Politecnico di Milano

A.A. 2016-2017

Software Engineering 2: “PowerEnjoy”

Requirements Analysis and Specifications Document

Version 1.5

Prasanth Ravulapalli

Fathima B

Lipika L

November 11th 2016

Contents

Introduction

Description of the given problem

We will project and implement PowerEnjoy, which is a service based on mobile and web based digital management system for car sharing which exclusively employs electric cars. It targets two set of people:

- Guest users
- Registered users

The system should allow the user to register into the system by providing their credentials and payment information using the provided web or mobile app. It should also help the users in finding the locations of the car within a certain distance from the user's specific location.

The app should also be provided with the freedom of cancelling the registration if the users doesn't pick up the car in specified time.

Users may not always be registered in the app; they should also be able to see the available cars just by entering the app. This kind of users are considered as guest users.

The system included various kinds of services and benefits accordingly.

The main purpose of the system is to be more efficient and reliable than the current apps in the same domain which tries to offer better services to the client.

Goals

- [G1] Allowing the guest users to enter the app.
- [G2] Allowing the users to find all nearby available cars and their locations.
- [G3] Allowing the users to register into the application by providing their credentials and payment information.
- [G4] Allowing registered users to book car in a selected geographical area.
- [G5] Allowing registered users to reserve only a single car and make it available for up to one hour before they pick it up.
- [G6] Allowing admin users to keep track of the cars are current locations.
- [G7] Allowing admin users to send notifications or alerts to some cars accordingly.
- [G8] Allowing registered users to choose option in sharing other passengers.
- [G9] Allowing registered users to choose on money saving option by letting them enter their destination.
- [G10] Allowing registered users to check the percentage of battery left in the car.
- [G11] Allowing registered users to check the price charged till that moment of time.
- [G12] Allowing registered users to find the safe areas for parking the car after their use.
- [G13] Allowing registered users to send the information of their current location for the car to get unlocked and ready for the drive.
- [G14] Allowing registered users to receive a confirmation SMS or an email stating that the car has been booked, location of the car and threshold time to collect the car.
- [G15] Allowing registered users to know the estimated price of the ride and optimal statistics.
- [G16] Allowing registered users to show the nearby charging points.
- [G17] Allowing registered users to unlock the car whenever needed.
- [G18] Allowing registered users to stop the ride whenever needed.

Domain properties

Assuming the following properties to be stick and not broken.

- The payment information of the user is genuine and transactions are not to be failed.
- The GPS system of the car and the user should be proper and accurate.
- Car's GPS should not be turned off at any point of time.
- User doesn't get a damaged car.
- User should not be charged more or less than the calculated price.
- Car contains all required documents.
- User should carry all required legal documents for the drive.
- Car should not carry more than the limited number of passengers.
- Car should be unlocked only if the respective reserved user is close by.
- User should be given the locations of the available cars in the sorted of the distances from the user to the car.
- Guest users should be stored in the system; they should be removed immediately after their session.
- The car codes and user ids should be unique.
- Discounts are fixed and system should act accordingly.
- Penalties are also fixed and the users are to be bound to pay the penalties in case of situations.
- The devices of the users should meet minimum hardware requirements of the application.
- User should be in complete sense at the time of entering the car.
- User should maintain a minimum balance at the time of unlocking the car.

Glossary

- Guest user: he is one of the clients of the application. Each should attach the following information whenever he tries to access the application data.
    - Name
    - Phone number
    - GPS location
- Registered User: he is another client of the application who has access to most of the part of the app. He can reserve/ cancel the cars. He will be charged accordingly for the drive he had made with the reserved car. He should provide the following details to the application for better service.
    - GPS location
    - Payment details
    - Sharing mode
    - Time of reservation
    - Name
    - Phone number
    - Email ID
- GPS: Global Positioning System, it provides the exact location of the respective object.
- Electric Cars: those are the cars which work using the electric energy rather than the normal fuels. These are eco-friendly devices.
- Locking of the car: When the car is placed in the parking zone and if there are no people inside the car, the car moves to the locked mode.
- Unlocking of the car: When the user is close to the car, car receives the signal and unlocks the car so that user can use it for his drive.
- Car sharing: It is possible for the user to share his drive with other users who may join on the way of the main user's path.
- Ride: A tour when the car gets unlocked and user starts the drive till the cars gets locked when users leaves the car is called a ride.
- Discount: A portion of amount that has been deducted from the user's ride as a token of appreciation.

