M.Sc. in Computer Science and Engineering
Software Engineering 2

Requirements Analysis and Specifications Document
(RASD)

Version 1.3
Released on: 12th November 2016

Authors:

Marilena Coluccia
Burcu Cesur
Mustafa Çeçe

Reference Professor:

Elisabetta Di Notto

CONTENTS

1. INTRODUCTION

1.1.DESCRIPTION OF THE GIVEN PROBLEM

A car sharing project named “PowerEnJoy” will be implemented which is a service based on mobile application with targeting car necessity in a precise instant of time. The system is designed to register new customers. Registered customers are able to reserve a electrical car with choosing from the available cars according to their informations. Reservation cancellations allowed under a charge. The system offers solutions with either identifying the current GPS location of the customer or a specific location which is given by the customer.

1.2.GOALS

When possible customers are taking into account, PowerEnJoy should provide a system to the specified users with features below:

Customers:

- Registration to the system
- Login to the system
- Log out from the system
- Allowed to manage profile information
- Allowed to reserve an available single car, in certain geographical region, for up to one hour before they pick it up.
- Allowed to make a cancellation of the reservation
- Allowed to know the current charges through a screen on the car after system starts charging the customer for a given amount of money
- Allowed finding the locations of available cars within a certain distance from the current location of the customer, at the current time.
- Allowed to list the properties (battery level, exact address, estimated distance can be made) of the available cars

- Allowed to see the safe area in the map that the car can be parked which is predefined by the management system
- Allowed to tell the system that customer is nearby so the system unlocks the car in order to enter
- Allowed to share the drive with other people to get a discount with the censors of the car
- Allowed to see special parking areas to recharge the car by taking care of plugging the car into the power grid in order to get a discount.
- Allowed to enable the many saving option, customer input final destination and the system provide the station information where to leave the car to get a discount.

Guest:

- Allowed to make a registration

1.3.DOMAIN PROPERTIES

Below conditions are supposed that are hold in the analyzed world

- GPS datas always give the rig
- ht coordinates.
- The GPS of the cars are always on.
- The system shows instantaneous current data.
- Customer that enters the car will start the engine in an amount of time.
- Cars have more battery than informed to leave a rescue chance.
- Access permission to unlock the car is given immediately after the reservation is made and expires after one hour from the reservation time.
- Customer's GPS and internet are always on.
- The system is able to detect the reserved car by matching the car's and the customer's GPS so the customer is able to tell he/she is nearby to unlock and enter in the car.
- Battery detection in the car system is always on.

- Power grid station areas provide location boundaries data to the system that can be matched with the car's GPS to determine if the cars is there or not.
- The screen on the car which shows the information is always on.
- Available cars are in public areas and reachable.
- Single car can be reserved in one hour there is no possibility to reserve more than one.
- The customer's credit card validation is already checked under some protocols during registration to be able to perform payment.
- The system remove the car from the available car list if it has an alert/error.
- Employees can only registered to the system by the administrator authority

1.4.GLOSSARY

Customer: A person that registered and has an account properties with the following information:

- Name,
- Surname
- Mobile Number
- Credit card number
- Drive license
- Password

Customer register to the system with name, surname, mobile number, credit card number and drive license then an access to the system is provided with a password that by sending an SMS to the mobile number used in registration.

The customer has only capabilities that are mentioned under the "Goals" title.

Employee: A person that registered to the system by the administrator and responsible to maintain the system flow in success by helping customers, taking care of cars.

Administrator: The administrator of the system is the person has a full access to entire data of the system and an authority to manage the system.

Guest: A possible customer who hasn't logged in or registered yet. Guests can access system functionality after making a registration and logging in.

Reservation: Can be made for a single car in a period of 1 hour. After customer reserves a car, the reservation is created immediately and expires after 1 hour.

Missed reservation: If the customer doesn't pick up the car that he/she reserved in one hour, customer is charged 1 EUR and the car is tagged as available by the system.

Canceled reservation: If the customer wants to cancel the reservation and he/she can be able to make a cancellation request in 30 minutes after the reservation is made. Customer is charged for 0.50 EUR if he/she cancels the reservation.

Charging: Charging is made with a given amount of money per minute. Charging starts as soon as the engine ignites and ends as soon as the engine stops and car is parked in a safe area and driver exits the car.

Payment: Payment is provided from the credit card number defined in the account after a certain amount of time passed from the time charging stopped.

Power grid station: Station that are equipped with certain amount of power grids for charging the cars and it is under control/management of the company and owned by the company.

GPS: The Global Positioning System is a satellite-based navigation system that transmits radio signals which is used for providing geolocation and time information. The GPS has a specific role in the system to obtain coordinates of the cars and customers.

Safe Area: Is an area where parking is only allowed in. It is shown in the map by default and contains the electrical cars.

Trip: It starts when the customer enters the cars and starts the engine.

Shared Trip: It is possible to have other people in the car, to have a discount customer should have at least two other passengers. Shared trip is automatically detected with the sensor system.

Shared Trip Discount: If the system detects the customer took at least two passengers into the car, the system applies a discount of 10% on the last trip.

Battery Friendly Discount: If a car is left with no more than 50% of the battery empty, the system applies a discount of 20% on the last trip.

Increment in charge: If a car is left more than 80% of the battery empty or 3 km from the nearest power grid station then the system charges 30% more on the last trip to compensate for the cost required to re-charge.

Push notification: Notifications that are sent to a smartphone that are specified with an application.

SMS: Short Message Service is a notification in which sent to a mobile phone by SMS gateway.

API: Application Programming Interface is used for interacting and communicating with other existing systems.

1.5.ASSUMPTIONS

Clarifications of the specification document are made with the following assumed facts.

- Employees are already exist and registered in the system and have determined jobs to maintain the system.
- Mobile numbers are/must be unique in the database which is checked within the registration in order to prevent a user making more than one reservation in an hour with more than one account.
- Drive licenses are/must be unique.
- Customer use emergency call only if there is a kind of emergency.
- The cars are taking cared by employees and customers.
- Customer that enters the car will start the engine in an amount of time otherwise car sends an emergency notification and sound the alarm to warn that there is an abusement of usage or an urgency case.
- Cars are already equipped with an passenger detection system in order to apply discounts.
- There is no possibility to have an active reservation more than one.

- Cars are already equipped with an IOT based own system to contact with the company employees to inform there is an error and/or car accident and/or lack of the battery.
- The car system is able to detect if the user parked in safe area and exited the car, if so it locks the car automatically and send acknowledgement to the system to make car available again and to stop charging customer.
- Customers park only in the safe area.
- Account and access that provided to the employee is deleted if the employee leaves the company.
- Cars only parked in public areas and defined stations with respecting traffic laws after usages.
- Employees handle the emergency calls, fix the errors, help the customers on time.
- The system interacts and gets car information from the car system.
- The administrator has a full access to entire data of the system, can manage all type of accounts, has an authority to make changes and distribute permissions in the system.

1.6.CONSTRAINS

The system require to customer the permission to get his position and he has to manage sensible data (name, surname, drive licence number, position, phone number, credit card number) respecting the privacy law.

To allow proper operation of the application the customer must possess a minimum 3G data connection, the GPS and the physical space for the installation of the application.

1.7.PROPOSED SYSTEM

We are going to develop a car sharing application depending on the mobile using Java EE platform with a database. The system is going to be comprised of a server which allows of using of mobile application. Additionally, there will be users kind of administrator, employees, and customers who

are going to interact with the server using mobile application and GUI. The system is going to use the technologies of GPS and SMS in mobile phone.

1.8.IDENTIFYING STAKEHOLDERS

The possible stakeholders of that system are:

- A company already deployed in the market that wants to make a technological breakthrough;
- Start-up company that wants to enter the industry to take a market share;
- A company that wants to differentiate its concrentrandosi growth in a new market with a new product

2. ACTORS IDENTIFYING

The system composed of 4 users.

- Guest
- Customer
- Employee
- Administrator

3. REQUIREMENTS

3.1.FUNCTIONAL REQUIREMENTS

Regarding to the domain properties that hold, in order to satisfy the determined goals, following requirements are derived. The requirements are grouped regarding to goal groups.

Customers:

- Registration to the system
    - The system must allow user to register with name, surname, mobile number, credit card number and drive license.
    - System must check the validation of the credit card under some protocols

- The system must not accept non valid credit cards and must not allow registration in that case.
- In order to complete registration the system must sends a password to the given mobile number via SMS.

