

TABLE OF CONTENT

TABLE OF CONTENT	2
INTRODUCTION	4
Description of the given problem	4
Actual system	4
Goals	4
Domain properties.....	5
Glossary	5
Assumptions	6
Constraints.....	6
Proposed System	7
Identifying Stakeholders.....	7
ACTORS IDENTIFYING	7
REQUIREMENTS.....	7
Functional requirements	7
Requirement 1	7
Requirement 2	8
Requirement 3	8
Requirement 4	8
Requirement 5	8
Requirement 6	9
Requirement 7	9
Requirement 8	9
Non-functional requirements.....	9
SCENARIO IDENTIFYING.....	9
Scenario 1	10
Scenario 2	10
Scenario 3	10
Scenario 4	10
Scenario 5	10

UML MODELS.....	12
Use case diagrams	12
Insert registration credential.....	12
Insert payment method.....	14
Insert login credentials	14
See own account information	14
Find available car	15
Book car	15
Unlock car	16
Pay for a car	17
Contact assistance during trip	17
Activate money saving option	17
Class diagram.....	19
Sequence diagrams.....	20
Activity diagrams	21
State diagrams.....	21
ALLOY MODELING.....	22
Model.....	22
Alloy result	22
World generated	22
FUTURE DEVELOPMENTS	22
USED TOOLS.....	22
HOURS OF WORK.....	22

INTRODUCTION

Description of the given problem

According to the assignment document the problem consists in the definition and implementation of a system that manages the activity of an electric car sharing. The activity itself doesn't need to be explained because car sharing is a largely spread out costume.

More in depth, the system have to allow people to join a community of car sharers by registering into the system and after this, they can share cars with neighbors and friends in order to save money together and reduce pollution made by public means of transport and private cars.

Actual system

Actually the car sharing works without a computer system and there are lots of employees dislocated in the safe areas that provide car keys when users want to book cars.

The only system available in the company is a system that manages all the bureaucracy such as car revisions, car maintenance, fine delivery and car recovery.

This system stores all the data about the cars in a MySQL database and tracks continuously all the cars to provide assistance by telephone in case of failure or user request.

Goals

In this section of the document, we are presenting the main goals that the software-to-develop must fulfill in order to satisfy the requirements:

- Users must have the possibility to register to the system to become part of the community. They must provide their credentials (such as first name, last name email address and so on) and a valid method of payment that can be an IBAN code or a credit card code. After the online registration, users will receive a confirmation email that includes a password for the access to the system and a link that will conclude the registration task and will show the user his personal area from where he/she can decide to do stuff.
- The user can select the parking lot from a subset of them where to pick the car. The user's position or the input of an address will be the base on which the system will show a subset of parking lots: the system will choose the parking lots with at least one available car in a range of 1 km from the given position.
- The user will be able to reserve a car for up to one hour from the pickup, from the list of the available ones in the selected parking lot. Then the system will lock automatically the selected car to everyone until the reservation time, with except to the user that will use his mobile device to identify himself and let the system know that he's picking up the car.
- The system must control that every reserved car will be picked in the time range: if this condition is not satisfied, it will assign an extra fee of 1€ to the owner of the unpicked car and then it will set the car as available in order to allow other people to pick that car.

- The system must charge the user by a certain amount of Euros per minute. The time outside the predefined areas will contribute to calculate the final amount for the user to pay and a built-in-car screen will show this information to the user/driver. This amount can be decreased or increased by:
 - The number of passengers of the car (a sensor will count the passengers in order to apply a 10% discount on the final amount);
 - The battery charge remaining (if the battery has more than 50% of the charge, the system must apply a 20% discount);
 - If the car is parked in a charging station and the user plugs the car BEFORE he/she registers the exit, a 30% discount will be applied;
 - If the car has less than 20% of the battery charge or the user leaves it in a parking area further than 3 Km from a charging station, a 30% extra free will be applied;
- The system must provide the user the possibility to select the money saving mode: with this option activated, the user has to input the final destination and the system will tell the user the best charging station where to leave the car in order to receive a discount. The system chooses the right charging station based on the position of the other cars and on the availability of power plugs to provide a balanced number of vehicles on each zone of the city.
- To close the bill, the user must park the car in one of the predefined parking areas. Then the system will detract the correct amount of money from user's payment method. The car will then set as available again when the battery charge reaches 100%.

Domain properties

Since the world is the base of the software, we assumed it in this way:

- Users have all a smartphone from where they can use the mobile application and their smartphones have GPS;
- Users have always internet connection from their devices in order to tell to the system that he/she is nearby;
- GPS give always precise information about car position;
- GPS cannot be faked or modified by anyone;
- All cars have GPS and the system can track them;
- All cars are the same and they consume the same amount of battery charge;
- If a car has a failure, it can't be used until it is repaired.
- When in maintenance, a car is out of service;
- Cars have a screen that shows,
- Cars have different sensors such as charging sensor and battery level sensor;
- To check if there is more than 1 passenger in the car, car seats have a sensor that observes a passenger's weight: if it gets a value greater than 40 Kg, the system will evaluate it as a passenger.
- There are 2 different kinds of parking: "safe areas" and "special parking areas";
- Special parking areas have sensors that measure the number of in-use plugs;
- The only compatible chargers for the cars are in the special parking areas.

Glossary

- Agreed zone: general name that identifies both the safe area and the special parking area (explained later).
- Charging spot: is a parking slot provided with an electrical plug used to charge the car battery.
- Charging station: synonym for special parking area.

