

## **Requirements Analysis and Specifications Document**

Li Xiaoxu	10545608
Jia Hongyan	10470662
Lang Shuangqing	10517902



# Contents

1. Introduction
  1. Description of the given problem
    1. Actual system
  2. Goals
    1. System
    2. Users
  3. Domain properties
  4. Glossary
  5. Text assumptions
  6. Constrains
    1. Regulatory policies
    2. Hardware limitations
    3. Interfaces to other applications
    4. Parallel operation
  7. Reference documents
2. Actors identifying
3. Requirements
  1. Functional requirements
    1. System
    2. Users
  2. Non-functional requirements
    1. User mobile interface
    2. Car screen interface
4. Scenario
  1. Scenario 1
  2. Scenario 2
  3. Scenario 3
  4. Scenario 4
  5. Scenario 5
  6. Scenario 6
  7. Scenario7
5. UML models
  1. Use case diagram
  2. Use case description
  3. Class diagram
  4. Sequence diagrams
  5. State diagrams
6. Alloy modeling
7. Future development
8. Used tools
9. Hours of work
  1. Li Xiaoxu
  2. Jia Hongyan
  3. Lang Shuangqing
10. Changelog

## Introduction

### Description of the given problem

We will implement *PowerEnJoy*, which is a car-sharing service based on a system(mobile application), with two different targets parts:

- system
- users

The system allows users who has registered this system to rent a car via mobile or web app, using GPS position to identify user's zone and find car in their needed zone. On the other side the system will accept the users' demand and give a route to the car by GPS. The car will be unlocked when users reach a reserved car and tell the system she's nearby, so the car is available and the users may enter.

The system includes extra services and functionalities such as car sharing.

The main purpose of the [system](#) is to offer a better service for the users.

## **Actual system**

Until now the system where the users have to provide their name、 credentials and payment information、 the position and the destination.They receive back a password that can be used to access the system.Then the system will provide some suitable cars、 routes、 estimated prices and estimated time for users .

This system stores car information into a Mysql database.

## **Goals**

### **System:**

- [G1] System is fully aware and in control of the current state of the car.
- [G2] System ensure the car does not overload.
- [G3] System has the function of intelligent fee-calculate.
- [G4]System will give a discount according to the use situation of battery.
- [G5] System can provide the situation of road.
- [G6] System allows users to input a destination.

## **Users:**

- [G7] Users can register and log in .
- [G8] Users can search cars according to their own requirements.
- [G9] Users can reserve cars before renting
- [G10] Users can check and make a payment of the bills
- [G11] Users can decide the number of passengers.

## **Domain properties**

We suppose that these properties hold in the analyzed world:

- All the GPS always give the right location.
- The GPS of the cars cannot be switched off by users.
- All the cars have no problems about machine or electronic components.
- Users can successfully pay the fees immediately.
- Each user can only reserve one car at same time.
- There is no car accident.
- All the users have no problem in their health.

## **Glossary**

- System: it means the software we built. It's an application for car renting and sharing.
- User: he is a user of the software. He can apply for a car with his information.
- Network: it includes 3G, 4G and Wi-Fi.
- 

## **Text assumptions**

- The battery of the car is always full before used.
- We assume that we have a fee for each user.
- We assume that all sharing users will get into car at one place.
- We assume that all users have driving license.

## **Constrains**

### **Regulatory policies**

The system must require user the exact location, in order to provide the cars' information to users. Moreover, the system cannot show the personal information of users to others.

### **Hardware limitations**

- Mobile app
  - Network connection
  - GPS
- ~~Web app~~

➤—Browsers

➤—Network connection

➤—GPS

## **Interface to other applications**

Interface to the payment application. The PowerEnJoy will connect with the online bank service system or application.

## **Parallel operation**

The PowerEnJoy supports parallel operations from different users. It can solve multiply requests at the same time.

## **Reference documents**

- RASD sample from Oct. 20 lecture.pdf

## **Actors identifying**

The actor of our system is totally one:

- User: he can rent a car with register or not. But no matter he register or not, he should insert his current information including name, phone number, location, and the most importantly, the driving license.



## Requirements

### Functional requirements

Assuming that the domain properties can be implemented in the paragraph[1.3] hold, and in order to fulfill the goals in paragraph[1.2], the following requirements can be gained.

The requirements are grouped under each goal from which it is derived.

The goals are grouped following under the users concerned.

#### System:

- [G1] System is fully aware and in control of the current state of the car:
  - The system must be able to know how many passengers are on car now.
  - The system must be able to ensure the position of each car.
  - The system must be able to know the current battery amount of the car.
  - The system can decide the car is locked or unlocked. And as soon as the car is parked in a safe area and the user exits the car, the system can lock the car automatically.
  - The system should make sure the battery of each car is full when users rent them.

~~—The system should make sure the battery of each car is full when users rent them.(move to domain properties)~~

- [G2] System ensure the car does not overload.

-The system can make sure the passengers never overwhelms the legal capacity of the car..

- [G3] System has the function of intelligent fee-calculate.

-The system can give users a estimated price after users provide the start position and destination.

-The system can charge the user for a given amount of money per minute; the user will notify the current charges through a screen on the car.

-The system detects the user took at least two other passengers onto the car, the system applies a discount of 10% on the last ride.

-The system can allow the user to choose the money saving option, he/she can input his/her final destination and the system provides information about the station where to leave the car to get a discount.

-The system allows the first users to choose sharing the car or not if users would like to share car they will both get discount during the sharing car time.

- [G4] System will give a discount according to the use situation of battery.

