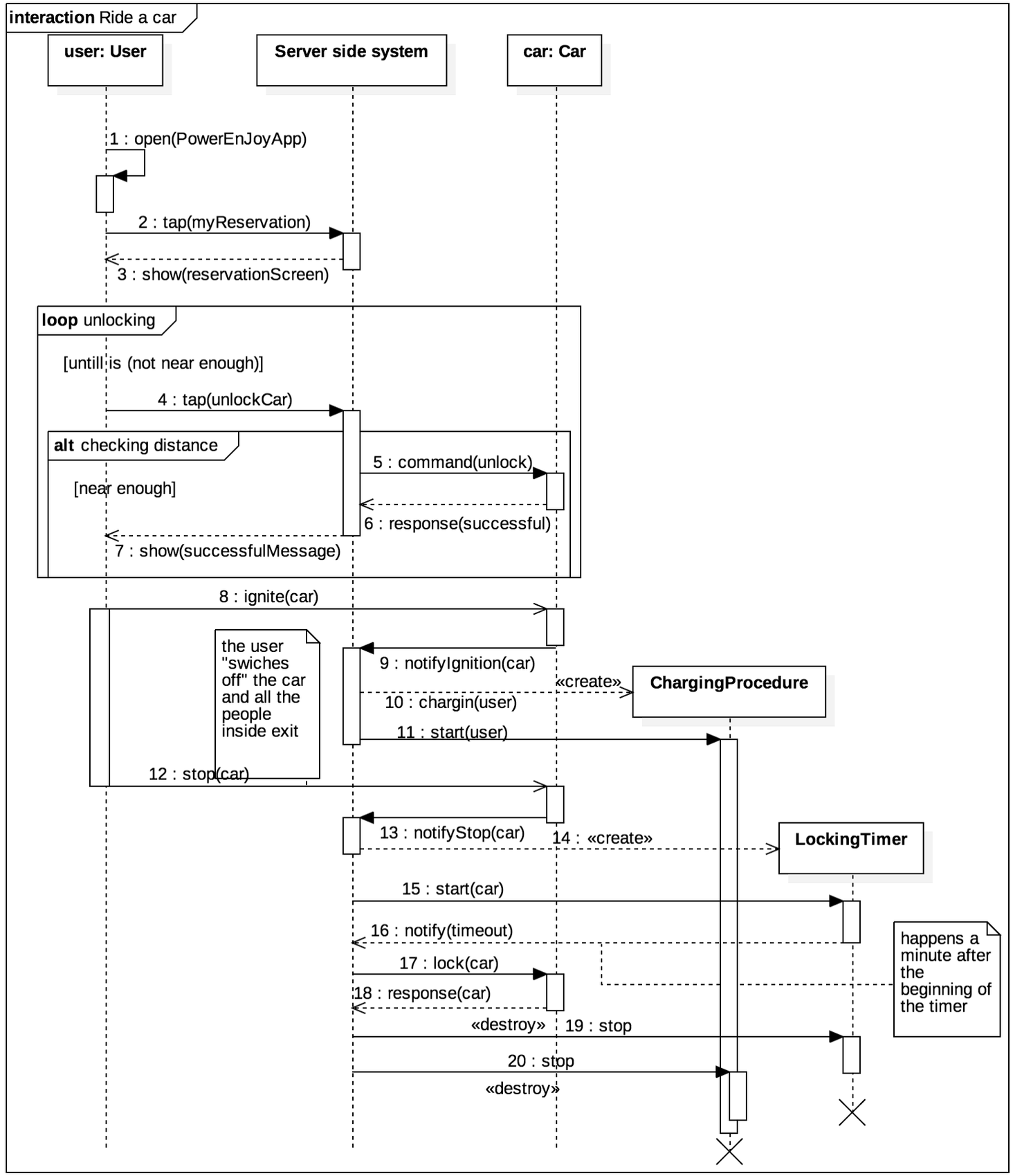
USE CASE DIAGRAM

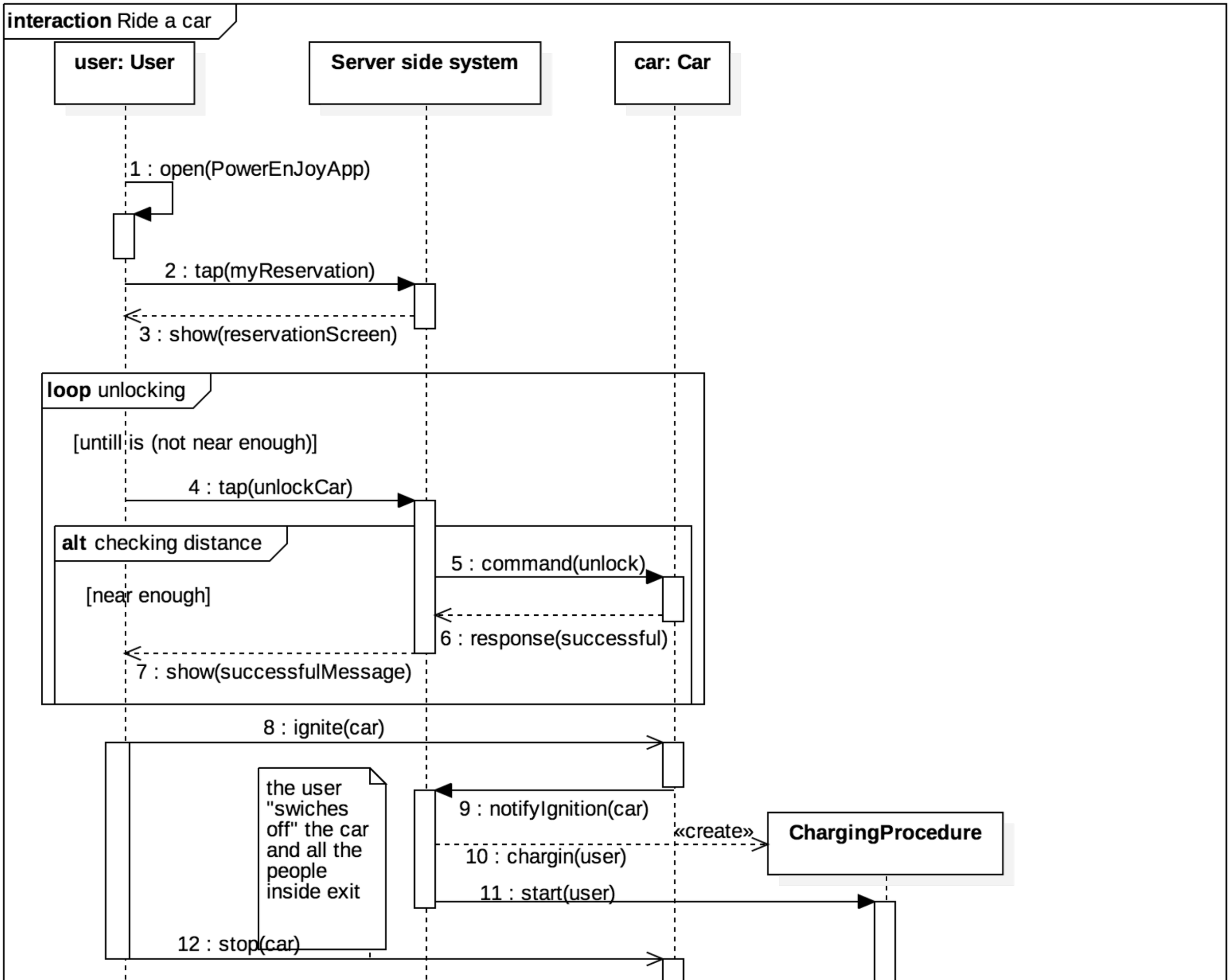


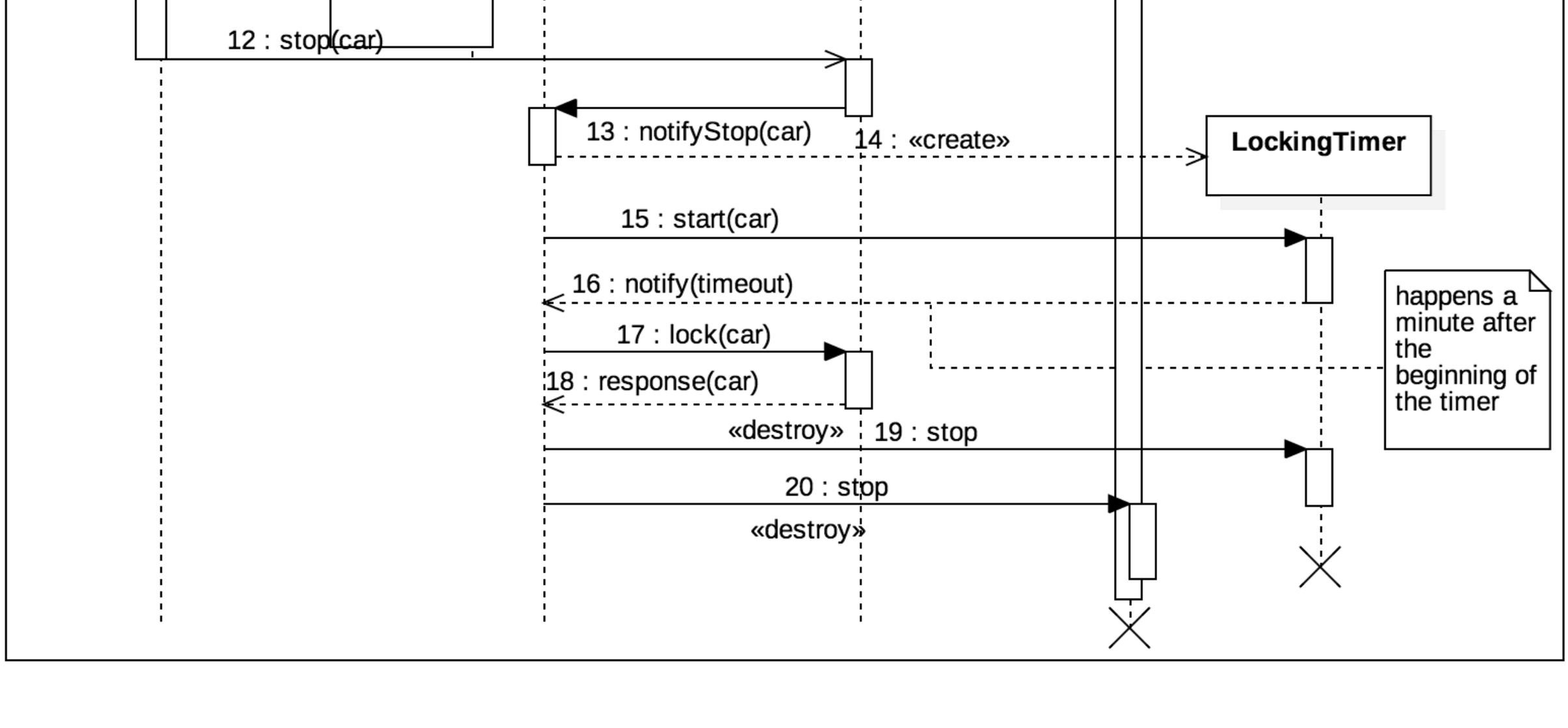
STATECHART DIAGRAM - STATES OF A CAR