- Credentials: these are the personal information of a user. They include first name, last name, email address, password, IBAN (explained later) or credit card number and geographical position.
- GPS (Global Positioning System): is a positioning system based on triangulation with satellites in order to give the exact position of a device in the world.
- IBAN (International Bank Account Number): is a unique string that locates a bank account.
- Parking lot: synonym for safe area.
- Safe area: is a parking agreed by the car sharing company where users should park rented cars. There are a certain number in the city and they are equally distributed in the territory.
- Special parking area: is a parking with charging spots agreed by the car sharing company where users should park rented cars. Users can park cars here and they can charge them. Special parking areas ARE safe areas, on the contrary, safe areas are not all special parking areas.
- UU: abbreviation for “unregistered user”.

Assumptions

- The only actor is the User: we decided to assume that all the system management and bureaucracy part (intended as fine payment, car maintenance, car retrieve and stuff like this) will be handled by a third party, so it concerns another system that will not be developed.
- There is a limited number of cars, stations and plugs so, if a user selects the money saving options and there are no available plugs, he/she will not receive a discount.
- The discounts can be accumulated and the system applies all of them on the amount calculated by time. The maximum discount that a user can get is 90%.
- When the user uses the service from mobile application, he/she has to share his/her location.
- When users book cars, they cannot revert the operation.
- Users can park the car in unsafe areas.
- If a user wants only to park the car, he has to select the “Pit Stop” option on the screen and then he can take the car back within a day if it is parked outside of the predefined zones.
- When a user wants to rent a car, the system will show only the full charged ones;
- All cars, after users end their trip a safe area, must be at full battery charge before being set as available again.

Constraints

The system can run on a mobile device (such as smartphones and tablets via the mobile app): the mobile device should have GPS, internet connection and some storage space where to install the application. When users access the service, they have to share their position in order to allow the system to show up the nearest parking.

Other important constraints to the system are the DBMS, that will store all user data, such as their credentials, payment method and email address, and the access to a SMTP server that allows the system to send email to the users.

The system interfaces with a third party system that relies on a different database that stores all information about cars such as fines written on them, car documentation (like revision papers), position and other important data.

The cooperation between this third party system and the software-to-develop will be explained better in the “functional requirements” section of this document.

Proposed System

We think that the best solution for the implementation is to build the software for a mobile application in order to allow everyone to use the service anywhere and anytime.

This implementation will allow users to benefit of the service everywhere in order to allow maximum accessibility in any condition.

The application will be available for all platforms (Android and iOS) in order to cover the most part of the smartphone users, and maybe, in future development, will be implemented a Windows Phone version too.

Identifying Stakeholders

In case this software has to be released, the stakeholders could be a car sharing company that needs a software system to manage the activity of the enterprise, but in this particular case, where the software has to be developed to demonstrate the ability to find requirements, test and other things, the stakeholders are the professors.

Our main goal is to demonstrate that we are capable of a work like this and that we can organize ourselves to perceive the same objective, even if we cannot encounter in real life due to different personal obligations.

Talking about the final user of this kind of system, we think that the most probable is the common citizen that lives in a city with a lot of traffic that doesn't want to use public means of transport for delays and other discomforts. Our solution could resolve a lot of people's problems.

ACTORS IDENTIFYING

The only actor that will use our system is the User. It is the representation of the common citizen that uses the service for moving/traveling purpose. It will use the system by registering an account, renting cars, notifying the system that it is nearby the car, locking and unlocking the car, checking out and eventually call assistance;

REQUIREMENTS

Functional requirements

Requirement 1

Allow the users to register in the system and to become part of the community:

- The system must be able to check driving license and payment method authenticity and validity.
- The system must send a password for login at the e-mail address provided.
- The system must provide to the user a redirection link for the activation of the account within the welcome e-mail.

Requirement 2

Allow the user to select a parking lot from a subset of them where to pick up a car:

- The system must be able to provide the list of all parking lots with at least one free vehicle, from the nearest to the furthest, within 1 km from the user's position or from the one inserted by user.
- The system must be able to detect user's location according to user's GPS.
- The system must be able to provide basic information about the car, such as tariff.

Requirement 3

Allow the user to reserve a car:

- The system must allow the user to book a car from the selected parking lot.
- The system must show the remaining pickup time to user.
- The system must remove the car from the available ones according to the user's choice.
- The system must be able to lock a reserved car until the user that booked it, unlocks it.

Requirement 4

Allow the user to unlock the car:

- The system must remove the car from the available ones until the user ends to use it.
- The system must be able to detract 1€ from the user's payment method if he does not pick up the car within the hour.
- The system must provide a feature in order to unlock the car.
- The system must be able to detect the car's location according to the GPS of the car.
- The system must be able to detect user's location according to user's GPS.
- The system must notify the user when the time is up if the car has not been unlocked and about the detraction.
- The system must add the car to the available ones if the car has not been unlocked in time.
- The system must be able to nullify the reservation in order to not permit the user to pick up the car if the time is up.

Requirement 5

Allow the user to use the car and to get a discount:

- The system must be able to check the time that passed from the first ignition of the motor to the checkout, in order to evaluate the final ride cost.
- The system must provide the update information about the actual ride cost to the user in any moment from the app and the built-in screen
- The system must apply the discount at the end of the ride and let know the actual cost to the user from the application
- The system must check the value of the sensors under the seats of the car.
- The system must check the car battery charge at the end of the ride.
- The system must check if a car is plugged or unplugged before the checkout

- The system must check the final position of the car after the checkout, from the car's GPS, and if it's in a safe park or not.