-The system detects the car using only left with no more than 50%

of the battery empty, the system applies a discount of 20% on the last ride.

-The system finds users leaving a car at special parking areas where they can be recharged and the user takes care of plugging the car into the power grid, the system applies a discount of 30% on the last ride.

-The system detects a car is left at more than 3 KM from the nearest power grid station or with more than 80% of the battery empty, the system charges 30% more on the last ride to compensate for the cost required to re-charge the car on-site.

-The system will warn users on screen when the battery has 60%, 50% and 20% left.

- [G5] System can provide the situation of road.

-The system will match GPS function and give users shortest route.

-If the money saving option is on, the system provides information about the station where to leave the car.

-The system can provide the set of safe areas for parking.

- [G6] System allows users to input a destination.

-The system require users to input destination ,then system will recommend some parking areas near destination for users.

## **Users:**

- [G7] Users can either log or register .

-Users can register the system and management their personal information, when users register in this system, they will receive a password which will be used to login in this system.

- [G8] Users can search cars according to their own requirements.

-Users can provide their starting position and destination then system will give users some shortest routes and discount options.

-Users can specify the number of passengers they take.

- [G9] Users can reserve cars before renting.

-Users must be able to reserve a single car for up to one hour before they pick it up.

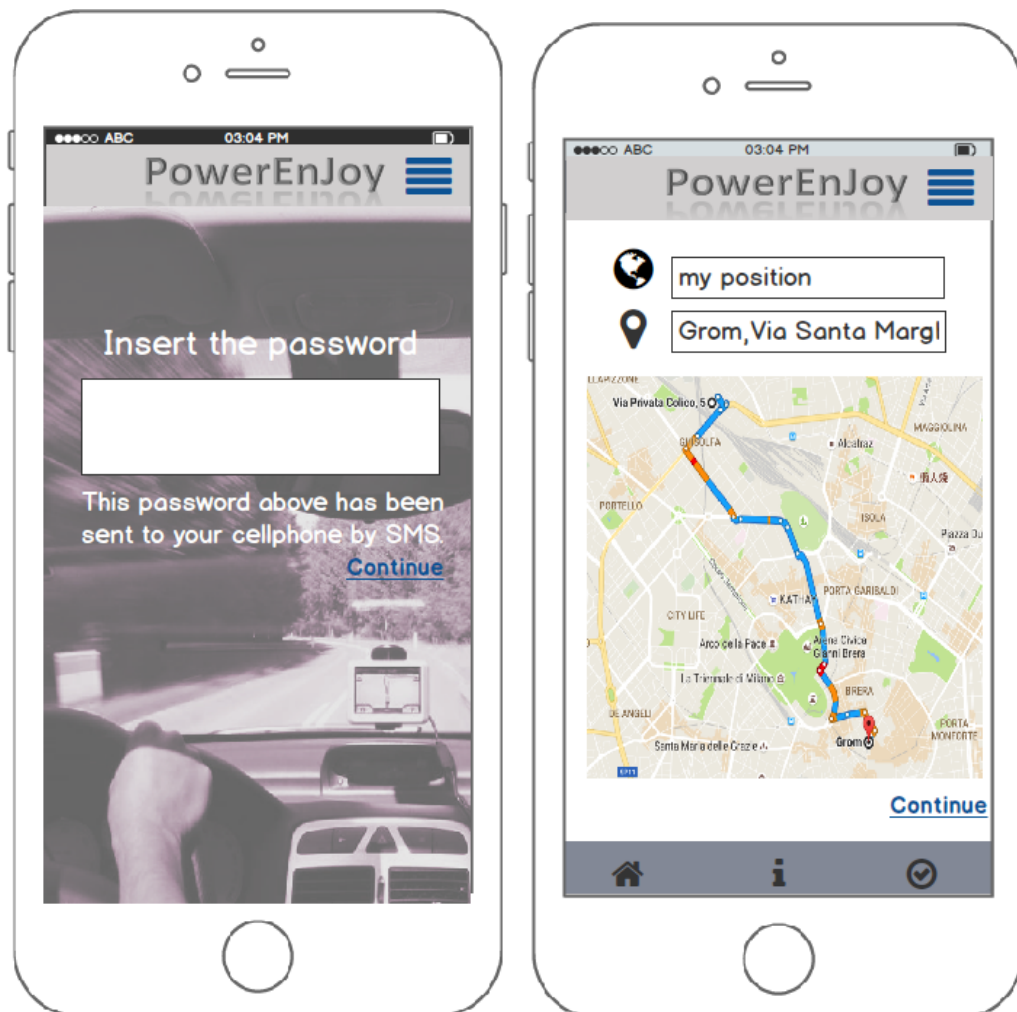
-The user pays a fee of 1 EUR, if a car is not picked-up within one hour from the reservation.