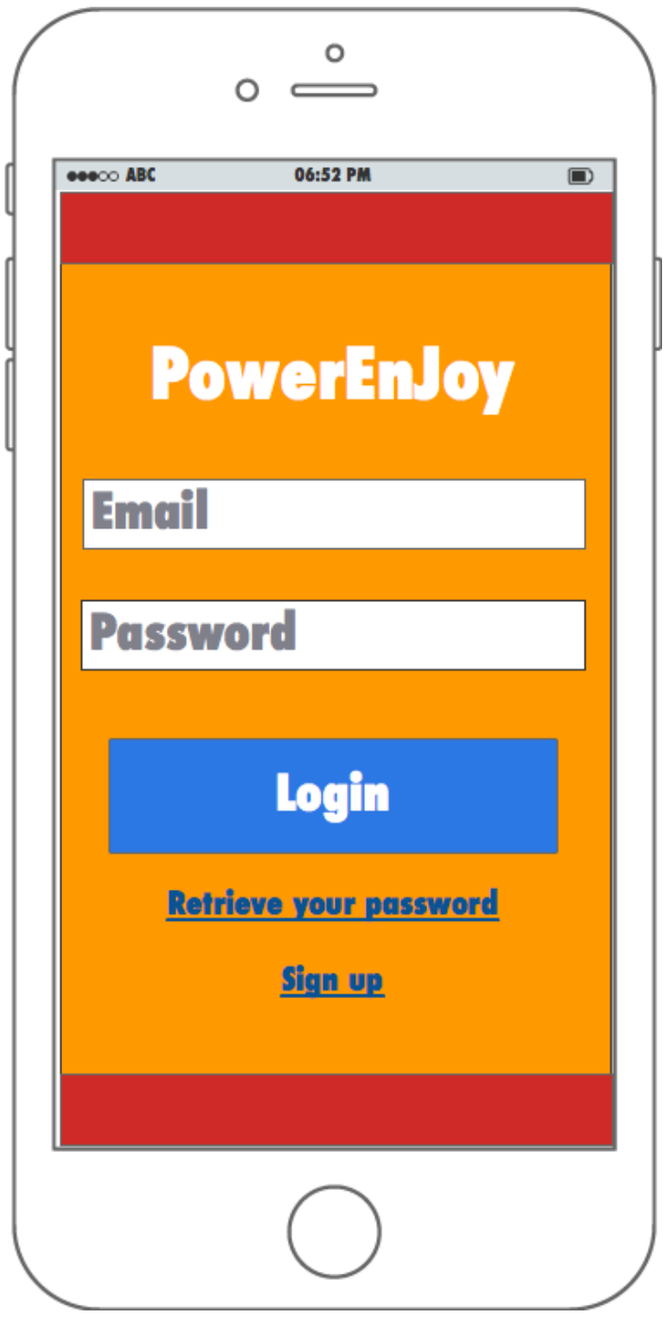
RIDE A CAR - SEQUENCE DIAGRAM

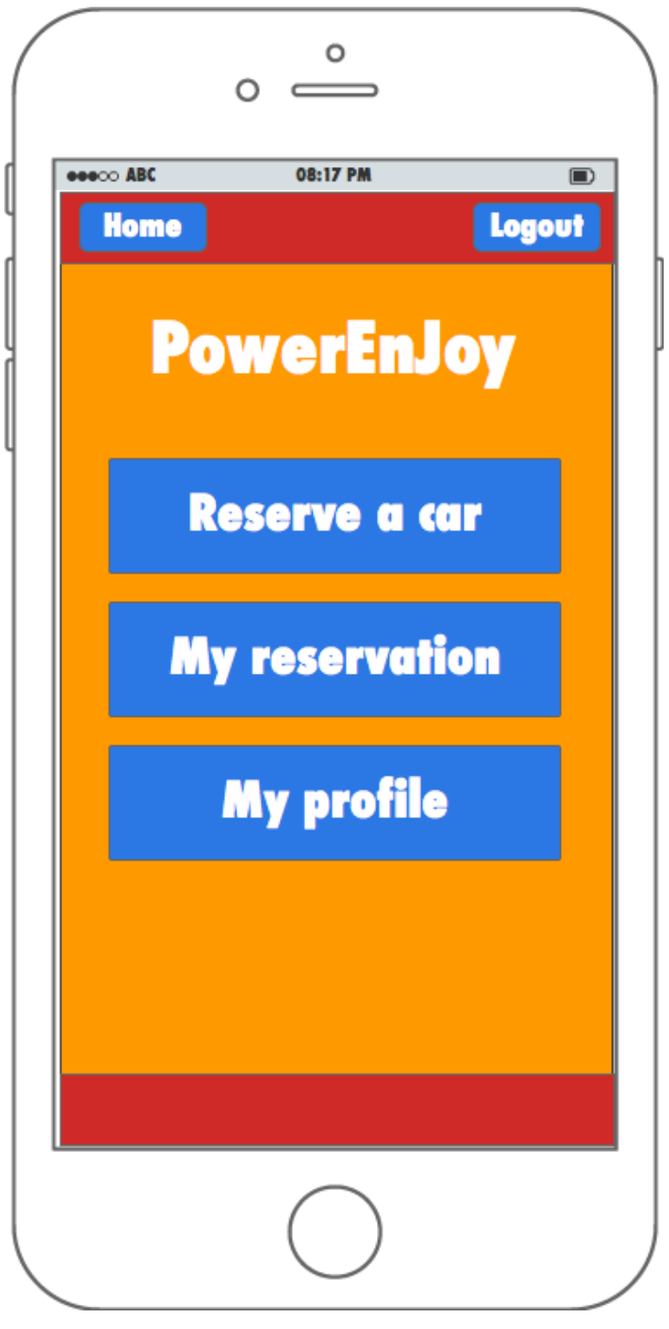


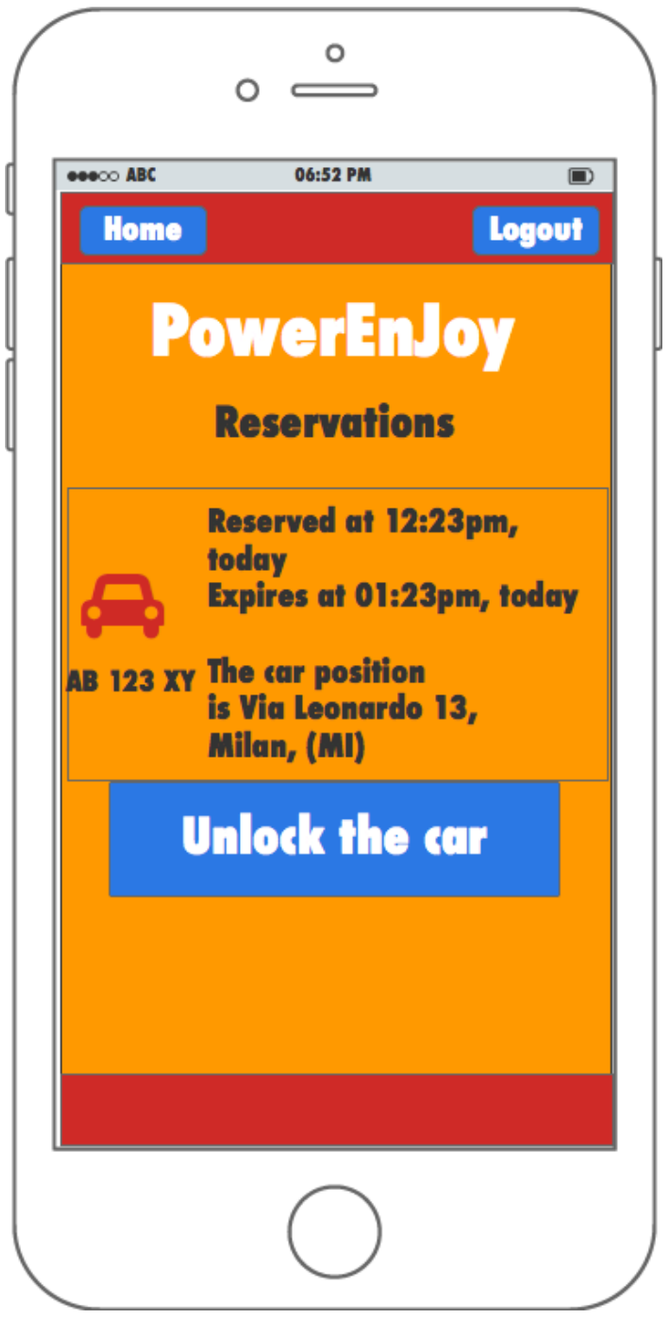

interaction Ride a car
user: User
Server side system
car: Car
1 : open(PowerEnJoyApp)
2 : tap(myReservation)
3 : show(reservationScreen)
loop unlocking
[untill is (not near enough)]
4 : tap(unlockCar)
alt checking distance
[near enough]
5 : command(unlock)
6 : response(successful)
7 : show(successfulMessage)
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)


12 : stop(car)
13 : notifyStop(car)
14 : «create»
LockingTimer
15 : start(car)
16 : notify(timeout)
17 : lock(car)
18 : response(car)
«destroy»
19 : stop
20 : stop
«destroy»
happens a minute after the beginning of the timer