Requirement 6

Allow the user to select the money saving mode:

- The system must provide to the user the possibility of activating the money saving option.
- The system must be able to locate the final destination provided by the user
- The system must be able to locate the nearest safe parking areas to the final destination
- The system must be able to tell the user of the destination provided exists
- The system must be able to provide to the user a list of possible safe parking areas ordered according to: number of free parking slots, number of plugged, charged and uncharged cars and the distance of the single parking area from the final destination.
- The system must be able to check the number of charged, uncharged and plugged cars in any parking area of the Company
- The system must be able to know the number of parking slots in any parking area of the company

Requirement 7

Allow the user to finish to use the car:

- The system must be able to detect if a car is in a safe parking area
- The system must allow the user to lock the car
- The system must be able to set the car as "to be charged" mode after the user locks it
- The system must be able to detract the right amount of money with the method of payment provider by the user after the locking of the car in "stop" mode.

Requirement 8

Allow users to pay via IBAN or credit card:

- System should automatically detract the correct amount of money from the payment method.
- System adds a not-payd receipts to user's account if the transaction fails.
- System must allow users to pay not-payd bills by clicking the relative button in the application.

Non-functional requirements

[...]

SCENARIO IDENTIFYING

Scenario 1

John is an employee who works in an important company in Milan but he lives in the suburbs, far from his workplace. John has already obtained the driving license and he haven't bought yet a car of his own so decides to rent a car to go to work, since the public means of transport are too expensive and too slow and there are car sharing parking near his house and his workplace.

He downloads the app, registers an account and books a car from the nearest parking; later, he arrives at the workplace and parks the car in the safe area in front of the building.

Scenario 2

Today John goes to work with a colleague of his own that has a car so he doesn't need to book a car.

Two hours before the end of the workshift, John's colleague tells him that he has to remain in the office a little longer so he cannot bring John home. Now he has to book a car from the car sharing for the end of the workshift (18.00) but the app doesn't permit it: he can book a car at most one hour before he picks up the car so he has to wait until 17.00 to book a car.

Scenario 3

John booked a car one hour ago for a possible meeting with his friends. It is almost time to pick up the car when John's phone rings: is his friend Bob and he tells that the meeting is going to miss for a personal obligation.

So John doesn't pick the car and the system detracts automatically from his payment method 1€ fee.

Scenario 4

Today is an important day for John because he has an interview in Rome. For the special occasion, he decides to book a car to go to Rome spending less money: in order to save more, he invites 3 of his friends for a daytime visit of the capital city and he also activates the money saving option in order to get the address of the best special parking area where to leave the car.

When they arrive in Rome, they find immediately a parking spot in the previously mentioned special parking area: furthermore the battery charge is greater than 50% but John plugs the car to the charging plug anyway. so he will get a lot of discounts when he checks out from the car.

After one hour he and his friends book another car and they go around the city to visit the most famous places of Rome.