- Penalty: A portion of amount need to be extra paid by the user in case of violation of rules.
- User request: it is the request from the user to reserve a car close to his location for his drive.
- Reserving a car: it means the car has been blocked and no one else other than the respective user who reserved the car can use it. If the user fails to arrive the car with in an hour, the reservation will be cancelled and the car will be available open for reservation.
- Request queue: all the requests by the user should be processed in a first in first out approach without maintaining any priority. First request should be provided with the resource only then the next.
- Matching ride: If two persons (X and Y) are matching the ride if they meet the following conditions:
    - Starting point and ending point of Y are close to the starting point and ending point of X. And starting point of Y is in between start and end points of X.
    - There should be enough vacancy in the car.
    - Location of X should not overtake location of Y before matching.
- Wait down charge: This is the charge applied on the user whenever the user keeps the car on unlock mode and still the user gets charged for owning the car for that amount of time of unlock.

Text assumptions

Here are the few assumptions which we make on the application which is being built.

- User's device should transmit its location either using a Bluetooth or infrared channel which restricts the distance between the car and the user to be limited. This is when car will be unlocked.
- We need the information about the car, optimal locations where the car be parked by the user.
- All the discount prices are fixed and may not be changed.
- Penalties are also fixed in case of breaking any rules.
- Phone number verification for the guest user may be required to avoid unnecessary requests.
- We have fixed formula for the price calculator which takes distance time and peak hours into consideration.
- User may be notified with the list of entries of other users who are having their requests as per the matching a ride. It is up to the chose the option of sharing the drive.
- All the parking slots have recharging facilities and they don't run out of charge.
- We may have to restrict the users who don't break the rules more often.
- User will be provided a pin which is useful to unlock the car every time he gets out. Pin expires once he stops the ride.
- Car will be containing enough charge at the time of pickup.

Constraints

Regulatory policies

GPS locations are to be transferred to another entity in the system only with the permission of the user. Confidential data or sensitive data must be public and to be used only within the organization. Car should be driven only in the specific region, it is not allowed to drive the car beyond the region.

Hardware limitation

- Internet connection (at least 50 Kbps)
- GPS support
- Smart phone in case of mobiles
- Web app support in case of PC

Parallel operations

The server needs to make parallel operations to process the requests parallelly.

Proposed System

The application will be implemented using AngularJS which supports Single Page Applications and will be continued using a MVC pattern.

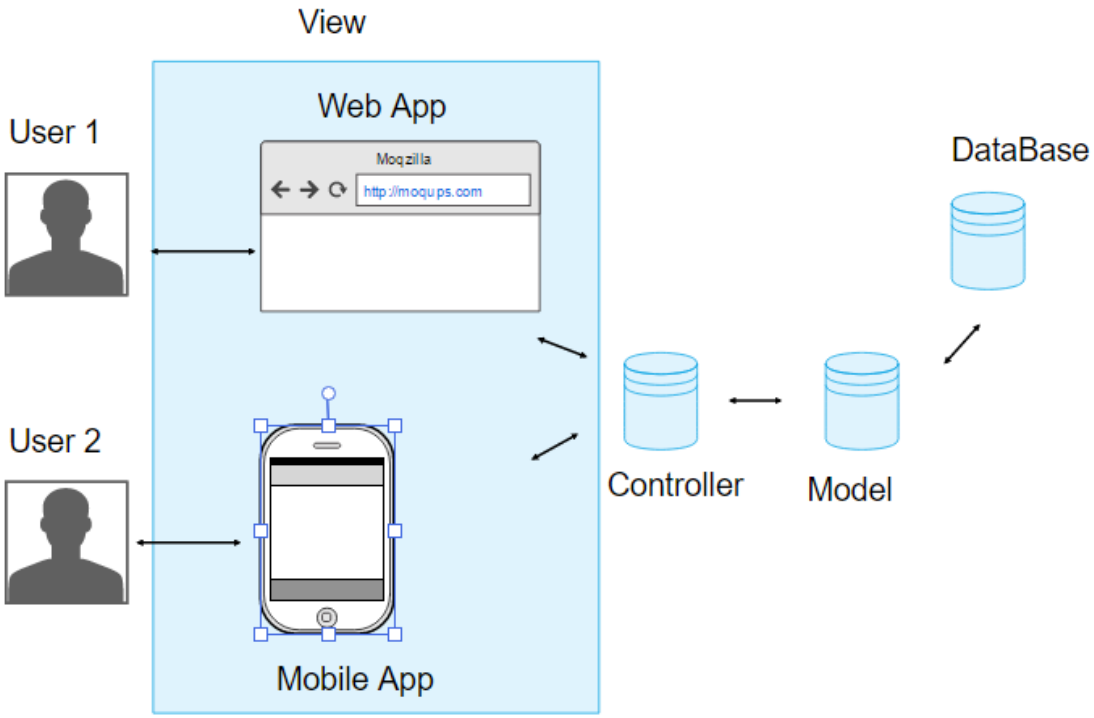

Figure 1: Architecture

Identifying stakeholders

There are internal and external stakeholder. All the staff, board members and volunteers of the organization come under internal stakeholders. The guests, registered users, car maintenance services fall under external stake holder category.