- Login to the system
  - The system must allow customer to login into the system with credentials consisting of mobile number and password.
  - The system must load map access page with correct credentials.
  - System also must present a “forget password” option who forgot their password so with this option they can retake their password via SMS by giving their mobile number that registered with.
- Log out from the system
  - The system must allow customer to log out from the system.
- Allowed to manage profile information
  - The system must present an profile managing with changeable information of credit card, which will require a new validation in that case, and password.
- Allowed to reserve an available single car, for up to one hour before they pick it up.
  - The system must show all the available cars in certain geographical region.
  - The system must show the power grid stations in the map.
  - The system must allow customer to reserve only a single available car.
  - The system must start a timer after reservation is created.
  - The system must expire the car reservation and charge customer 1 EUR when 1 hour is passed and the car is not taken.
  - The system must remove the reserved car from the available cars.
- Allows to make a cancellation of the reservation

- The system must allow customer to cancel their reservation. If cancellation is made up to 30 mins from the creation of reservation customer charged 0.50 EUR, if it is made after 30 mins customer is charged 1 EUR.
- The system must change the status to available of the car that it's reservation canceled.
- The system must maintain the timer.

- Allowed to know the current charges through a screen on the car
  - The system must start charging customer as soon as the engine is started.
  - The system must have calculation functionality that calculates the charging for a given amount of money during the trip.
- Allowed finding the locations of available cars within a certain distance from the current location of the customer, at the current time and/or from a specified location, at a determined time by the customer.
  - The system must able to determine the location coordinates of the customer via GPS.
  - The system must able to determine the current time zone of the user belong via GPS.
  - The system must have a functionality of using custom location and time as inputs.
  - The system must able to calculate an area with a certain distance radius from the input location.
- Allowed to list the properties (battery level, exact address, estimated distance can be made) of the available cars
  - The system must list the properties of cars that can be obtained.
- Allowed to see the safe area in the map that the car can be parked which is predefined by the management system
  - The system must show the safe area in the map.

    - The system must have a functionality of transmission from customer to the employee.

- Allowed to tell the system that customer is nearby so the system unlocks the car in order to enter
    - The system must have functionality of detecting if customer is nearby by matching car's and customer's GPS.
    - The system must able to interact with the car's system if the customer is nearby in order to car system open the doors.
- Allowed to share the drive with other people to get a discount with the sensors of the car.
    - The system must able to calculate the discount by interacting and getting the discount type that detected from the car system.
- Allowed to see special parking areas to recharge the car by taking care of plugging the car into the power grid in order to get a discount.
    - The system must have a functionality to show the special parking areas in map.
- Offers to enable the many saving option, customer input final destination and the system provide the station information where to leave the car to get a discount.
    - The system must have a functionality of getting a location as an final destination input.
    - The system must show the station information by the functionality of calculation the distribution of cars in the city based on both the destination of the customer and availability of the plugs at the selected station.

<u>Guest:</u>

- Allowed to make a registration
    - The system must have a functionality to show registration form and proceed the registration process after register requested.

3.2.NON-FUNCTIONAL REQUIREMENTS

Availability

- **Description 1:** A PowerEnJoy system's availability, or "uptime," is the amount of time that it is operational and available for use.
- **Fit Criteria 1:** This will be checked twice a month and the PowerEnJoy system will not be available during this period.
- **Rationale 1:** This is specified because some systems are designed with expected downtime for activities like database upgrades and backups.

Maintainability

- **Description 1:** The ability to change the system to deal with new technology or to fix defects.
- **Fit Criteria 1:** The PowerEnJoy system will be checked every hour to pretend any interruption in the process of repairing.
- **Rationale 1:** This is significant because this kind of systems should be open to new technologies and system defects should be pretended because of users should not be aggrieved.

Performance

- **Description 1:** Response times:
  - Loading of Main Page is not more than 2 seconds.
  - Searching of car is not more than 2 seconds.
  - Reservation of car is not more than 1 seconds.
  - Cancelling of reservation is not more than 1 seconds.
- **Fit Criteria 1:** System performance is going to be tested every week, because of being certain about there is no any delays.
- **Rationale 1:** This is essential because this kind of failures will dissatisfy the users and these loading times should not longer than it should.

Backup and Restore

- **Description 1:** When the PowerEnjoy system falls down, the purpose is to recover data after its loss.

- **Fit Criteria 1:** Data will be stored each month for using on back-up and restore.
- **Rationale 1:** The purpose of this storing process provides the latest data when the system is restored again.

Security

- **Description 1:** The PowerEnJoy system will allow for a restricted level of access:
    - Administrator: Full access to PowerEnJoy system.
    - Employee: Access to all sub-systems of PowerEnJoy except Administrator.
    - Customer: Access to all sub-systems of PowerEnJoy except Administrator and Employee.
    - Guest: Access to registration.
- **Fit Criteria 1:** This situation will be checked every day to provide restricted level of access.
- **Rationale 1:** This is important for each users to use their access right.

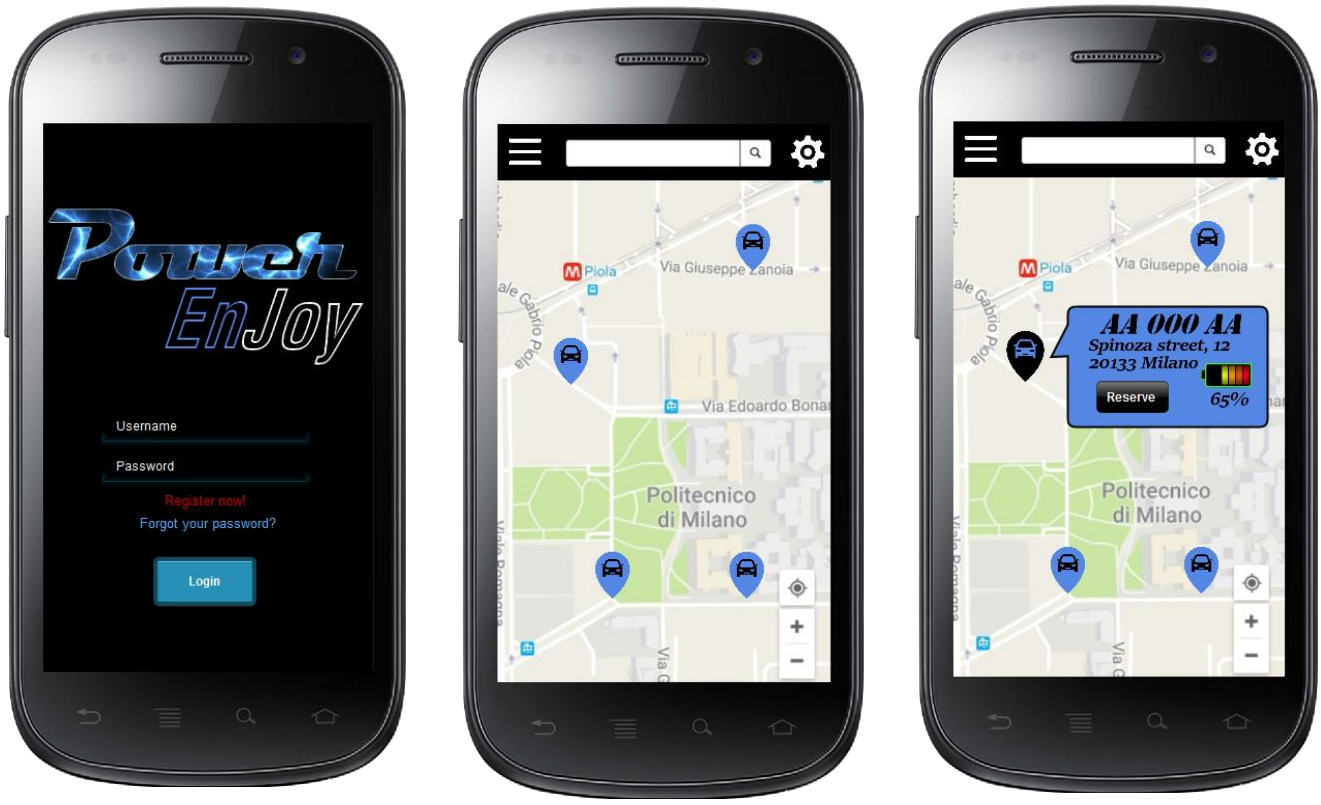
Capacity

- **Description 1:** At the beginning, 100 synchronical users will get access to network.
- **Fit Criteria 1:** The system performance will be checked every day, because of user satisfaction.
- **Rationale 1:** This is essential for this system, in that when all users are logged to system at the same time, the system should not slow down.
- **Description 2:** The system should be stored 10 GB data a day.
- **Fit Criteria 2:** The system will be checked every day to ensure that it can overcome over 10 GB of data stored.
- **Rationale 2:** This is useful to see how much data flows within the system.