USER INTERFACE

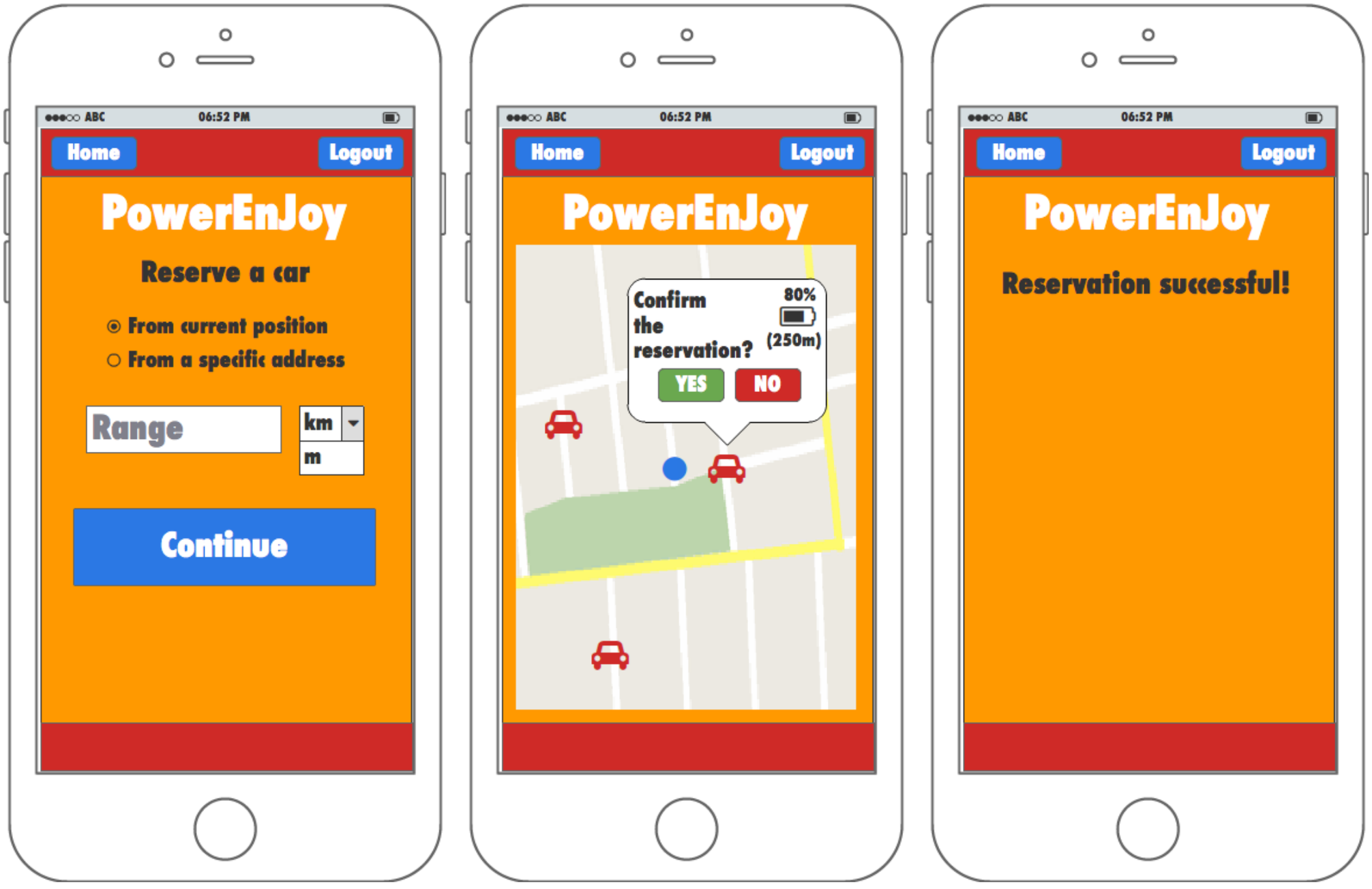




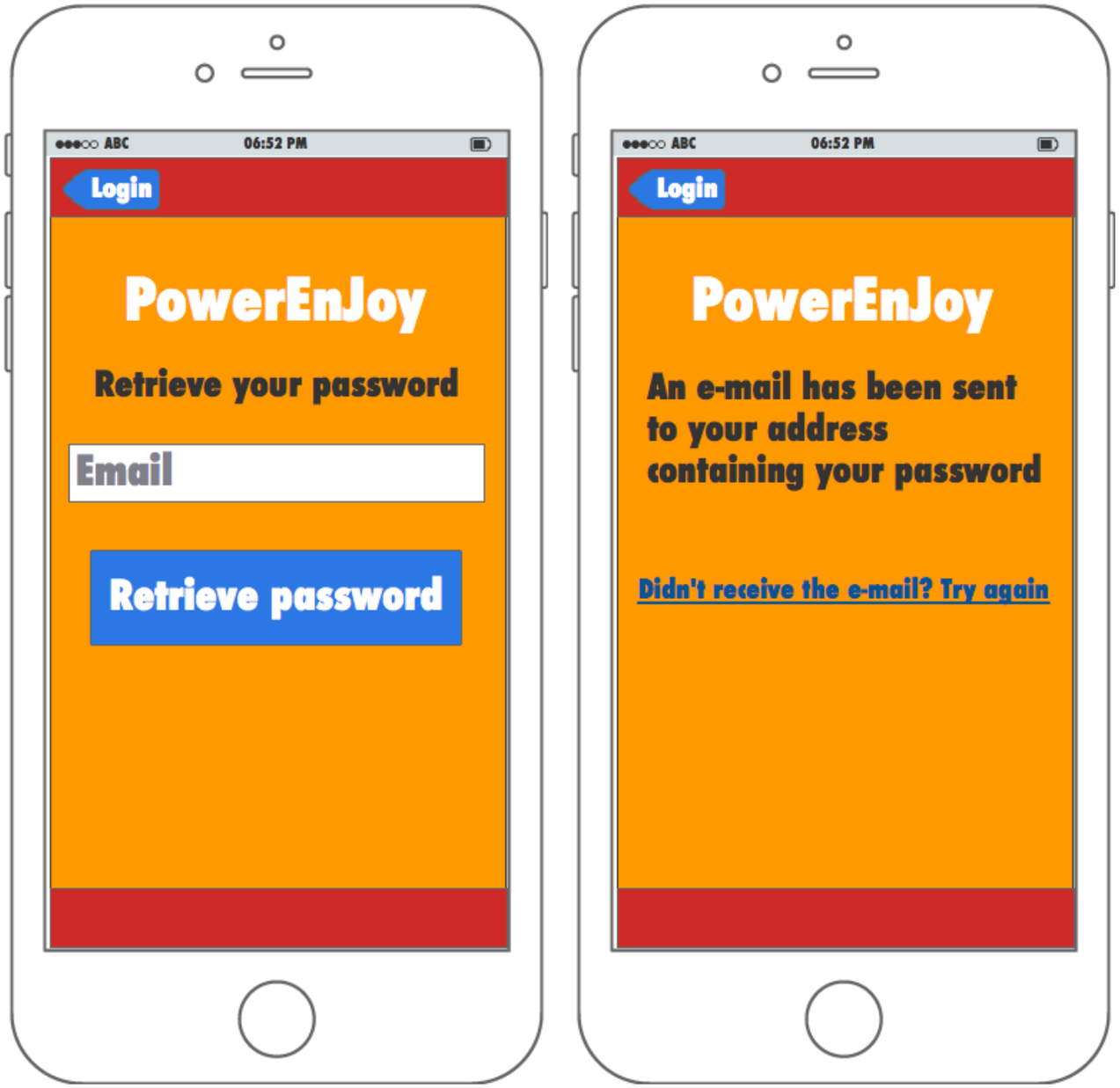


USER INTERFACE

USER INTERFACE



USER INTERFACE

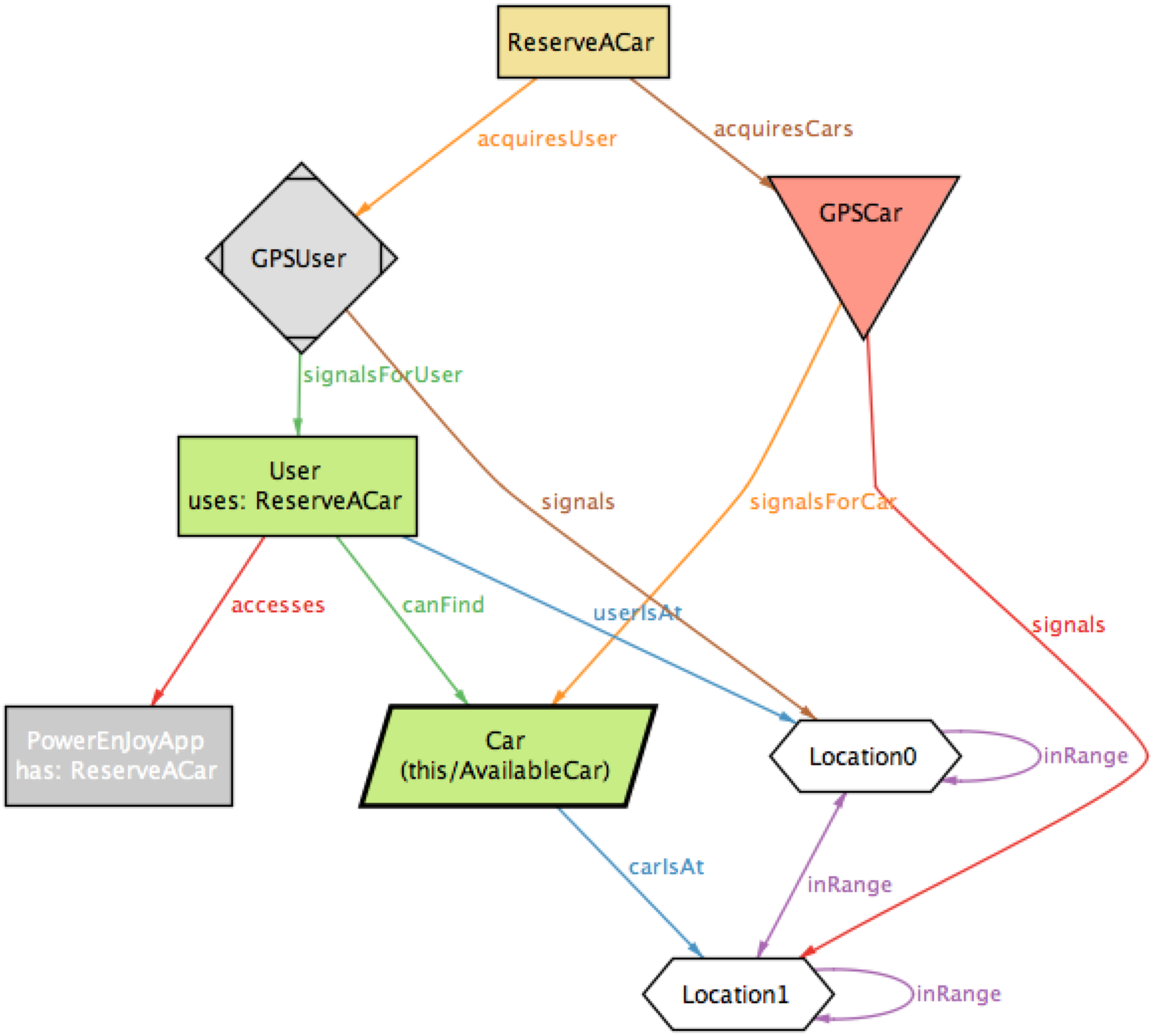


```
//ALLOY MODEL FOR GOALS [G2.0], [G2.1] AND RELATIVE REQUIREMENTS
//----------------------THE WORLD ----------------------------------------------------

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

}{

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 ) }

//the app has every one of its functions

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
```

GENERATED WORLD



GOALS - REQUIREMENTS - DOMAIN ASSUMPTIONS

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

GOALS - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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.

GOALS - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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.

GOALS - REQUIREMENTS - DOMAIN ASSUMPTIONS

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 - REQUIREMENTS - DOMAIN ASSUMPTIONS

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

GOALS - REQUIREMENTS - DOMAIN ASSUMPTIONS

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)

NON-FUNCTIONAL REQUIREMENTS

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

ANY

QUESTION?