Reference documents

- Requirement Engineering part I
- Requirement Engineering part II
- Requirement Engineering part III
- Example docs: RASD sample from Oct. 20 lecture.pdf
- SRSExample-webapp.pdf
- http://www.slideshare.net/indrisrozas/example-requirements-specification

Actors identifying

There are four main actors in the system:

Guest user: He is not registered in the system. He can check all the nearby available cars.

Registered user: He is registered in the system. He can check nearby cars and can also reserve them.

Admin: He is the responsible person for the system. He has rights to add/ remove a user. He can also send notifications to the users and cars in case of situation.

Requirements

Functional requirements

With all the assumptions, domain properties we can achieve the goals listed in the above section of the document by deriving the following requirements. Each goal has its own set of requirement and they are mentioned below.

- [G1] Allowing the guest users to enter the app.
    - The system should take user name and phone number as input.
    - The system should let the guests enter only if they enter a valid phone number.
- [G2] Allowing the users to find all nearby available cars and their locations.
    - The system must request the user for his GPS location.
    - The system must use the GPS location to find the nearest car hubs where the user can find the car.
    - The system must provide the user with appropriate results.
- [G3] Allowing the users to register into the application by providing their credentials and payment information.
    - The system should take user name, password, and email id for registration.
    - The system should verify the email id by sending a confirmation mail.
    - The system should take the payment details when the email gets verified.
- [G4] Allowing registered users to book car in a selected geographical area.
    - With the proper login credentials the system should show the user with all nearby available cars.

    - The system should let the user to select the car chosen by the user if it is available.
    - The chosen car should be made available for this user and make it blocked for others.
- [G5] Allowing registered users to reserve only a single car and make it available for up to one hour before they pick it up.
    - The system should not allow user to book another car if there is already the user blocks one car.
    - The system should keep track of time of when the reservation was made.
    - If the difference between current time and the time of reservation is more than an hour the system must cancel the reservation.
    - In case of cancellation of the cars by the system or by the user, a penalty of 1 euro.
- [G6] Allowing admin users to keep track of the cars are current locations.
    - The system should keep track of all the information about every car in the data base.
    - In case of any request the tracked data should be shown to the admin.
- [G7] Allowing admin users to send notifications or alerts to some cars accordingly.
    - The system should make sure that admin is connected to every individual.
    - The system should take care of sending notifications or alerts to an individual by the admin.
- [G8] Allowing registered users to choose option in sharing other passengers.
    - When the user blocks a car, the system should ask the user if they are interested in sharing the ride.
    - If the user is interested, the car should be shown to others who are willing to have a shared ride.
    - In case of shared rides, the system should take user's discount of 10% into consideration and calculate the price accordingly.

- [G9] Allowing registered users to choose on money saving option by letting them enter their destination.
    - The system should keep track of all the information like the parking slots, more densely booked areas.
    - The system should use all the above stored information and calculate the optimal fare and show the user on how to proceed to the destination to achieve the optimal fare.
- [G10] Allowing registered users to check the percentage of battery left in the car.
    - The system should provide an interface to the user where the battery percentage of the car is shown whenever requested by the user.
- [G11] Allowing registered users to check the price charged till that moment of time.
    - The system must calculate the fare of the ride by using the ride statistics and should be shown whenever requested by the user.
    - The system must make sure that tracking the data of the ride should never be interrupted through the ride.
- [G12] Allowing registered users to find the safe areas for parking the car after their use.
    - The system must provide the proper parking slots which is advantageous for both user and the car.
    - The system should also let the user know about special parking areas and the which he can claim a discount of around 30% on his last ride if he parks the car in one of the special parking areas.
- [G13] Allowing registered users to send the information of their current location for the car to get unlocked and ready for the drive.
    - The system should keep listening to the user's location ping and make sure that the car gets unlocked by the time of the user's arrival.
- [G14] Allowing registered users to receive a confirmation SMS or an email stating that the car has been booked, location of the car and threshold time to collect the car.