3.2.1. USER INTERFACE

The system is designed as a web app. For this reason, if implemented in a scalable way, it can be reached either via pc and smartphone.

Considering that the primary use is a smartphone, a mockup of the system will be presented below



The homepage is designed the simplest possible. It is necessary log in to access the system functionality. There is the possibility to register or recover lost credentials.

Once logged in, the map will be shown and the cars currently available arranged, as colorful pins, on it. Clicking on the pins of each car, car information will be displayed and the car can be reserved.

Among the information it seemed essential to insert the car battery power left.

3.2.2. DOCUMENTATION

The following documents will be released in order to plan the implementation of the system

- RASD, Requirement Analysis and Specification Document, which defines our goals and assumptions and contains an overall description of the system (using scenarios and use-cases) and the models describing requirements and specifications.
- DD, Design Document, which contains a functional description of the system using models such as UML diagrams.

- ITPD, Integration Test Plan Document, which contains the plan and the results of the testing activity performed on a system.
- PP, Project plan, which is the system development plan, divided for each group member.

4. SCENARIO IDENTIFYING

The following paragraphs will describe the system usage scenarios.

The actors are Jessica, a young girl who just moved in town and does not own a car, and Mark, who has just been hired by the company, with the aim of dealing with the management of the cars of the business.

Jessica has been invited by some friends to spend the evening together. Being aware of the fact she might come back home late and considered that she had seen the advertisement of the service offered by the system, she decided to register.

After having filled out the form, having verified the authenticity via phone and having received a SMS regarding her password, her account is now ready and she can use it immediately.

At the end of the evening, Jessica takes her smartphone and, after logging in into the system, she finally reserves a car that is 350m far from her and it is charged for 70% of the full battery. Despite this, her friends convince her to remain a little longer with them, so she decides to cancel the reservation.

It's finally time to go back home, Jessica approaches the car she reserved in advance (not the one she cancelled, but another one) and, with her smartphone, she alerts the system to unlock the doors. Once entered into the car, she begins her way home and the onboard computer keeps her updated on the situation of the car (i.e. the battery charge level, if there is any damage) and how much money she is spending.

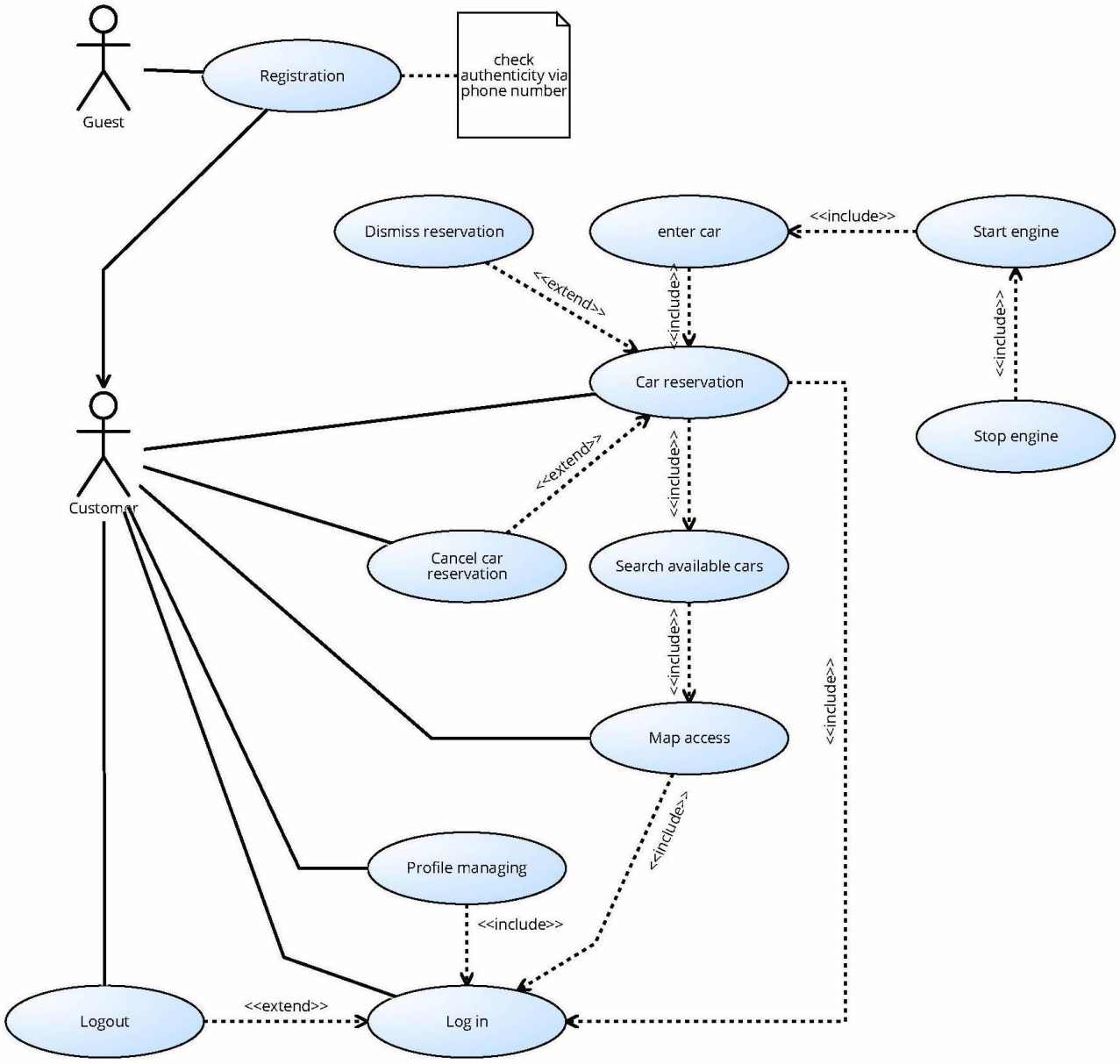
Once back to her final destination, she stops engine and the doors lock as soon as she gets out the car.

5. UML MODELS

5.1.USE CASE DIAGRAM

Some use cases can be derive from the scenarios identified in the previous paragraph:

- Registration
- Login
- Logout
- Map Access
- Car Reservation
- Cancel Car Reservation
- Profile Managing



5.2.USE CASE DESCRIPTION

5.2.1. Registration

Name	Registration
Actors	Guest
Entry Conditions	The guest isn't registered to the system
Flow of events	• The guest enters the application (web or phone app); • The guest clicks on the "Register now" button; • The guest fills in the form where he has to write: → Name → Surname → Phone number → Drive licence number → Credit card number And optionally: → Address → Picture • The guest clicks the "register" button; • The system shows him home page.
Exit conditions	The system shows the home page.
Exceptions	The information inserted by the user are wrong/not consistent with the fields. The System shows an error

	message to the user.

5.2.2. Login

Name	Log in
Actors	Customer
Entry Conditions	Customer has successfully signed up to the system
Flow of events	• The customer enters the web site. • The customer fills in the text fields in the home page with username and password. • The customer clicks on the “Login” button.
Exit conditions	The system shows the map page.
Exceptions	The password and/or username inserted by the customer are wrong. The System shows an error message to the customer .

5.2.3. Logout

Name	Log out
Actors	Customer
Entry Conditions	Customer has successfully log in to the system
Flow of events	• The customer is logged to the system. • The customer clicks on the “Logout” button.

Exit conditions	The system shows the home page.
Exceptions	If something goes wrong during the “Logout” the system shows an error message to the customer.