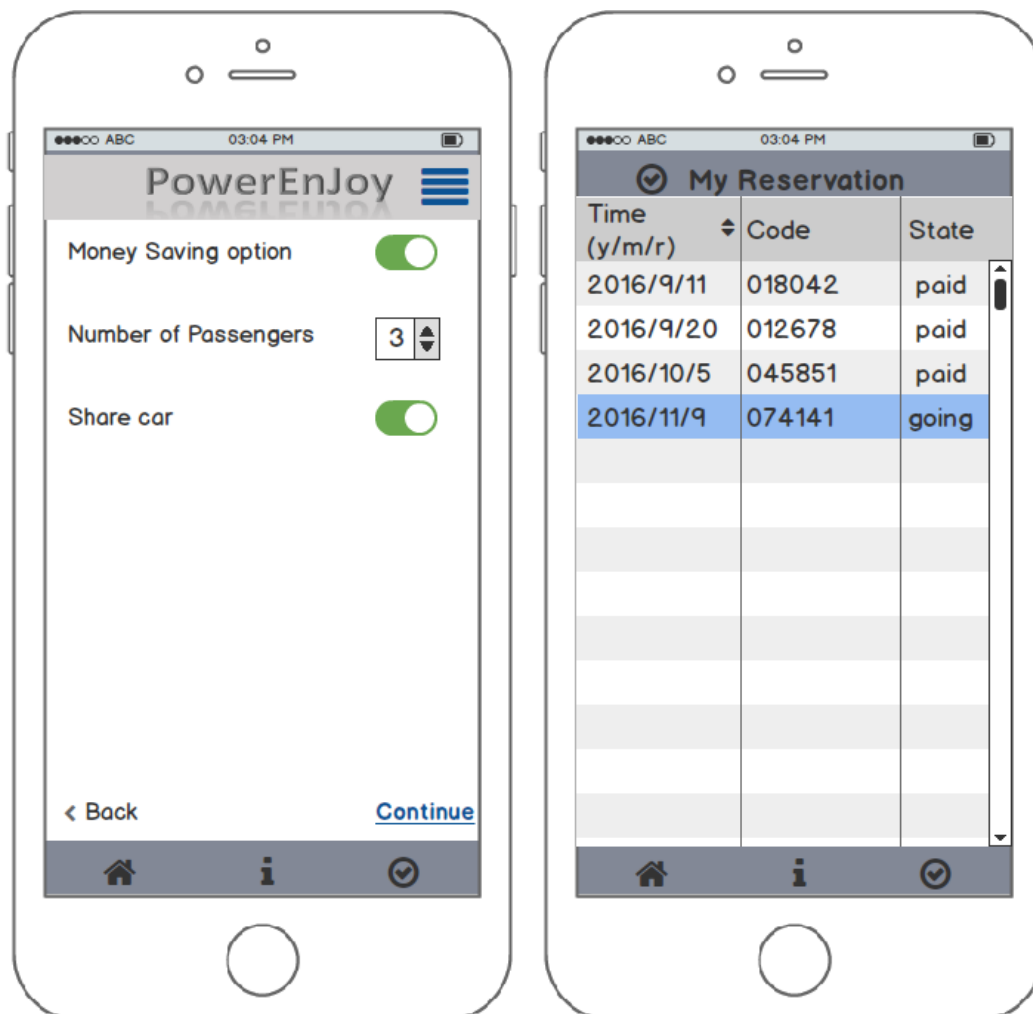
-Users when reach near the car tell the system she's nearby, the system unlocks the car and the user may enter.

- [G10] Users can check the bill and make a payment.
  - Users can check the bills after getting off. If users confirm the price and discount, they can pay the bill.
- [G11] Users can decide the number of passengers.
  - Users who drive the car can decide to share it or not.
  - Users can select up to four passengers.

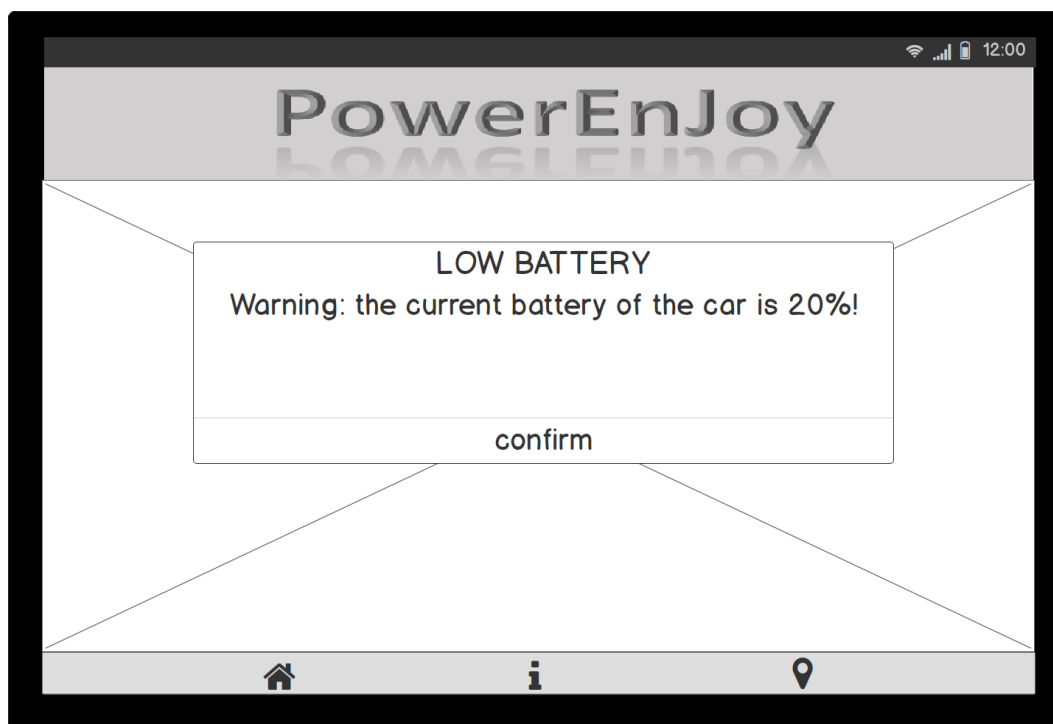
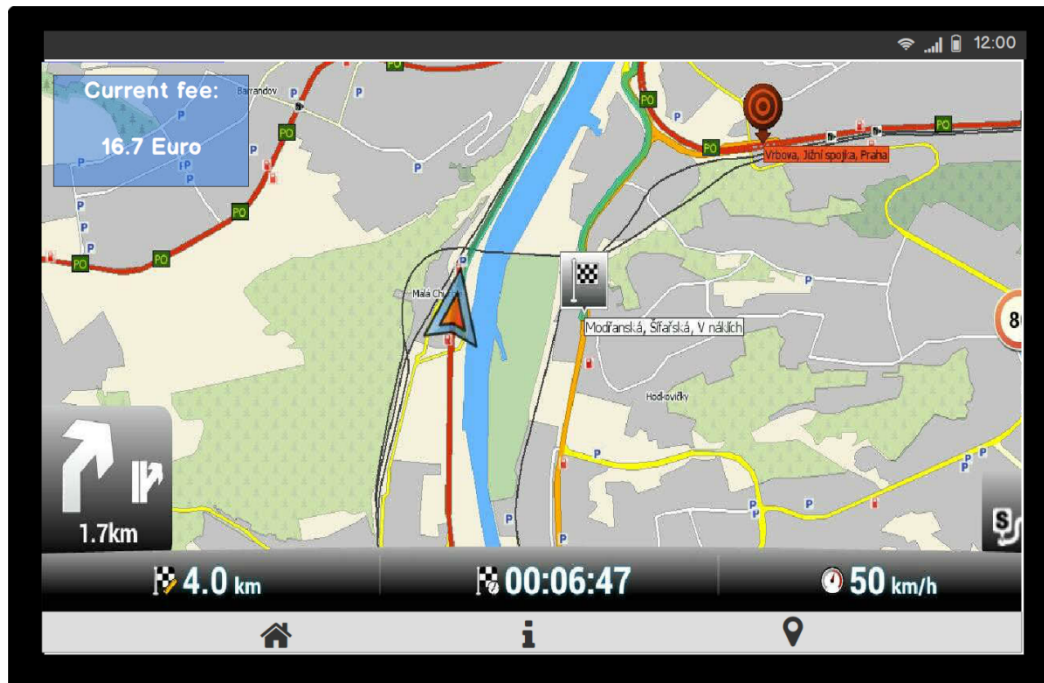
## Non-functional requirements