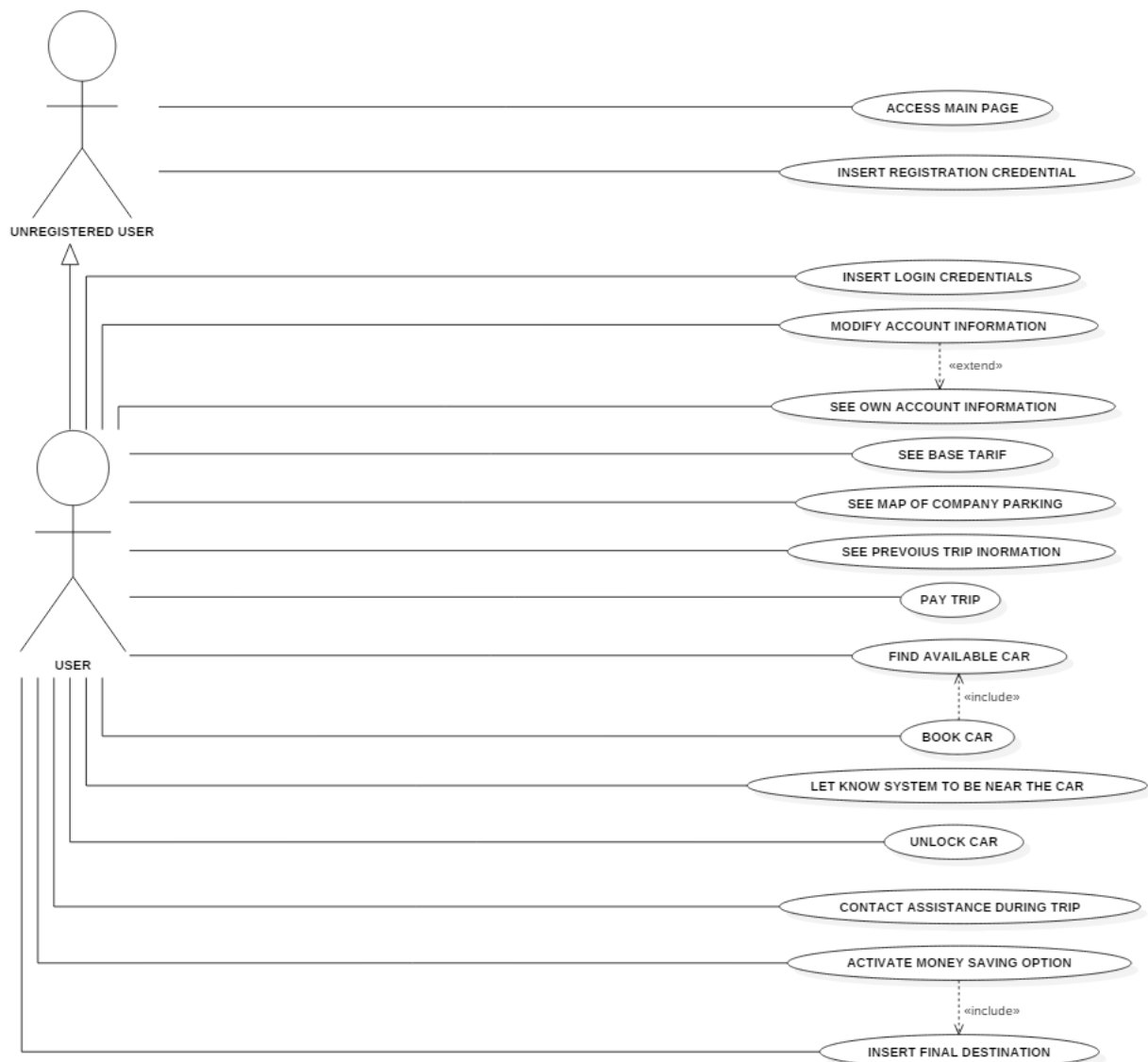
Scenario 5

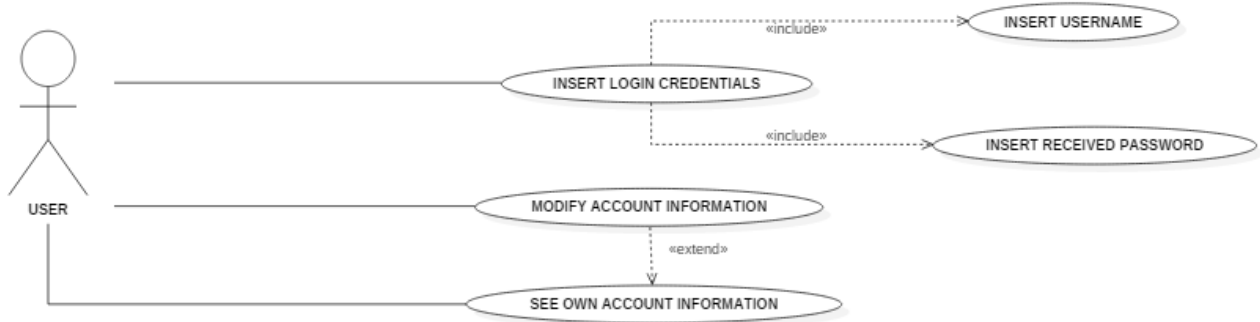
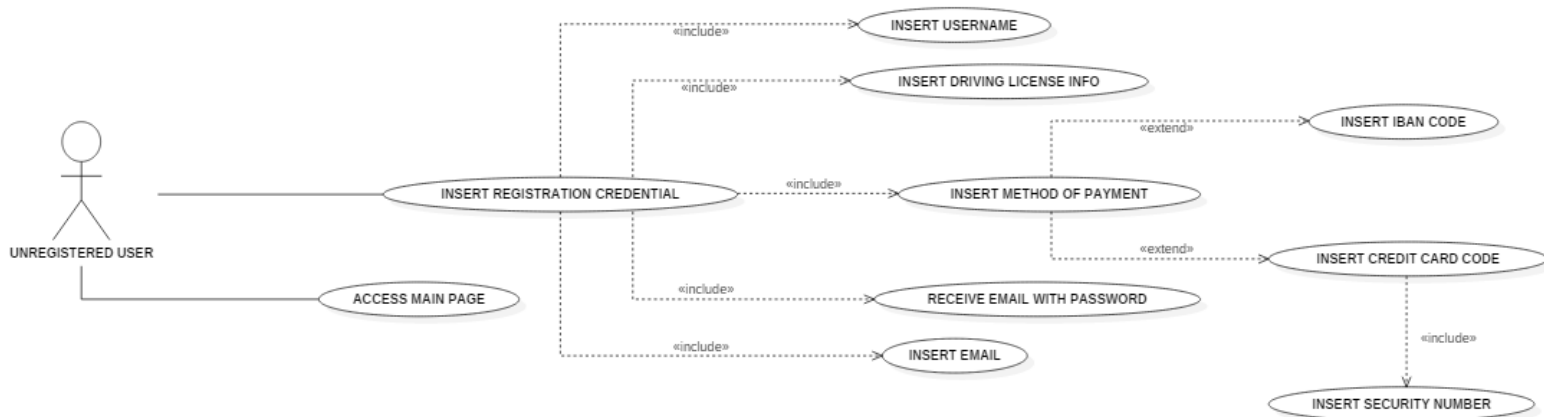
Today is a bad day for John: it's rainy, he forgot to book a car from the parking near his house so he has to get a car from a further safe area. Furthermore the traffic is particularly heavy due to the bad weather so he gets caught in it. The battery of the car runs out of charge in the middle of the road, 3,1 Km far away from the nearest charging station so he has to leave the car on the road in pit stop mode (he has 24h to

retrieve the car but he decides to leave it there until an operator recovers it) and he has to go to his workplace by foot or by public means of transport. Unfortunately he has to pay a little bit more at the end of the pit stop time because of the low battery and the parking in a not agreed zone.

UML MODELS

Use case diagrams





Insert registration credential

Actors: Unregistered User.

Entry conditions:

- The user selects to register himself into the system from the main page.

Flow of events:

- The user inserts his credentials in the featured form.
His credentials include:
 - First Name;
 - Last Name;
 - Date of birth;
 - Username;
 - E-Mail;
 - Driving license info;
 - Payment method.
- The user clicks the registration button;
- The user is successfully registered.
- The system sends a welcome mail to the user that contains his password;

Exit conditions:

- The user is redirected to the login page.