5.2.4. Map Access

Name	Map access
Actors	Customer
Entry Conditions	Customer has successfully log in to the system
Flow of events	The user have access to the map and he can: ● see the available cars ● see safe area ● see parking areas
Exit conditions	
Exceptions	

5.2.5. Car Reservation

Name	Car Reservation
Actors	Customer
Entry Conditions	Customer has successfully log in to the system and the system show the map page
Flow of events	● The customer search a car ● The customer select the car chosen

	• The customer press the “Reserve” button • The system show a countdown (time customer have to arrive to the car or cancel the reservation)
Exit conditions	the system show a countdown
Exceptions	The car is no more available when the customer press the “reserve” button. The system show an error message to the customer

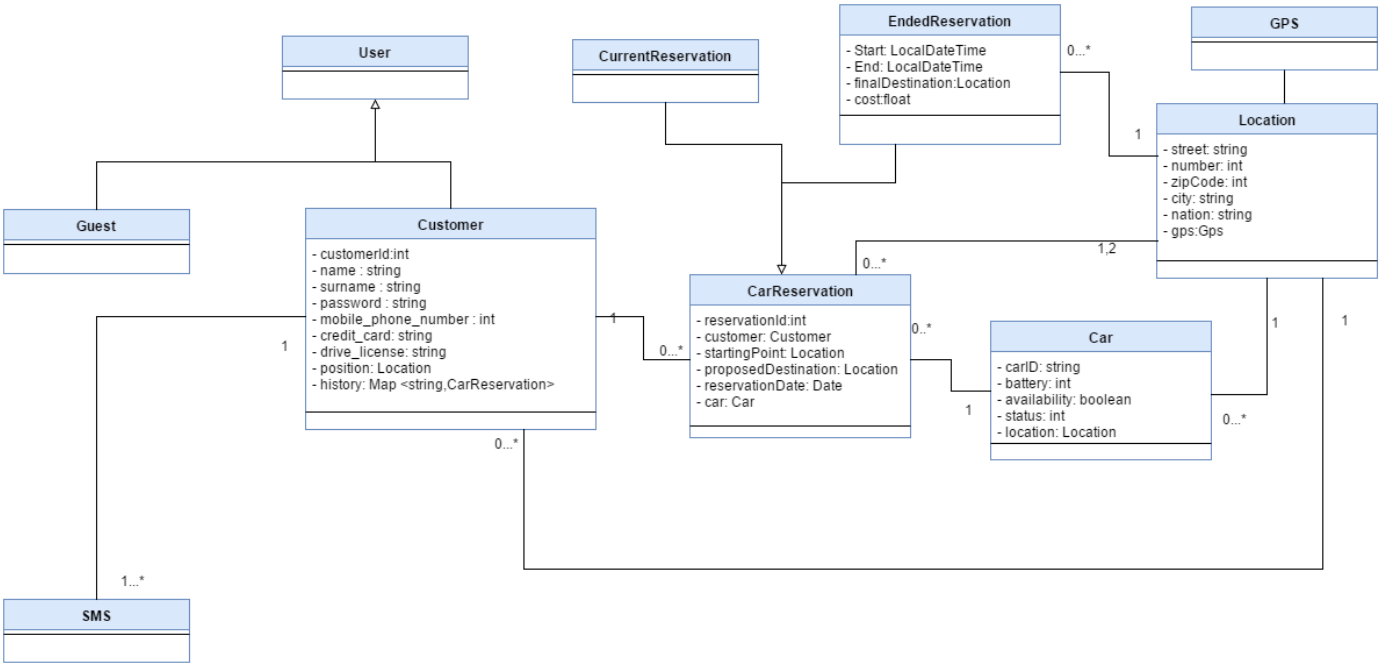
5.2.6. Cancel Car Reservation

Name	Cancel Car Reservation
Actors	Customer
Entry Conditions	Customer reserved a car and the countdown is started
Flow of events	• The system show a customer a countdown • The customer press “cancel” button • The system charge the customer a fee of 0.50 EUR for cancellations made up to 30 mins. • The system charge the customer a fee of 1 EUR for cancellations made after 30 mins.
Exit conditions	The system show the map page
Exceptions	If something goes wrong during the cancellation the system shows an error message to the customer.