### User mobile interface

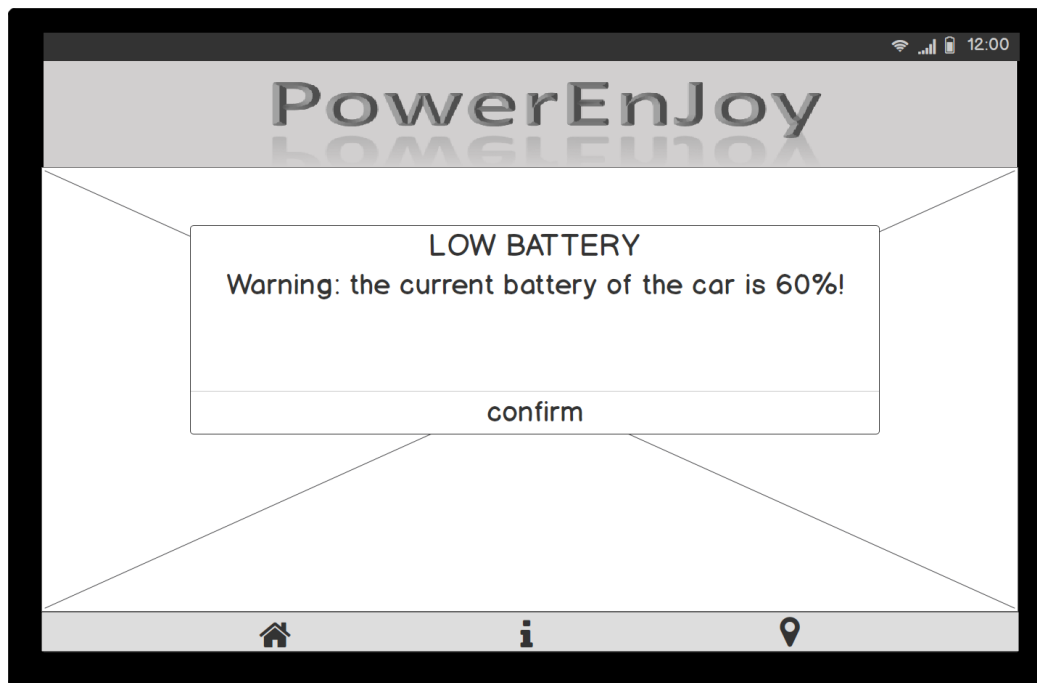




## Car screen interface







## Scenario

### Scenario 1

Jack just comes off work at 7pm, he wants to drive around the city to wind down. So he opens *PowerEnJoy* to write the start point and destination to rent a car. When he comes near the car just telling the system then the car will be unlocked until jack put it in parking area.

### Scenario 2

Jack wants to go to work and have adequate time. So he chooses to share car for a discount with another two passengers and they do not know each other before.

At 7am Jack leaves home. He opens *PowerEnJoy*, write down the initial position and destination and opens sharing function. The system chooses two matched passengers to take the car so that they three will all get the discount during sharing car time. Then Jack and another two passengers meet at the position of the car.

After Jack gets off the car one of the other passengers becomes the driver, reach their destination and parking the car in the special area.

### **Scenario 3**

Jack will go to airport at 2pm, he makes a schedule, opening the *PowerEnJoy* to reserve a car in 13.30pm, he just chooses reservation option input use time and site. After submitting the reservation, he receives a confirmation message which shows his reservation will be performed and of the information about the car. Near 2pm when Jack nearby the car, the system unlocks the car and he may enter.