    - The system should send a confirmation message to the user in any form either as a text message or a mail stating that the car has been booked for him and the details of the care should be provided.
- [G15] Allowing registered users to know the estimated price of the ride and optimal statistics.
    - The system should be transparent with the users by letting him know the estimated price of the ride whenever requested by the user.
    - The system should also provide the user the optimal statistics on how he can reduce the fare, how he can claim the discount and all such related things.
- [G16] Allowing registered users to show the nearby charging points.
    - The system should help the user in knowing nearby charging points so that user can park the car or use the charging point near that location.
    - This also helps the user to claim for the discount of 20% on his last ride if the car is left with no more than 50% of the battery.
- [G17] Allowing registered users to unlock the car whenever needed.
    - The system should allow the user to unlock the car and allow him to do his work, then use the car again.
    - During the time of unlock, the system should not be shut down and the price should be kept on calculating. User should be informed the wait down charge.
- [G18] Allowing registered users to stop the ride whenever needed.
    - The system should provide the user with the freedom to stop the ride when the user wants to or when the destination has arrived.
    - The final price should be deducted from the user's card and notify him the amount.
    - The system should also consider the discounts applied for the ride and update the user's profile accordingly.
    - The system should also detect the nearest power grid station from the point of the current location and the current percentage of battery. If the nearest power grid station is

farther than 3km and if the battery is less than 80%, the user should be charged with a penalty of extra 30% on his ride.

- The system should make sure that the car is available for the new ride and should be shown in the available list of the other users.