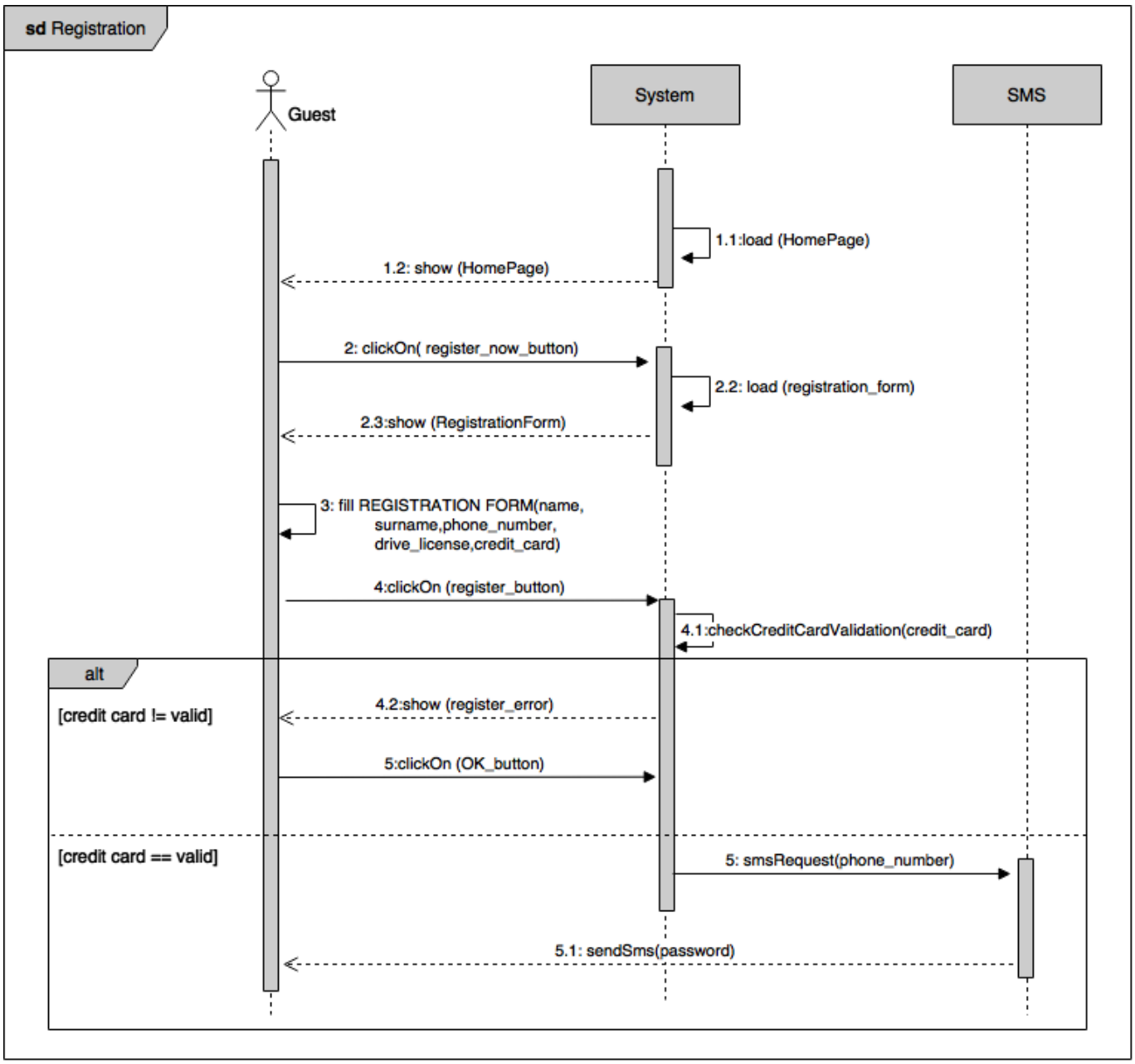
5.2.7. Profile Managing

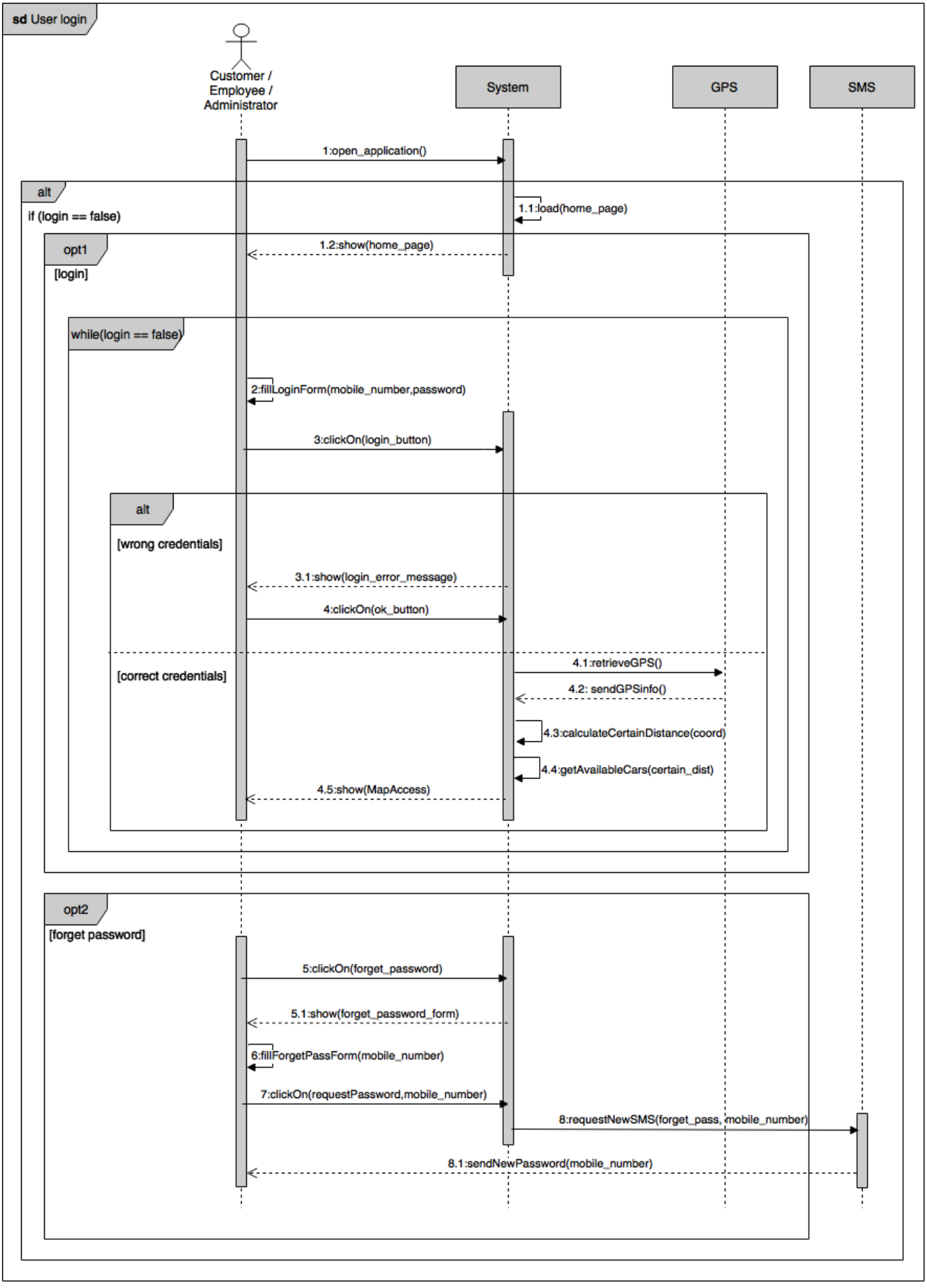
Name	Profile Managing
Actors	Customer
Entry Conditions	Customer has successfully log in to the system
Flow of events	• The customer is logged to the system. • The customer press the "profile" button • The customer can change credit card information and password • the customer press "save" button
Exit conditions	The system show the map page
Exceptions	If something goes wrong during the modification the system shows an error message to the customer.