### **Scenario 4**

Mason has been invited by his friends Jack to attend a party. There is a long distance between the Mason's home and Jack's and Mason leave home so late. Mason drive as quickly as possible, he parks the car at more than 3 KM from the nearest power grid station and with more than 80% of the battery empty, the system charges 30% more on the last ride to compensate for the cost required to re-charge the car on-site.

### **Scenario 5**

Tom wants to go to stadium for a fantastic football match, for an economic way he chooses a money saving option. At the weekend Tom has plenty of free time, so he decides to leave home early to choose *PowerEnJoy* money saving option which will make him to park car in

special area where users can be recharged and the user takes care of plugging the car into the power grid. So Tom use short time accomplish this mission and parks the car in the special area then walking a short distance to destination. He gets a discount and watches a fantastic football match.

### **Scenario 6**

Julia plans to see a film. Her friend Jack recommends PowerEnJoy to Julia, then she reserves the car in 6pm. But she spends plenty of time to make up at home she reaches the car at 6.10pm. Julia must pay 1 Euro because the system rules users must pick up the car within one hour. So the reservation expires and Julia pays a fee of 1 EUR. She has not used PowerEnJoy successful.

### **Scenario7**

Beck wants to go to a park for visit , he opens PowerEnJoy to rent a car. For leading a economic and health life, Beck decides just to use 50% of the battery empty, then the system applies a discount of 20%. After Beck parks the car in parking area he walks about 10 minutes to reach the park.

### **Scenario-8**

~~Julia has yoga course in the centre of Milan , she wants to go there and back use the same car for saving time ,.For reducing the cost of the rides she decides to try to share the car with someone also has yoga course.Then she has to fill a form to indicate all the information about the reservation .After doing this she has to repeat the whole operation to reserve the return trip only for getting a car as quickly as possible after yoga course.(remove)~~

## UML models

### Use case diagram

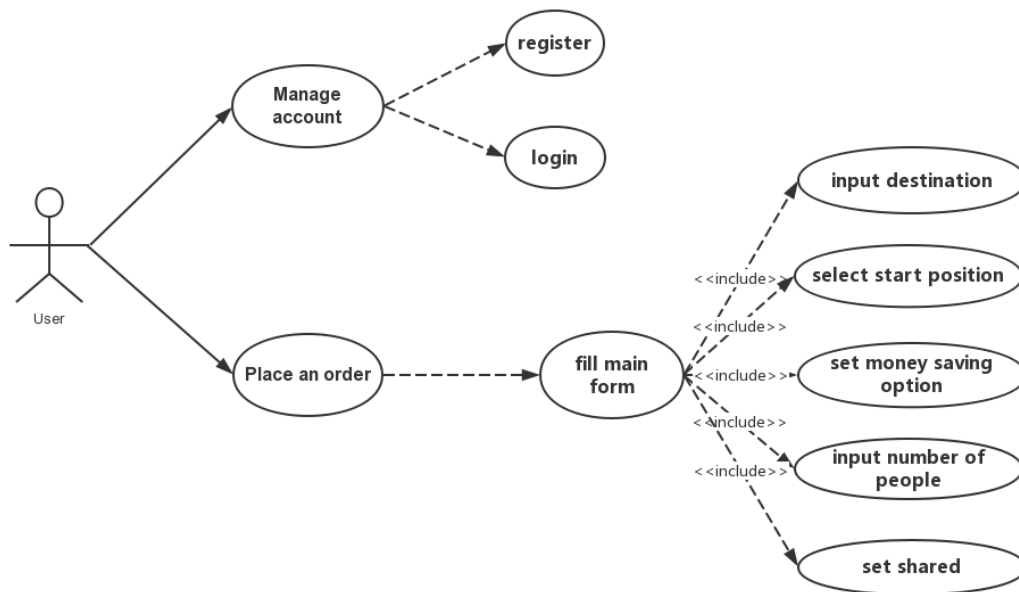


Figure 1: use case diagram

### Use case description

In this paragraph some use cases will be described. These use cases can be derived from the scenarios and the use case diagram.

---

#### User manages account

**Name:** User manages account

**Actors:** User

**Entry conditions:** There are no entry conditions.

**Flow of events:**

- The user inputs his personal details to create an account.
- The system send passwords back to user.

- The user can log in the system by username and passwords.

**Exit conditions:** The user successfully logs in his account.

**Exceptions:** There are no exceptions for this use case.

---

## User place an order

**Name:** User places an order

**Actors:** User

**Entry conditions:** The user must have registered and logged in.

**Flow of events:**

- The user presses the “rent a car” to input relevant information.
- The user fills the form about positions and number of people.
- The user decides if he wants to share the car with other users.
- The user decides if he wants to get some discounts according to some special rules.
- The system will show user the estimated fees.
- The user confirms the order.
- When user arrives to the car which the system recommended, the user should tell the system he has arrived in order to unlock the car.