Exceptions:

- Some of the information the user provided are incorrect. The user is not redirected but is notified of the error fields.

Insert payment method

Actors: Unregistered user.

Entry conditions: No entry conditions.

Flow of events:

- The user chooses the payment between the available options and completes the related fields:
 - If he chooses IBAN, he inserts the IBAN code
 - If he chooses credit card, he fills up the following fields:
 - Credit card number;
 - CVV;
 - Holder name;
 - Expiration date.

Exit conditions:

- The system displays a notification of the correctness of the information provided and enables the registration button.

Exceptions:

- The system displays an error notification associated to the incorrect fields and disables the registration button.

Insert login credentials

Actors: User.

Entry conditions:

- The user is registered and has received the password.

Flow of events:

- The user opens the app for the first time;
- The user inserts his credentials that include:
 - Username or e-mail;
 - Password.

Exit conditions:

- the user is redirected to the research page of the app where he can now book a car.

Exceptions:

- Some of the information the user provided are incorrect. The user is not redirected but is notified of the error fields.

See own account information

Actors: User.

Entry conditions:

- The user is successfully logged.

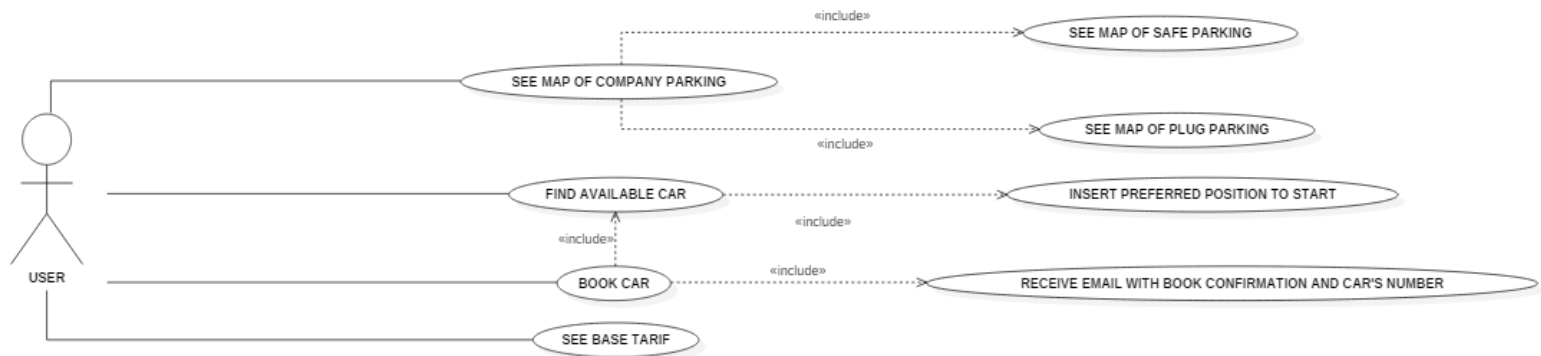
Flow of events:

- The user clicks on the account button of the app;
- The user sees his information and eventually edits them.

Exit conditions:

- The user successfully goes on another page.

Exceptions: there are no exceptions.



Find available car

Actors: User.

Entry conditions:

- the user is successfully logged.

Flow of events:

- The user goes into the research section;
- The system shows the available cars near his current position;
- If the user inserts a different position or place, then the system displays the available cars near that position;

Exit conditions:

- The user changes page;
- The user selects an available car from the system displayed options;
- The user presses "book" button;
- The user presses "map" button.

Exceptions:

- The GPS is not activated: The system doesn't display any car but an option link that redirect the user to the device settings in order to activate the GPS;
- The user inserts an unrecognized position or place. The system displays a message that no available cars are found.

Book car

Actors: User.

Entry conditions:

- The user selects an option from the research page.

Flow of events:

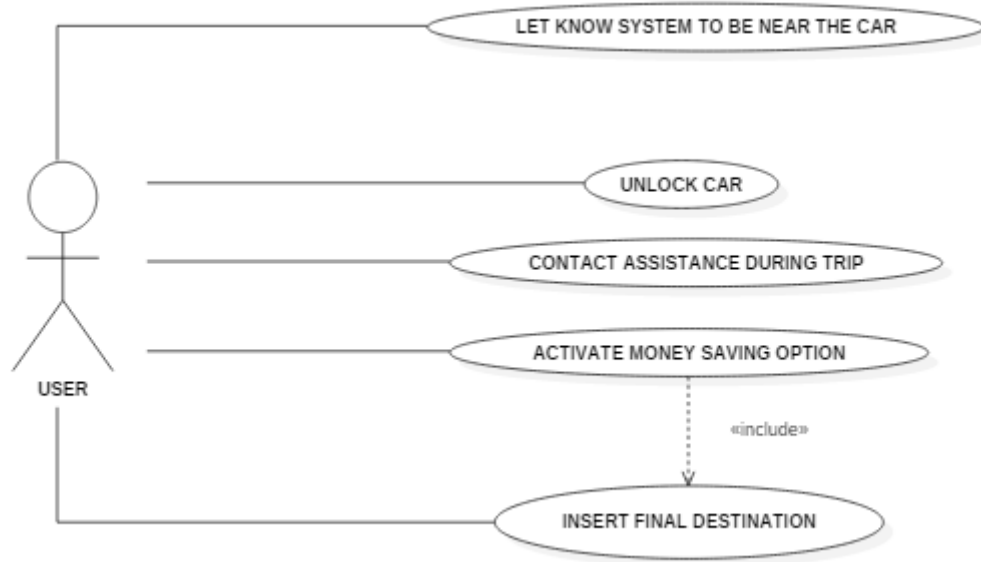
- The user sees the information related to the car;
- The user can eventually activate the money saving option;
- The user clicks on the book button;
- The system checks the availability of the payment method;
- The system checks for previous not yet paid trips.