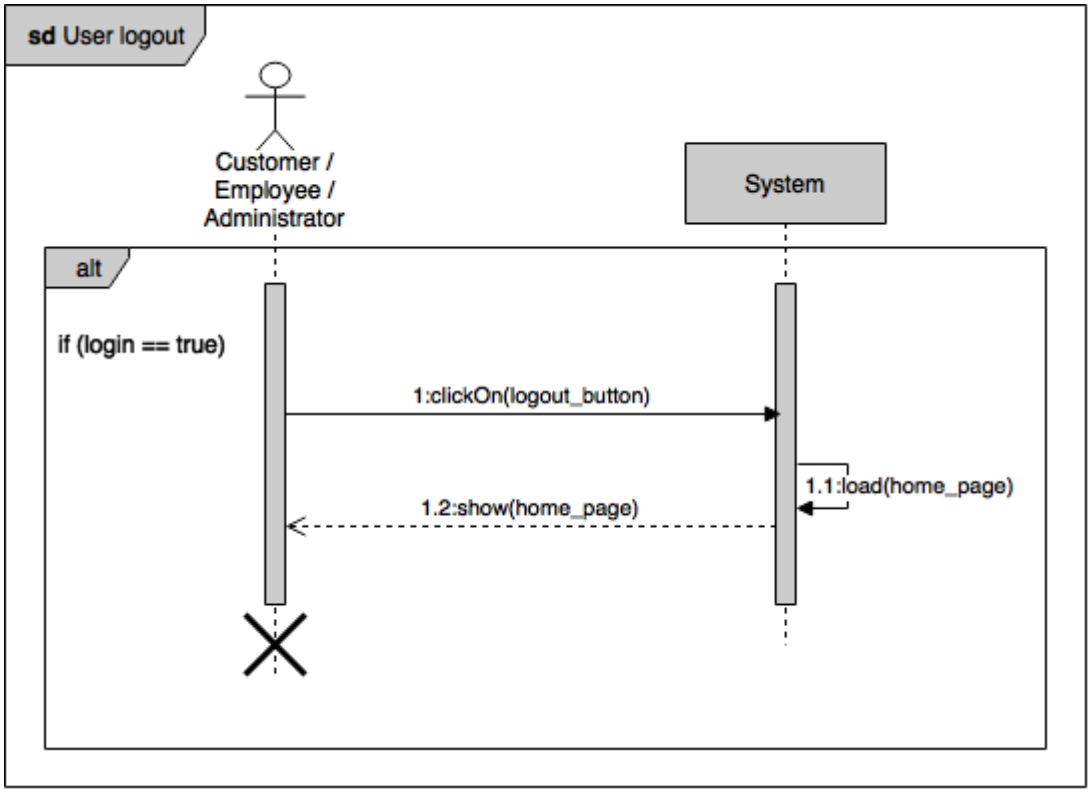
5.3.CLASS DIAGRAM

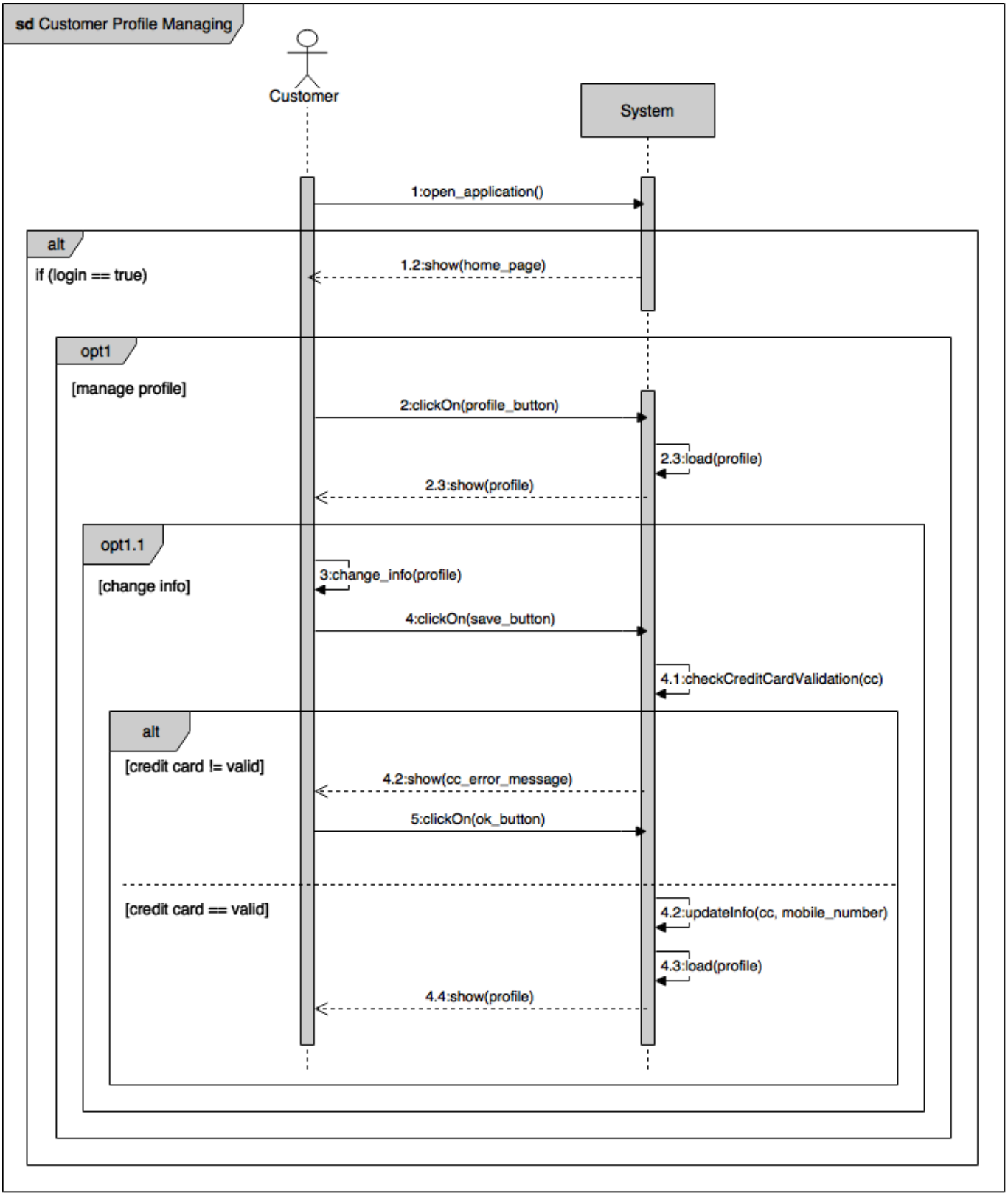


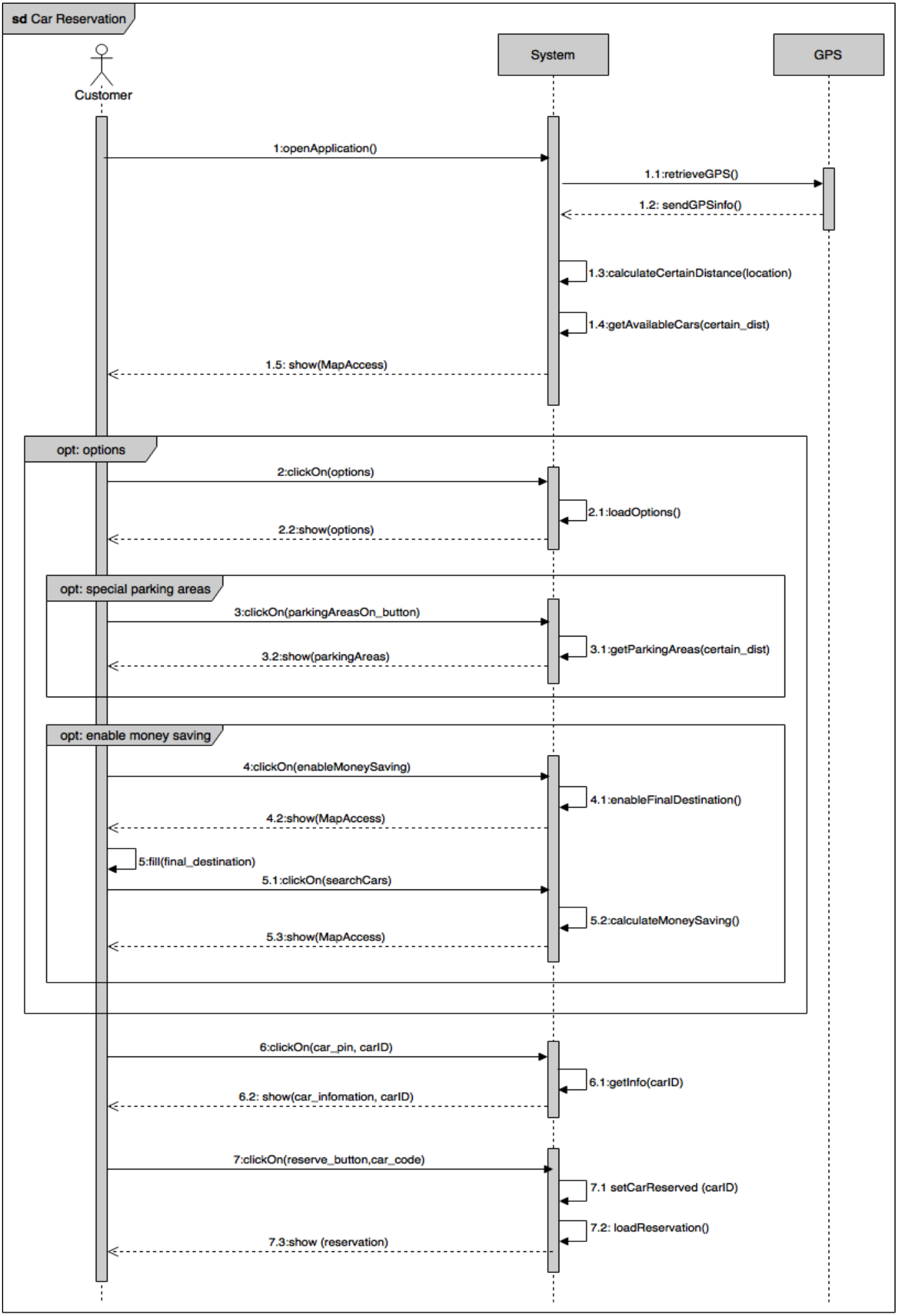
5.4.SEQUENCE DIAGRAMS

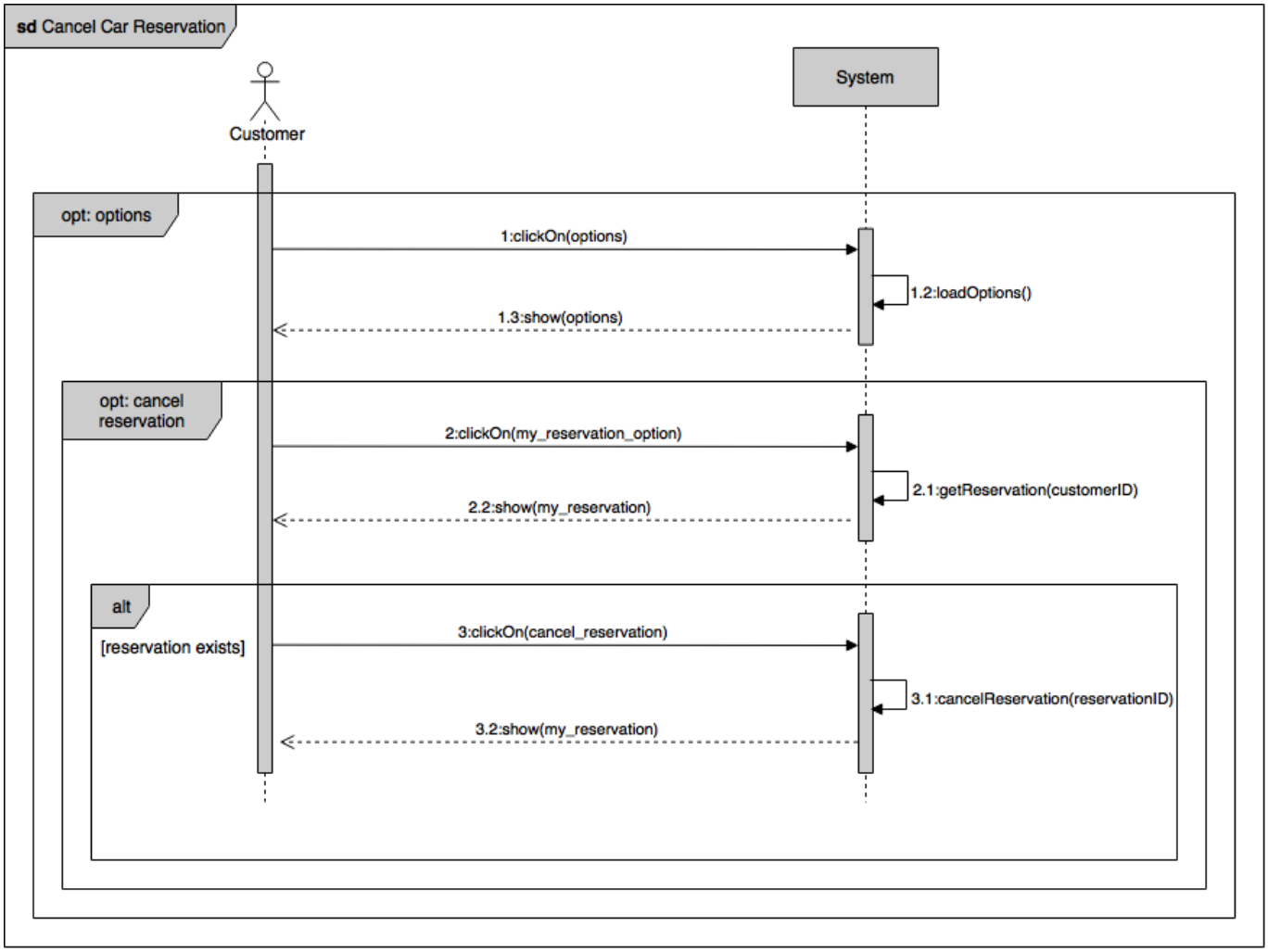



sd User login
Customer / Employee / Administrator
System
GPS
SMS
1:open_application()
alt
if (login == false)
1.1:load(home_page)
opt1
[login]
1.2:show(home_page)
while(login == false)
2:fillLoginForm(mobile_number,password)
3:clickOn(login_button)
alt
[wrong credentials]
3.1:show(login_error_message)
4:clickOn(ok_button)
[correct credentials]
4.1:retrieveGPS()
4.2: sendGPSinfo()
4.3:calculateCertainDistance(coord)
4.4:getAvailableCars(certain_dist)
4.5:show(MapAccess)
opt2
[forget password]
5:clickOn(forget_password)
5.1:show(forget_password_form)
6:fillForgetPassForm(mobile_number)
7:clickOn(requestPassword,mobile_number)
8:requestNewSMS(forget_pass, mobile_number)
8.1:sendNewPassword(mobile_number)