**Exit conditions:** The user arrives to his final position and pays the fees.

**Exceptions:** There are no exceptions for this use case.

## Class diagram

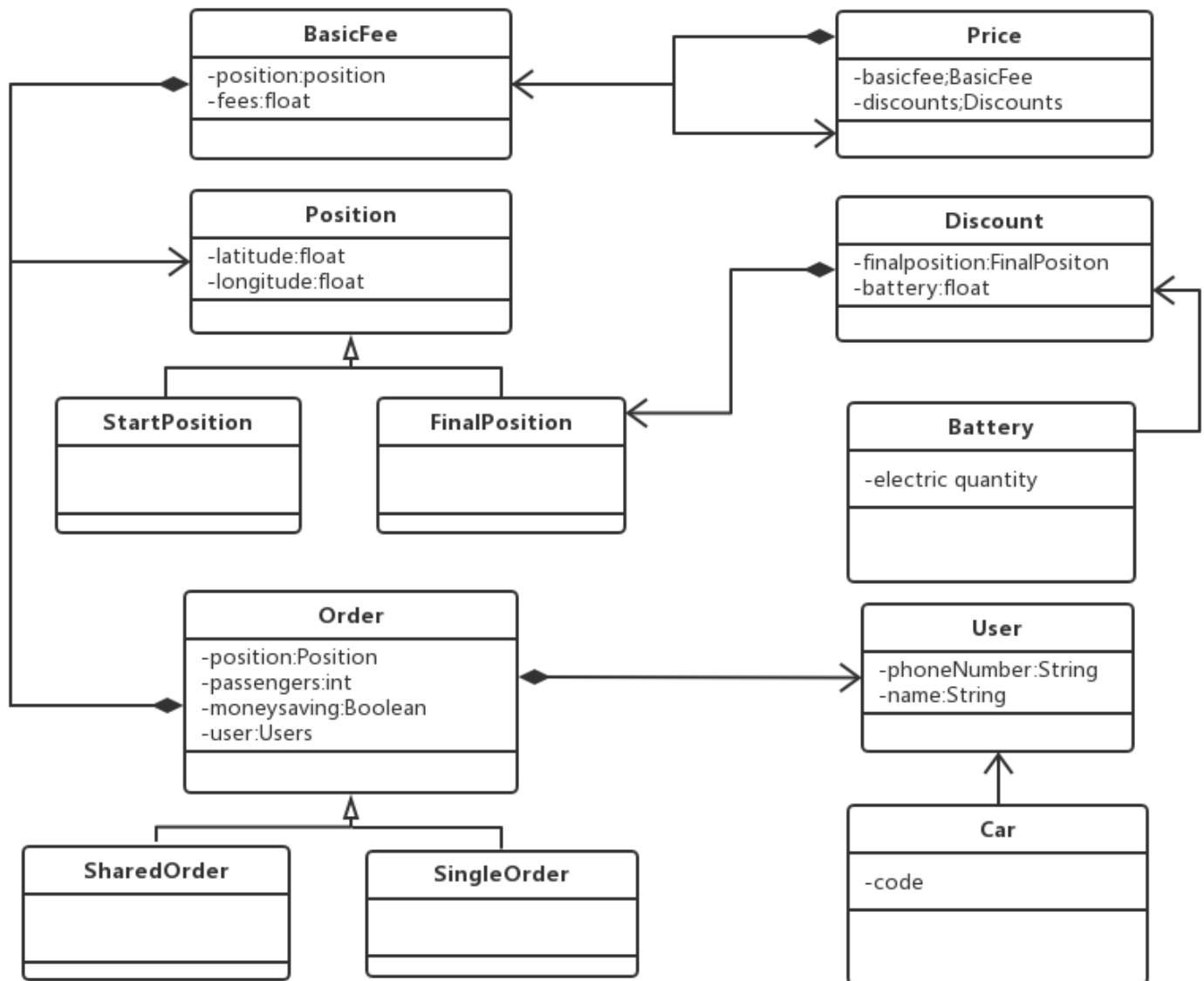


Figure 2: class diagram



## Sequence diagrams

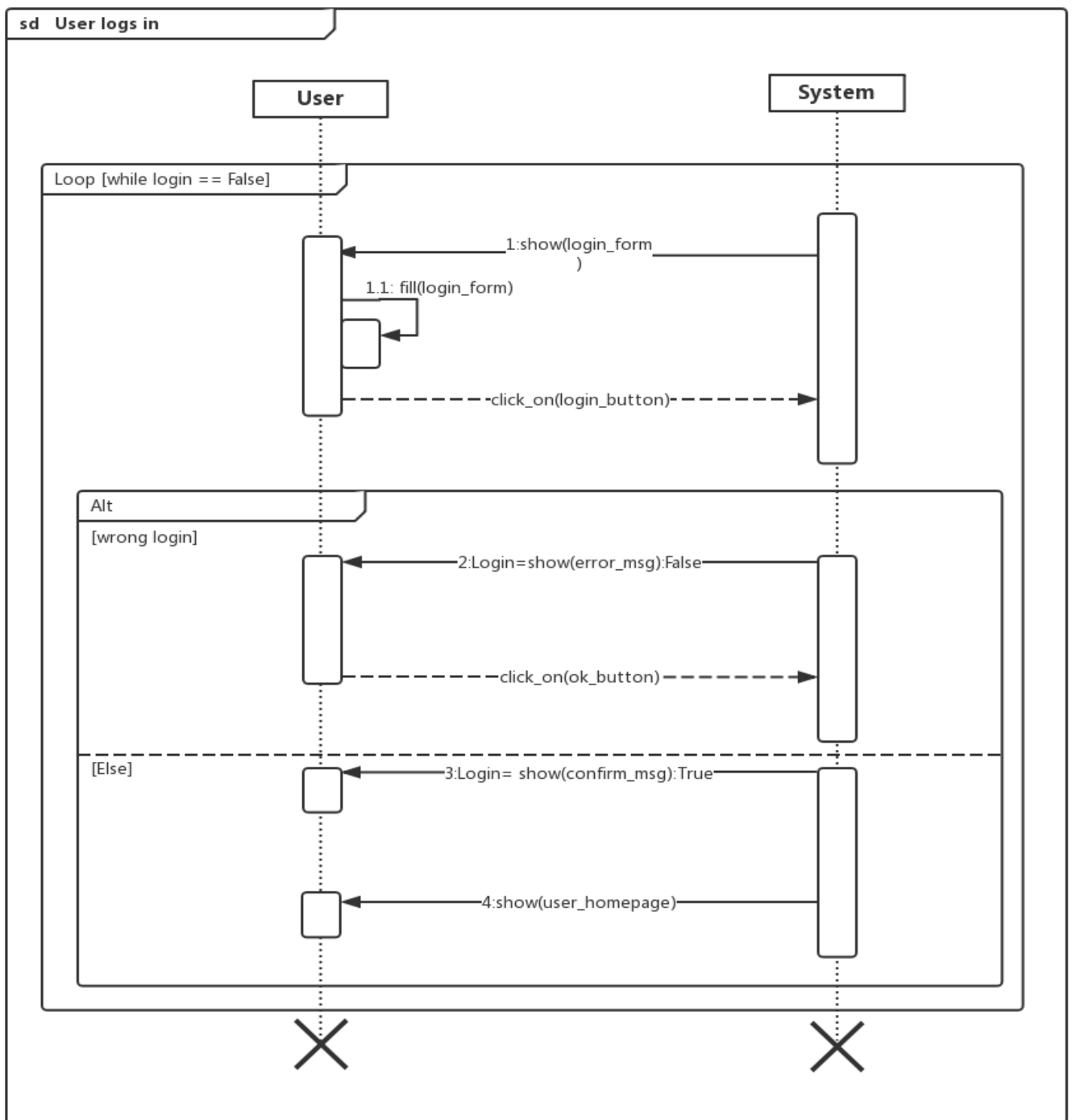


Figure 3: sequence diagram I

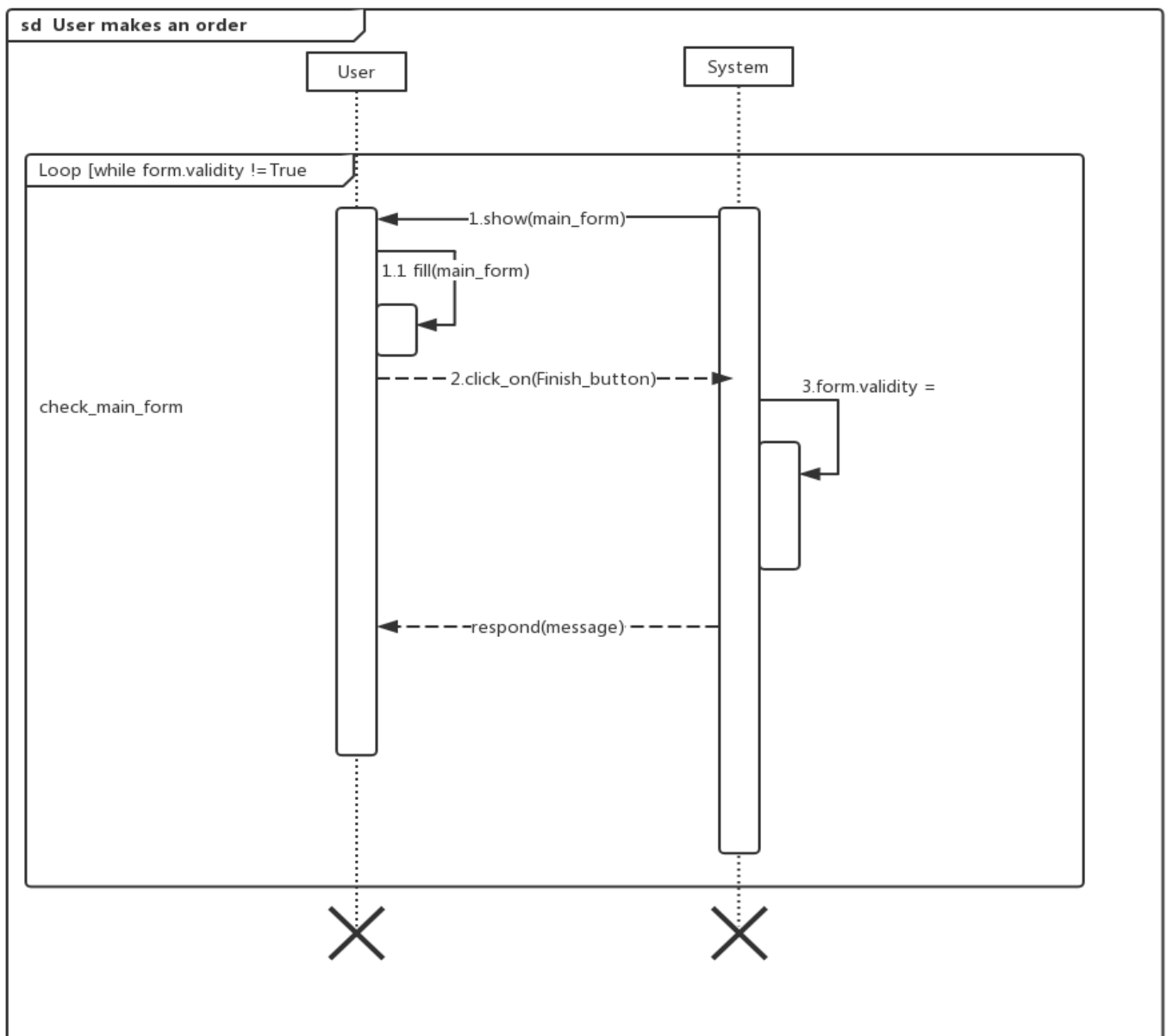


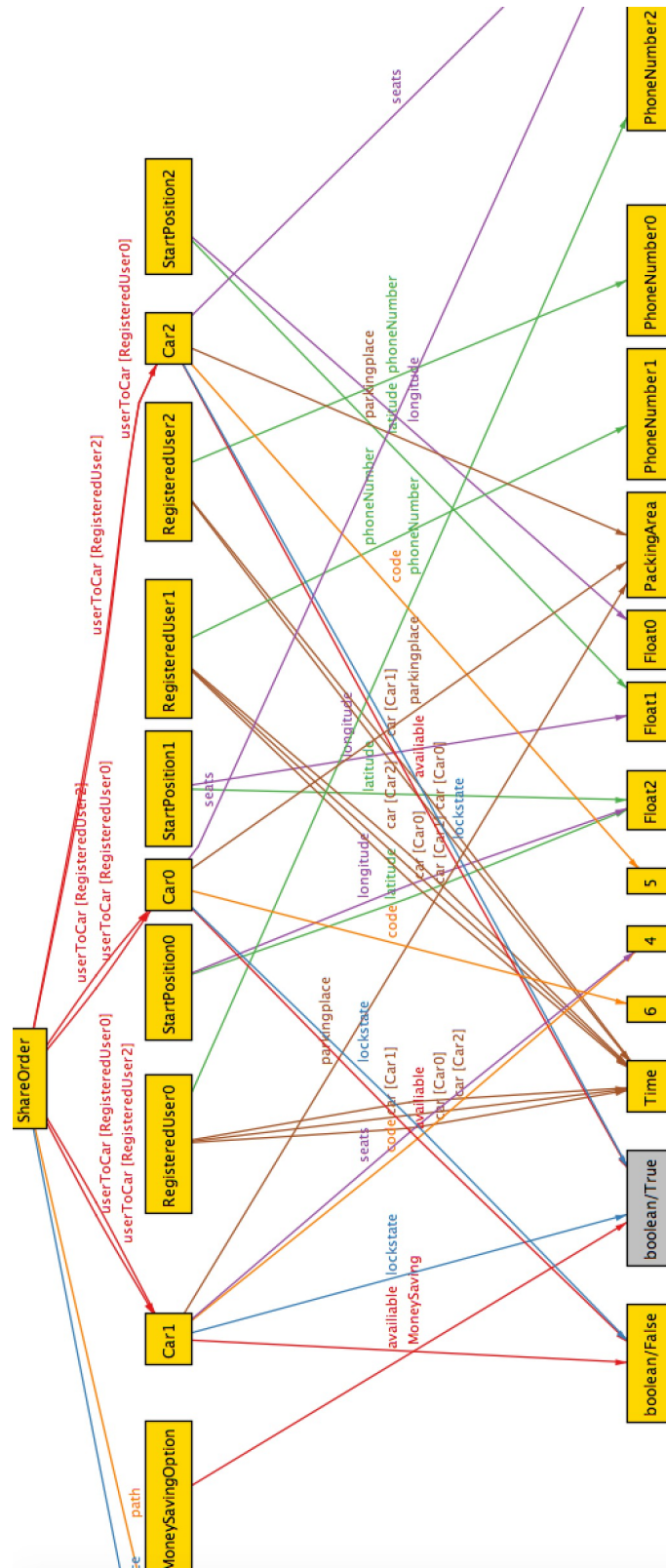
Figure 4: sequence diagram II

## State diagram



Figure 5: state diagram

# Alloy Modeling



## **Future development**

Allow users to rent the car all day which will be convenient for users who want to travel around the city at weekend and festival.