Exit conditions:

- The system notifies the user of the successful booking, starts 1-hour countdown visible on the app as picking up threshold, redirects the user to the "car" page where he can see the vehicle information.

Exceptions:

- The user inserts incorrect information. The system notifies the user about the incorrect fields;
- There's a pending payment related to a previous trip not payed yet. The system redirects the user into the payment details section.



Unlock car

Actors: User.

Entry conditions:

- The user has booked the car less than an hour ago;
- The user has parked the car and less than the pit stop time limit of the car has passed.

Flow of events:

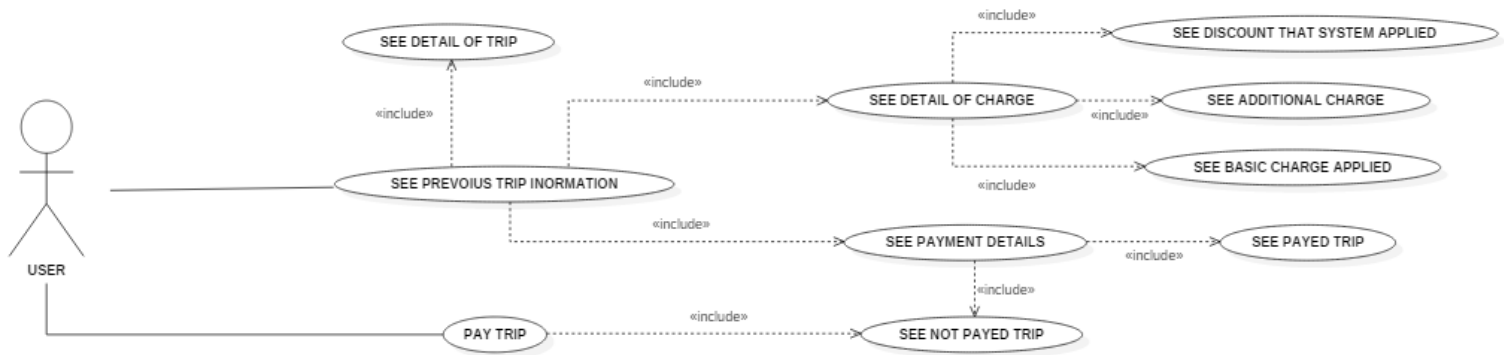
- The user accesses to the system;
- The user goes on "car" page;
- The system checks the user position by GPS and enables the unlock button if the user is near the car;
- The user clicks on the unlock button.

Exit conditions:

- The user can now use the car.

Exceptions:

- the GPS is not working. The system notifies the user and redirects him to the device settings in order to activate it;
- The internet connection is not available. The system notifies the user to activate the Bluetooth and to put the device near the car in order to unlock it;
- The user cannot unlock the car using his device. The user could ask for support to the authorized personnel.



Pay for a car

Actors: User.

Entry conditions: The user has to pay the last ride but an error occurs during the automatic checkout.

Flow of events:

- The user goes into the payment section;
- The user clicks on "Pay" button;
- The system checks that the method of payment is correct;
- The system detracts the amount of money of the last ride;

Exit conditions:

- The system notifies the user of correctness of the transaction.

Exceptions:

- The check of the method of payment fails: the system notifies the user about it and redirect him to the account settings;
- The system cannot detract the amount of requested: it notifies this to the user suggesting to control the residual credit.

Contact assistance during trip

Actors: User.

Entry conditions:

- The user is using the car.

Flow of events:

- The user selects the assistance from the display in the car;
- The system notifies the personnel of the call.

Exit conditions:

- The user is contacted by the personnel immediately.

Exceptions:

- The internet connection or the telephone network does not reach the car;
- The personnel will contact the user as soon as an internet or telephone connection will be available.

Activate money saving option

Actors: User.

Entry conditions:

- The user has booked a car.

Flow of events:

- The user access to the system;
- The user selects the money saving option from the "car" page;
- The system shows a form where the user chooses the final destination;
- The system provides information about the nearest station to the final destination where to leave the car to get a discount.