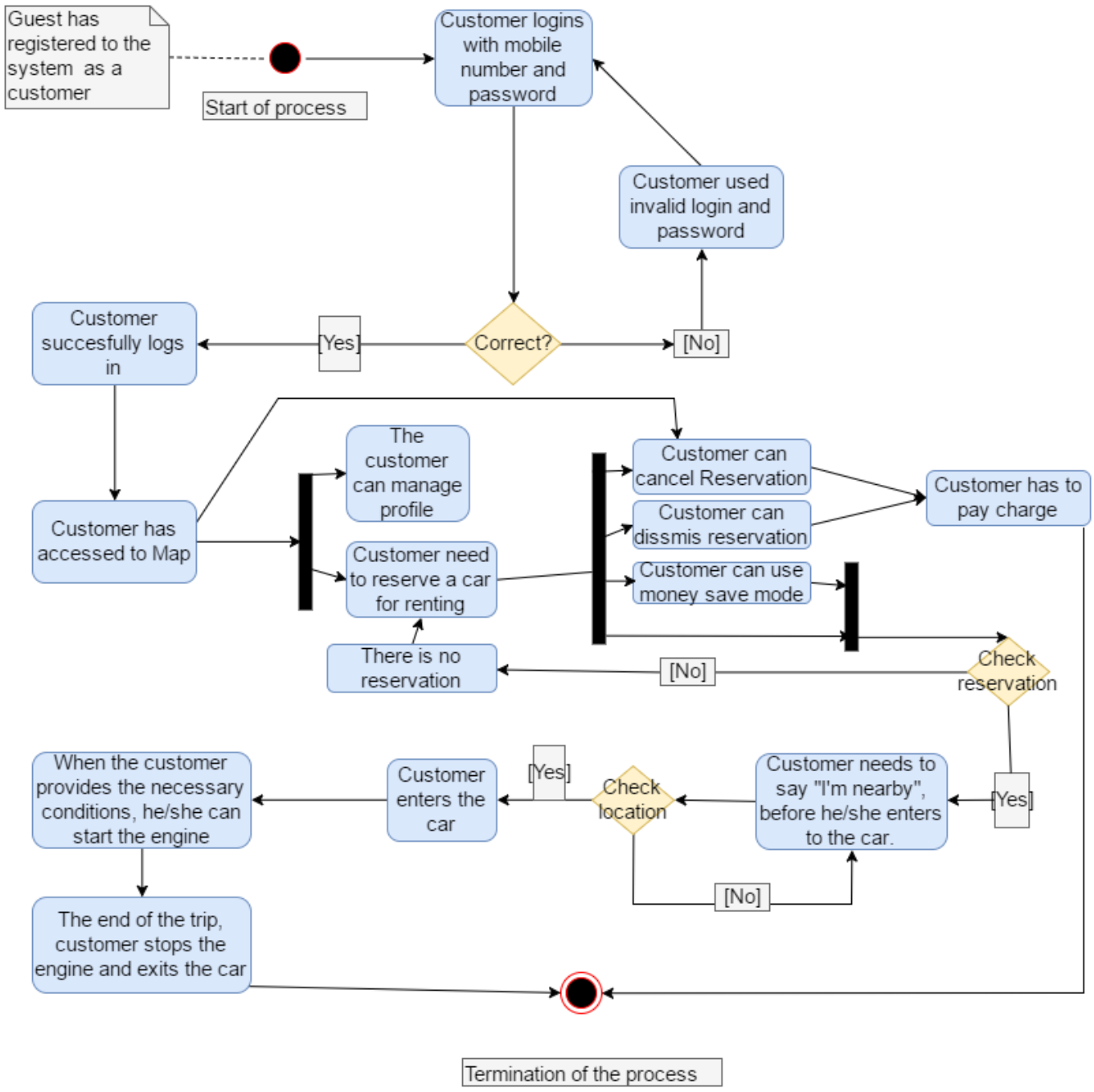

sd User logout
Customer /
Employee /
Administrator
System
alt
if (login == true)
1:clickOn(logout_button)
1.1:load(home_page)
1.2:show(home_page)


sd Customer Profile Managing
Customer
System
1:open_application()
alt
if (login == true)
1.2:show(home_page)
opt1
[manage profile]
2:clickOn(profile_button)
2.3:load(profile)
2.3:show(profile)
opt1.1
[change info]
3:change_info(profile)
4:clickOn(save_button)
4.1:checkCreditCardValidation(cc)
alt
[credit card != valid]
4.2:show(cc_error_message)
5:clickOn(ok_button)
[credit card == valid]
4.2:updateInfo(cc, mobile_number)
4.3:load(profile)
4.4:show(profile)


sd Car Reservation
Customer
System
GPS
1:openApplication()
1.1:retrieveGPS()
1.2: sendGPSinfo()
1.3:calculateCertainDistance(location)
1.4:getAvailableCars(certain_dist)
1.5: show(MapAccess)
opt: options
2:clickOn(options)
2.1:loadOptions()
2.2:show(options)
opt: special parking areas
3:clickOn(parkingAreasOn_button)
3.1:getParkingAreas(certain_dist)
3.2:show(parkingAreas)
opt: enable money saving
4:clickOn(enableMoneySaving)
4.1:enableFinalDestination()
4.2:show(MapAccess)
5:fill(final_destination)
5.1:clickOn(searchCars)
5.2:calculateMoneySaving()
5.3:show(MapAccess)
6:clickOn(car_pin, carID)
6.1:getInfo(carID)
6.2: show(car_infomation, carID)
7:clickOn(reserve_button,car_code)
7.1 setCarReserved (carID)
7.2: loadReservation()
7.3:show (reservation)


sd Cancel Car Reservation
Customer
System
opt: options
1:clickOn(options)
1.2:loadOptions()
1.3:show(options)
opt: cancel reservation
2:clickOn(my_reservation_option)
2.1:getReservation(customerID)
2.2:show(my_reservation)
alt
[reservation exists]
3:clickOn(cancel_reservation)
3.1:cancelReservation(reservationID)
3.2:show(my_reservation)

5.5.ACTIVITY DIAGRAM



6. ALLOY MODELING

6.1.ALLOY CODE / MODEL

```
sig Integer{}
sig Strings {}
sig Float{
    decimal:int[1],
    underDecimal:int[1]
}
sig Boolean{
    status:int[1]
}
sig Date{}
sig LocalDateTime{}
sig Location{
    street: one Strings,
    number:one Integer,
    zipCode:one Integer,
    city:one Strings,
    nation: one Strings,
    gps: one Gps
}
sig BatteryConstant{
    battery: int[1]
}
abstract sig User {}
sig Guest extends User{}
sig Customer extends User{
    customerId: one Integer,
    name:one Strings,
    surname:one Strings,
    password:one Strings,
    phone:one Integer,
    creditCard:one Strings,
    driveLicense:one Strings,
    history: set CarReservation,
    position: lone Location
}
```