## **Used tools**

**WPS:** For editing text.

**ProcessOn:** a tool makes UML part.

**Google Drive:** for version controller.

**Balsamiq Mockups 3:** for making the users interface.

## **Hours of work**

Li Xiaoxu

- 20/10/16: 2h
- 21/10/16: 2h
- 23/10/16: 2h
- 24/10/16: 2h
- 26/10/16: 1h
- 28/10/16: 1h30
- 29/10/16: 3h
- 30/10/16: 5h
- 31/10/16: 4h
- 02/11/16: 1h30
- 05/11/16: 3h
- 07/11/16: 3h
- 08/11/16: 6h
- 10/11/16: 3h

Jia Hongyan

- 23/10/16: 3h
- 24/10/16: 2h30
- 27/10/16: 2h30
- 28/10/16: 2h
- 31/10/16: 2h
- 03/11/16: 3h30 morning + 2h night
- 05/11/16: 2h
- 07/11/16: 3h
- 09/11/16: 4h
- 11/11/16: 2h
- 13/11/16: 3h

Lang Shuangqing

- 22/10/16: 2h
- 24/10/16: 2h
- 26/10/16: 3h
- 28/10/16: 2h
- 31/10/16: 4h
- 01/11/16: 2h
- 03/11/16: 3h
- 05/11/16: 4h
- 06/11/16: 2h
- 08/11/16: 2h
- 09/11/16: 3h
- 11/11/16: 4h
- 13/11/16: 3h



## Change log

### V1.1

-[G1] The system should make sure the battery of each car is full when users rent them. (move to domain properties).

-Scenario1 fixed.

-Scenario8 removed.

-Alloy part fixed.

### V1.2

-Requirement fixed.

-Uml models modified.

-Web app removed

### V1.3

-Modified regulatory policies.

-The overall format fixed.

-Add the function of battery warn.