Nonfunctional requirements

User Interface

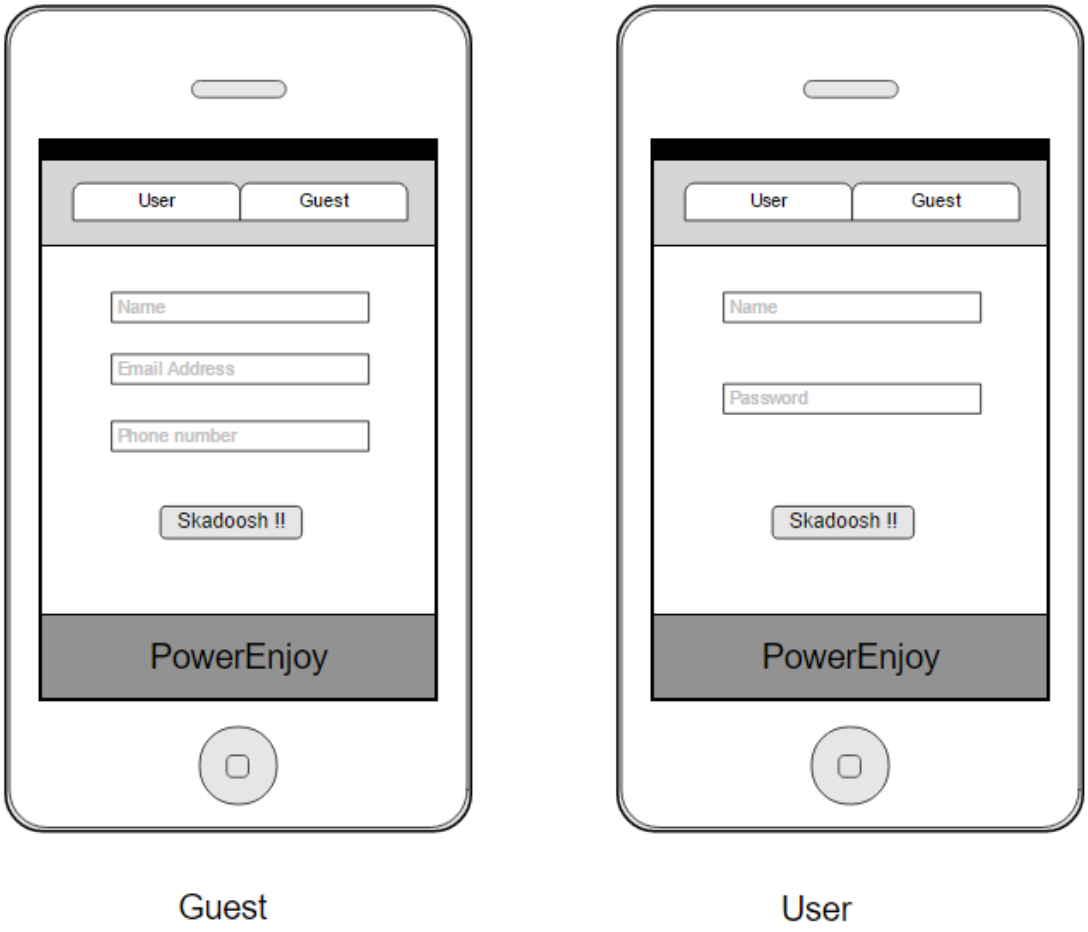

Figure 2: Mobile login pages

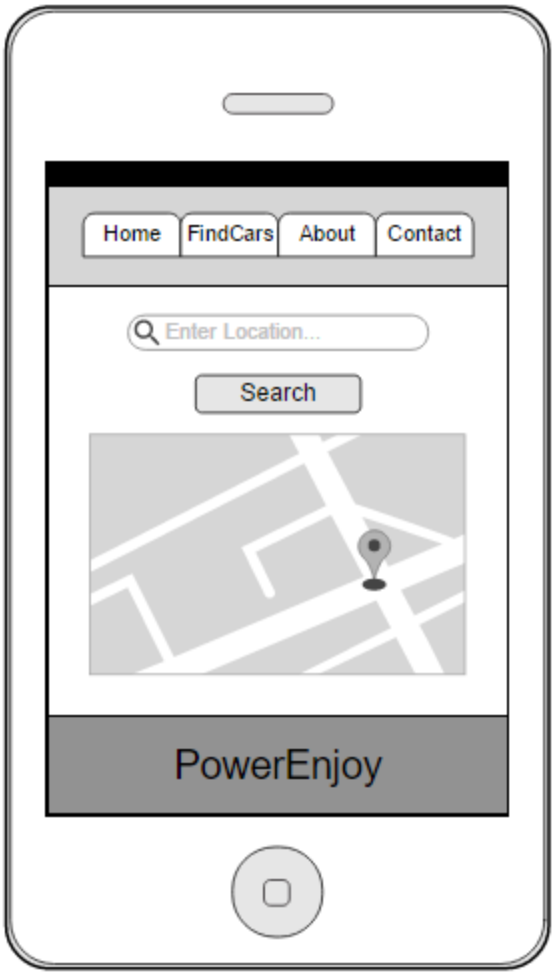

Figure 3: Mobile search page

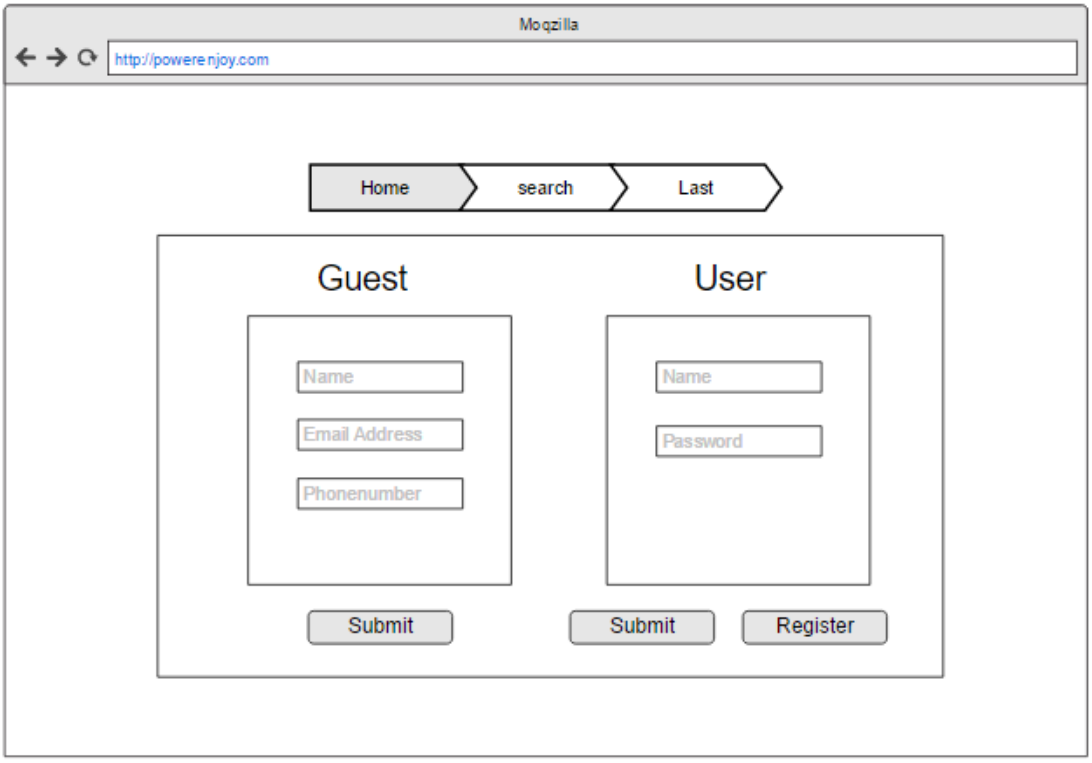

Figure 4: Desktop user/ guest login page

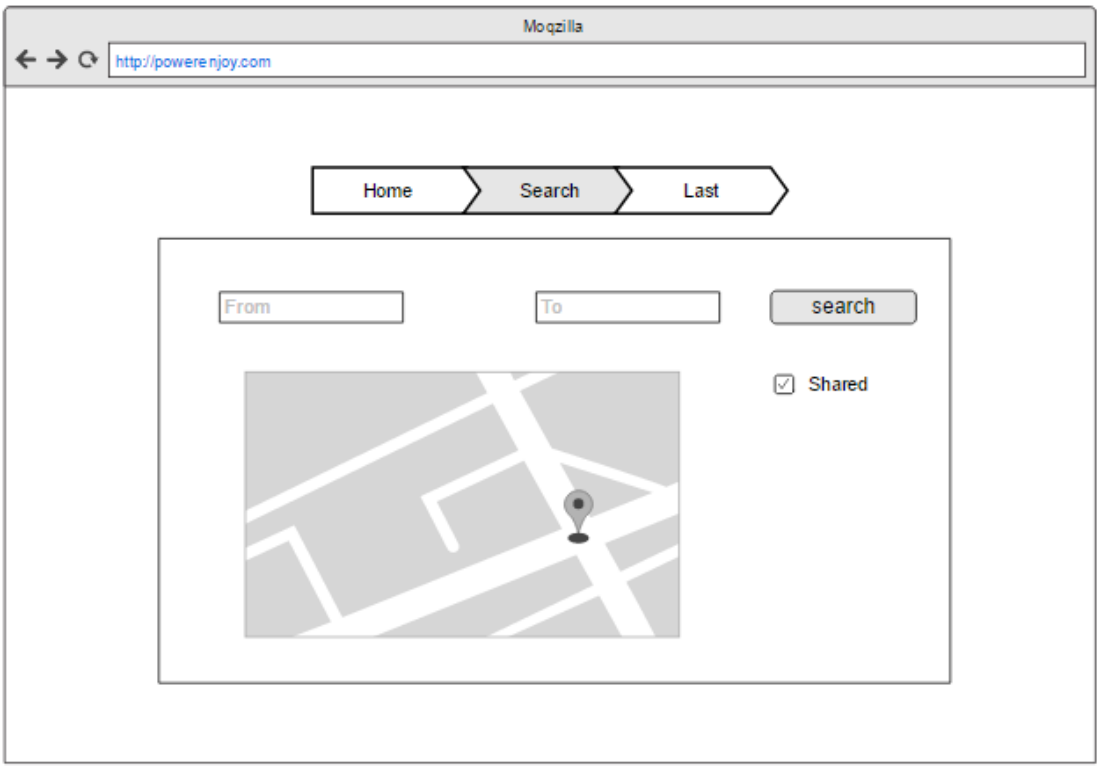

Figure 5: Desktop location car search page

Documentation

All the documents to well organize the work in the way to do in a fewer time the best way as possible will be drafted properly:

- RASD: Requirement Analysis and Specification Document, to well-understand the given problem and to analyze in a detailed way which our goals are and how to reach them defining requirements and specification.
- DD: Design Document, to define the real structure of our web application and its tiers. project.

Architectural Consideration

We will use the following technologies:

- Apache with php (with CodeIgnitor framework) as API server and task service.
- OracleDB as sql server to store data persistently, it is the same of the old system.

- Apache server for static documents.
- NodeJS will be used to take care of the synchronous part in the system.
- JSON for API communication over HTTP(S).
- Javascript (with AngularJs framework), CSS and HTML to create responsive site that communicate to server.
- Modern browser with javascript and ajax support
- Java and swift respectively for android and iOS apps, using original SDK.
- Internet connection for communication of data.
- External rest APIs to send SMS.

Scenario Identifying:

Here are some possible scenarios of the usage of this application.

Scenario 1:

Giorgia has an urgency to go to a place which is only accessible through car within the time. She enters as a guest user into the app to search for nearby cars and finds one. She registers in the application by providing the credentials and payment information for booking the car. Finally, she receives a SMS confirmation stating that the car has been booked thereby providing location of the car and threshold time to collect the car. She reaches the location and spots the car bay as per the information provided by the application.

Scenario 2:

Giorgia reaches the car bay to take the car. She founds an unlocked car in the bay which signals through the indicator. She goes to that car and enters inside to drive it. Once she reached the destination, she gets out of the car and the car gets locked automatically.