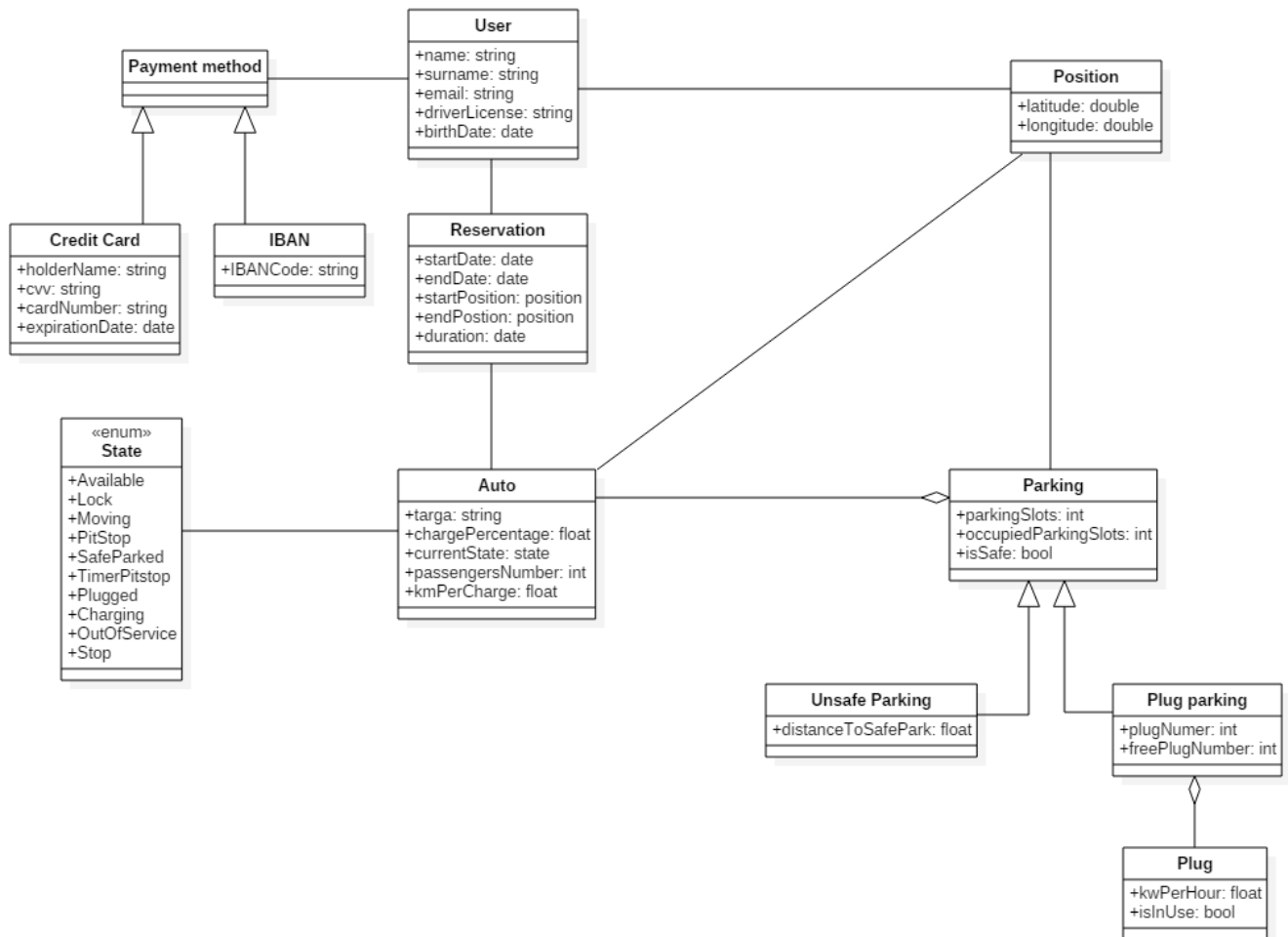
Exit conditions:

- Deactivate the money saving option;
- Drive to the station indicated by the system;
- The user makes a pit stop in an unsafe area and passes the time limit.

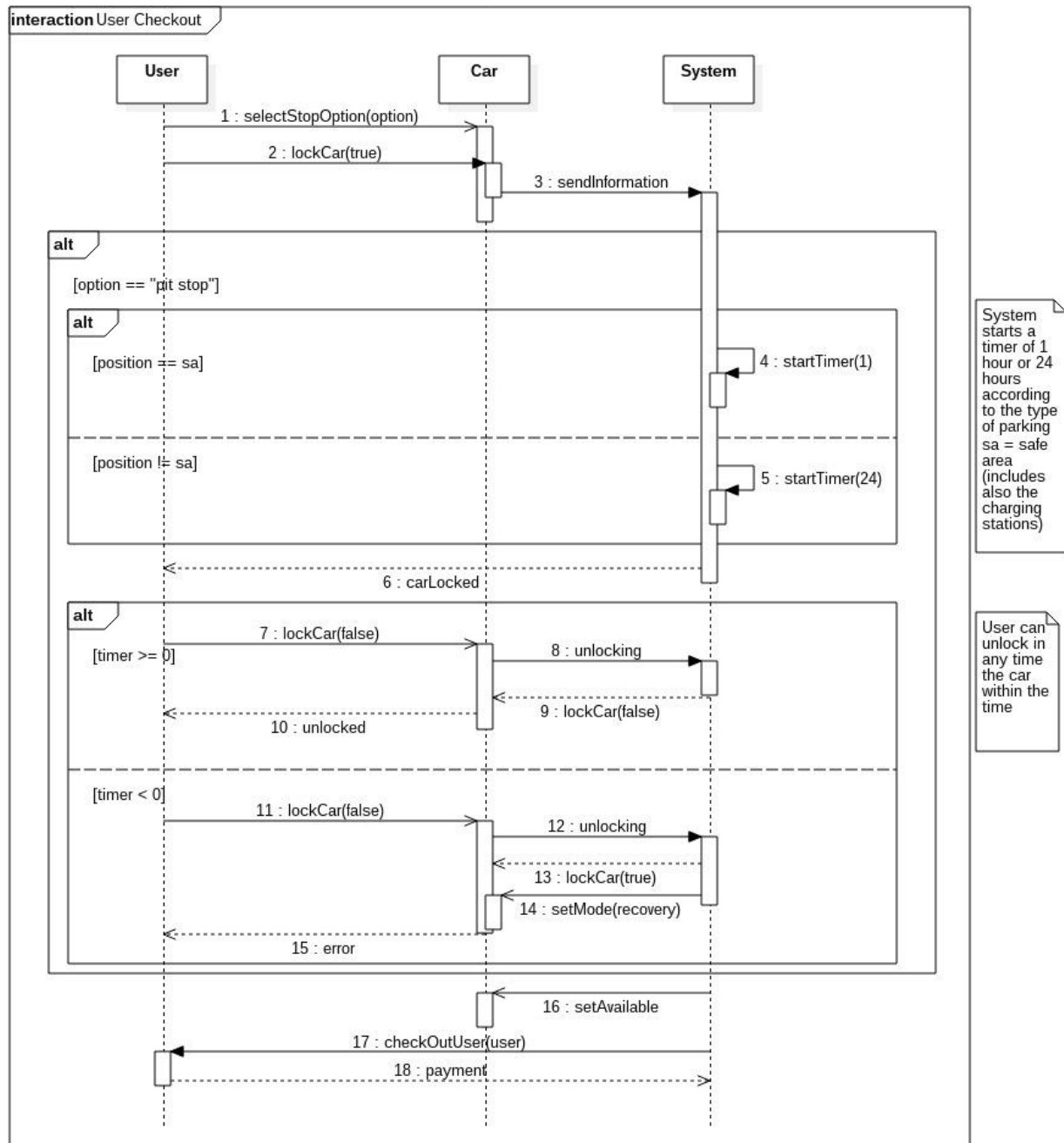
Exceptions:

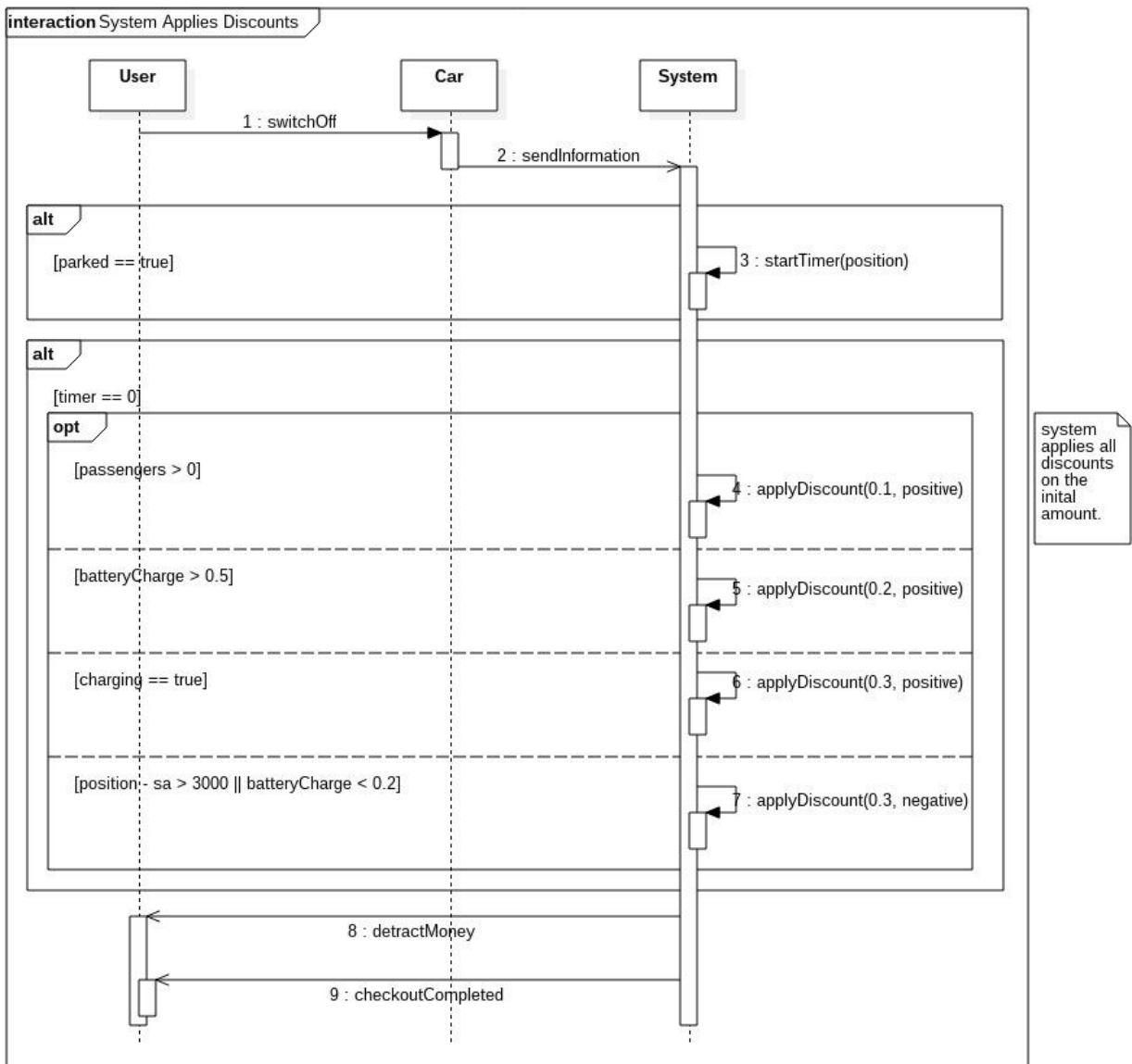
- The internet connection is not available;
- The system shows a connection error and allows the user to retry;
- The user inserts an unrecognized position or place. The system shows a "not found" error asking the user to check if the destination is correct;
- The user parked in the right place but doesn't receive a discount. The user could contact the personnel for more information.

Class diagram



Sequence diagrams





Activity diagram

State diagram

ALLOY MODELING

Model

Alloy result

World generated

FUTURE DEVELOPMENTS

USED TOOLS

HOURS OF WORK