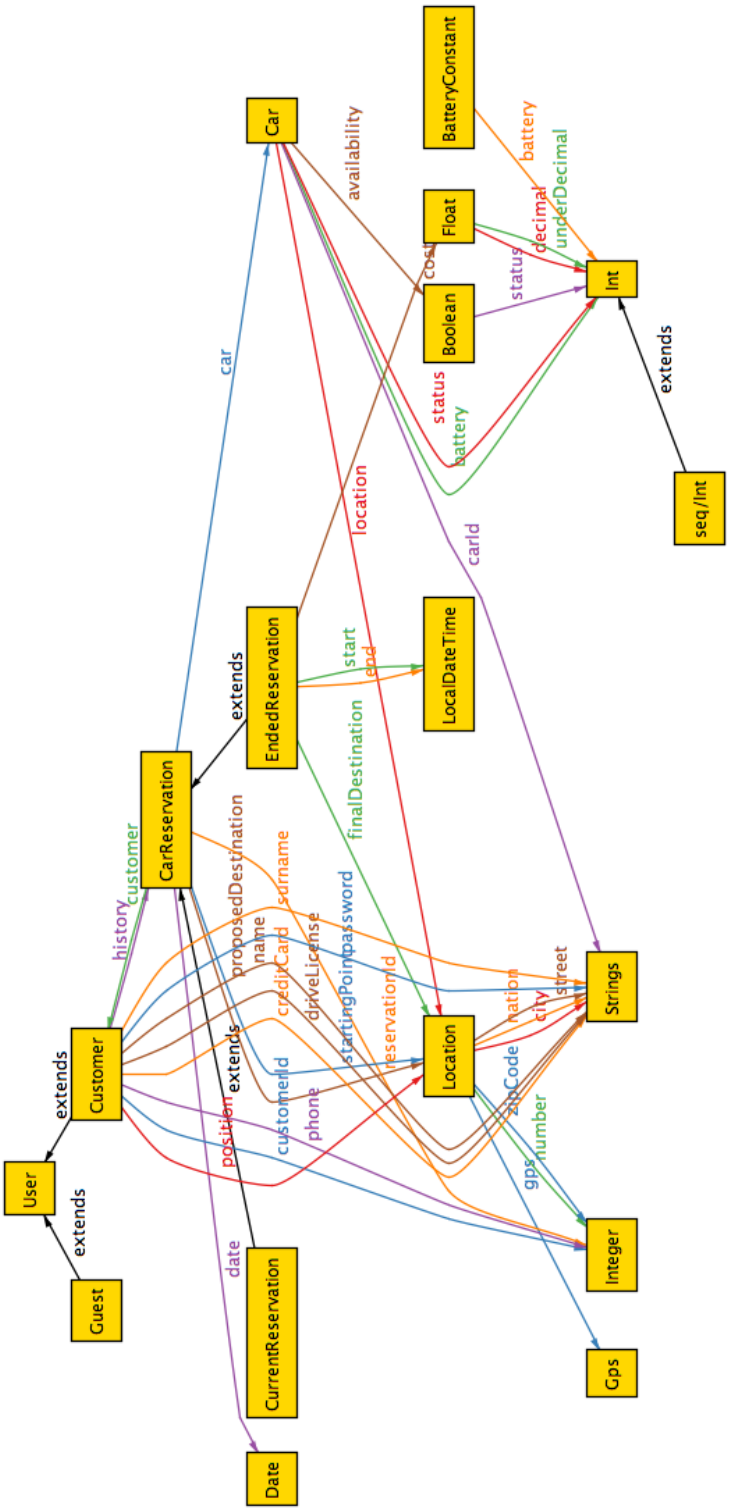
```
abstract sig CarReservation{
    reservationId: one Integer,
    customer: one Customer,
    car:one Car,
    startingPoint:one Location,
    proposedDestination:lone Location,
    date: one Date
}
sig CurrentReservation extends CarReservation{}
sig EndedReservation extends CarReservation{
    start:one LocalDateTime,
    end:one LocalDateTime,
    finalDestination:one Location,
    cost: one Float
}
sig Gps{}
sig Car{
    carId: one Strings,
    battery: int[1],
    availability: one Boolean,
    status: int[1],
    location: one Location
}
```

```
// ---Facts---//

//Boolean class//
fact booleanProperties{
   all b:Boolean| (b.status=0)or(b.status=1)
}
//No empty Costumer important fields//
fact noEmptyCustomer{
   all c:Customer |(#c.customerId=1)and (#c.creditCard=1) and (#c.name=1) and (#c.surname=1) and (#c.phone=1)and
    (#c.password=1) and (#c.driveLicense=1)
}
//No empty Location//
fact noEmptyLocation{
   all l:Location |(#l.street=1) and (#l.number=1)and (#l.zipCode=1) and (#l.city=1) and (#l.nation=1)and(#l.gps=1)
}
//No empty CarReservation//
fact noEmptyCarReservation{
   all c:CarReservation | (#c.reservationId=1)and(#c.car=1)and(#c.customer=1)and(#c.startingPoint=1)
   and(#c.proposedDestination=1)and(#c.date=1)
}

//No empty EndedReservation//
fact noEmptyEndedReservation{
   all c:EndedReservation| (#c.reservationId=1)and(#c.car=1)and(#c.customer=1)and(#c.startingPoint=1)
   and(#c.proposedDestination=1)and(#c.date=1)and(#c.start=1)and(#c.end=1)and(#c.cost=1)and(#c.finalDestination=1)
}

//No empty Car//
fact noEmptyCar{
   all c:Car|(#c.carId=1)and(#c.battery=1)and(#c.availability=1)and(#c.status=1)and(#c.location=1)
}

//No duplicate Location//
fact noDuplicateLocation{
   no disj l1,l2:Location|(l1.gps=l2.gps)
}

//No duplicate Customer//
fact noDuplicateCustomer{
   no disj c1,c2:Customer|(c1.customerId=c2.customerId)or(c1.driveLicense=c2.driveLicense)
   or(c1.phone=c2.phone)
}
```

```
//No duplicate Car//
fact noDuplicateCar{
   no disj c1,c2:Car|(c1.carId=c2.carId)
}

//No duplicate CarReservation//
fact noDuplicateCarReservation{
   no disj c1,c2:CarReservation|(c1.reservationId=c2.reservationId)
}

//Cost domain//
fact costDomain{
   all c:EndedReservation| c.cost.decimal>=0 or c.cost.underDecimal>=0
}

//no multiple reservation//
fact noMultipleReservation{
   no disj r1,r2:CurrentReservation | r1.customer=r2.customer
}
fact differentTime{
   all c:EndedReservation| disj[c.start,c.end]
}



//---Predicates---//

pred addReservation(r:CarReservation, c1,c2:Customer){
    r not in c1.history  implies c2.history=c1.history+r
}
run addReservation for 5

pred cancelReservation(r:CarReservation, c1,c2:Customer){
       r in c1.history  implies c2.history=c1.history-r
}
run cancelReservation for 5

pred show(){}
run show for 5


//---Asserts---//

assert addReser{
   all r:CarReservation, c1,c2:Customer|(r not in c1.history)  and addReservation[r,c1,c2] implies( r in c2.history)
}
check addReser for 5

assert carCustomer{
   no r:CarReservation |(no c:Customer|r.customer=c)
}
check carCustomer for 5

assert cancelReser{
   all r:CarReservation, c1,c2:Customer|(r in c1.history)  and cancelReservation[r,c1,c2] implies( r not in c2.history)
}
check cancelReser for 5
```

```
6 commands were executed. The results are:
  #1: No counterexample found. addReser may be valid.
  #2: No counterexample found. carCustomer may be valid.
  #3: No counterexample found. cancelReser may be valid.
  #4: Instance found. addReservation is consistent.
  #5: Instance found. cancelReservation is consistent.
  #6: Instance found. show is consistent.
```

6.2.ALLOY WORLDS



7. FUTURE DEVELOPMENT

- Employees should receive alerts and emergencies as push notifications via GPS from the area that defined for them.
- Employees should able to manage and handle alerts and send acknowledgement and feedbacks
- Employees should allowed to enter any car by requesting access to car in which system detects the car via GPS.
- Customers should able to make an emergency call

8. USED TOOLS

Microsoft Office Word 2016: to redact and to format this document

Google Drive: to store, share and manage project files

Google Docs: to modify project files

Alloy Analyzer 4.2: to prove the consistency of the model

Drawio: to create UML Models

Github: to keep documents and versions

9. HOURS OF WORK

Name & Surname	17.10.2016	25.10.2016	04.11.2016	13.11.2016	Outside of Group	Total
Burcu CESUR	6 hour	5 hour	8 hour	4 hour	15 hour	38 hour
Marilena COLUCCIA	6 hour	5 hour	8 hour	4 hour	16 hour	39 hour

Mustafa ÇEÇE	6 hour	5 hour	8 hour	4 hour	12 hour	35 hour