Scenario 3:

Giorgia drives the car in a direction which is out of the area limit set by the admin users. She receives alert from the GPS regarding the off-limit drive, which she disregards it. Eventually the car gets switched off within few minutes after the alert message. As per rules set by the admin users, she gets out of the car and enters the app in which she notifies the car lockdown. She receives the notification from the admin to enter the premises within the duration and the car gets switched on. Finally, she enters the limit and take a different route to reach the destination.

Scenario 4:

Giorgia reserves a car and takes it for a ride. He also chooses the option to share his ride with someone. And luckily, he finds 2 other users who needs to travel in the same direction as him. He picks him up and drops him at the destination. Since he shared the ride, per the discounts provided by the application, he got an off 10% on his last ride. He also parked the car in the special parking zone with the car plugged in with the power grid, which made him also claim another 30% of discount. In this way Giorgia had cut down the cost which helped both him and the admin.

Scenario 5:

Giorgia reserves a car and takes it for a ride. He is such a bad user. He left the car 3km away from the nearest power grid station and left the car with less than 20% battery left. The system immediately detects the misconduct of the him and charges him extra 30% on his previous ride.

UML Models:

Use Case Diagram

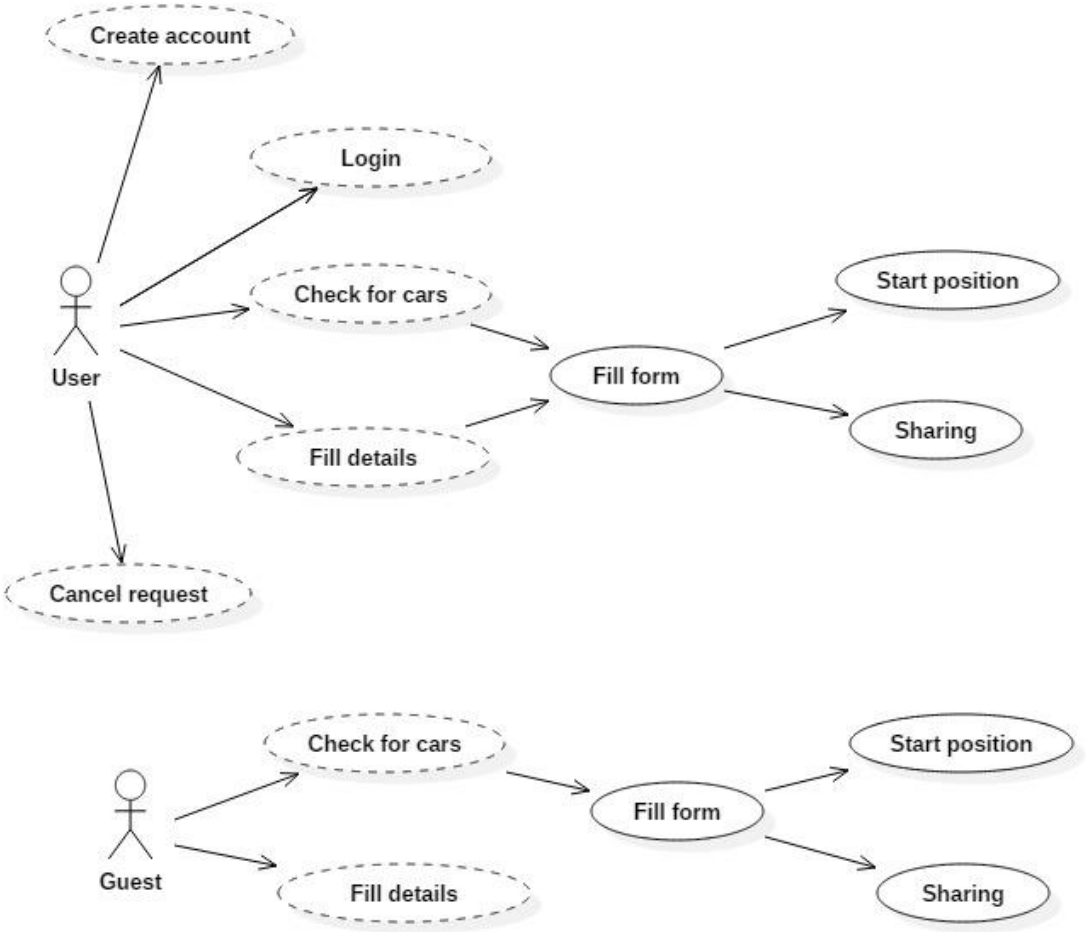

Figure 6: Use case diagram

Use Case Description

User logs in

Name: User log in

Actors: User

Entry Conditions: There are no entry conditions.

Flow of events:

- User enters the login page in the application.
- User inputs his user id and password.
- User clicks on the submit button.
- The system redirects the user to the dashboard page.

Exit conditions: The user successfully redirects to the dashboard.

Exceptions: The user ID and the password provided by the user are not correct. In this case, the system does not redirect the user to his dashboard but notifies him that an error has been made and allows to input his user ID and password again.

User checking for cars

Name: User checking for cars

Actors: User

Entry Conditions: User needs to log in successfully.

Flow of events:

- User enters to his personal dashboard.
- User clicks on the search button.
- The system switches on the GPS of the user's device.
- The system displays all nearby cars which he can rent.

Exit conditions: The system displaying all nearby cars.

Exceptions: There are no exceptions.

User cancelling a reservation

Name: User cancelling a reservation

Actors: User

Entry Conditions: The user must reserve a car.

Flow of events:

- User reserves a car and enters his personal dashboard.
- User selects the car which he reserved.
- User clicks on the car and request for cancellation.
- The system cancels the car and charges the user a penalty.

Exit conditions: The system cancels the car and charges the user a penalty.

Exceptions: If the user does not book a car, this will never be done.

Guest checks for the car

Name: Guest checks for the car.

Actors: Guest

Entry Conditions: There are no entry conditions.

Flow of events:

- Guest enters the home page in the application.
- Guest chooses the guest mode and request for the nearby cars.
- Guest inputs his email id and phone.
- Guest clicks on the submit button.
- The system switches on the GPS of the guest's device.
- The system redirects the guest to the dashboard page and displays all nearby cars.

Exit conditions: The guest successfully redirects to the dashboard and sees all nearby cars available.

Exceptions: The email ID or the phone number provided by the guest are invalid. In this case, the system does not redirect the guest to the dashboard but notifies him that an error has been made and allows to input his email ID and phone number again.

Class Diagram

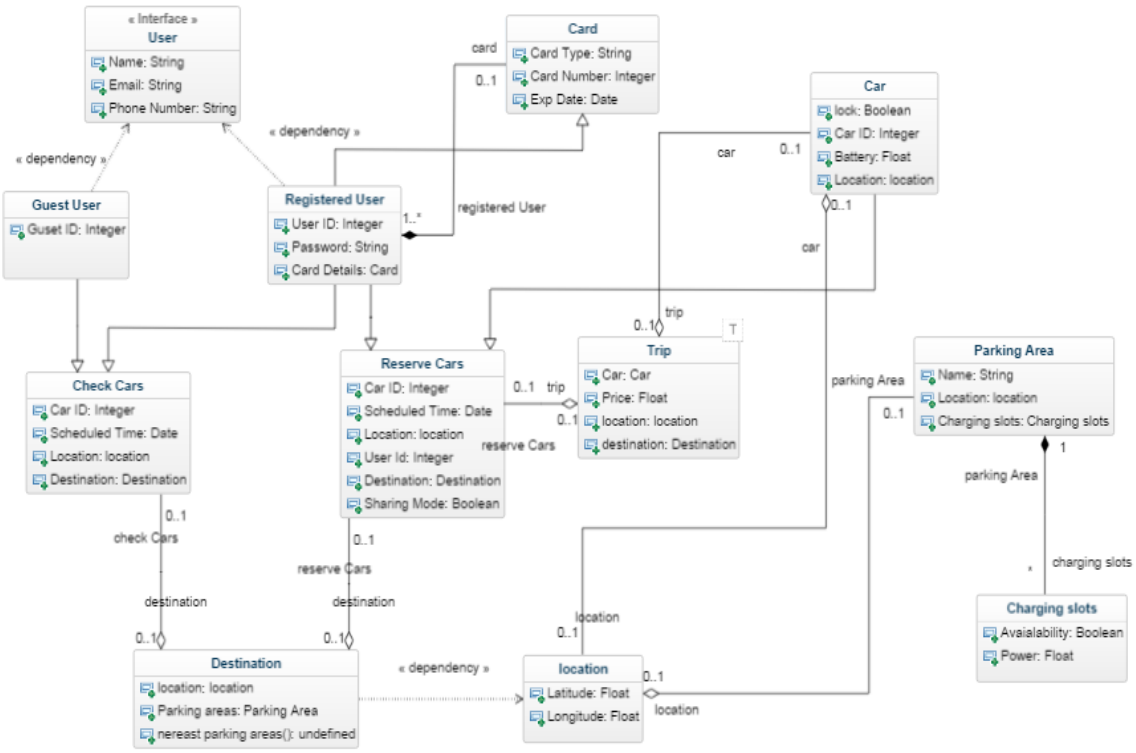

Figure 7: Class Diagram

Sequence Diagram

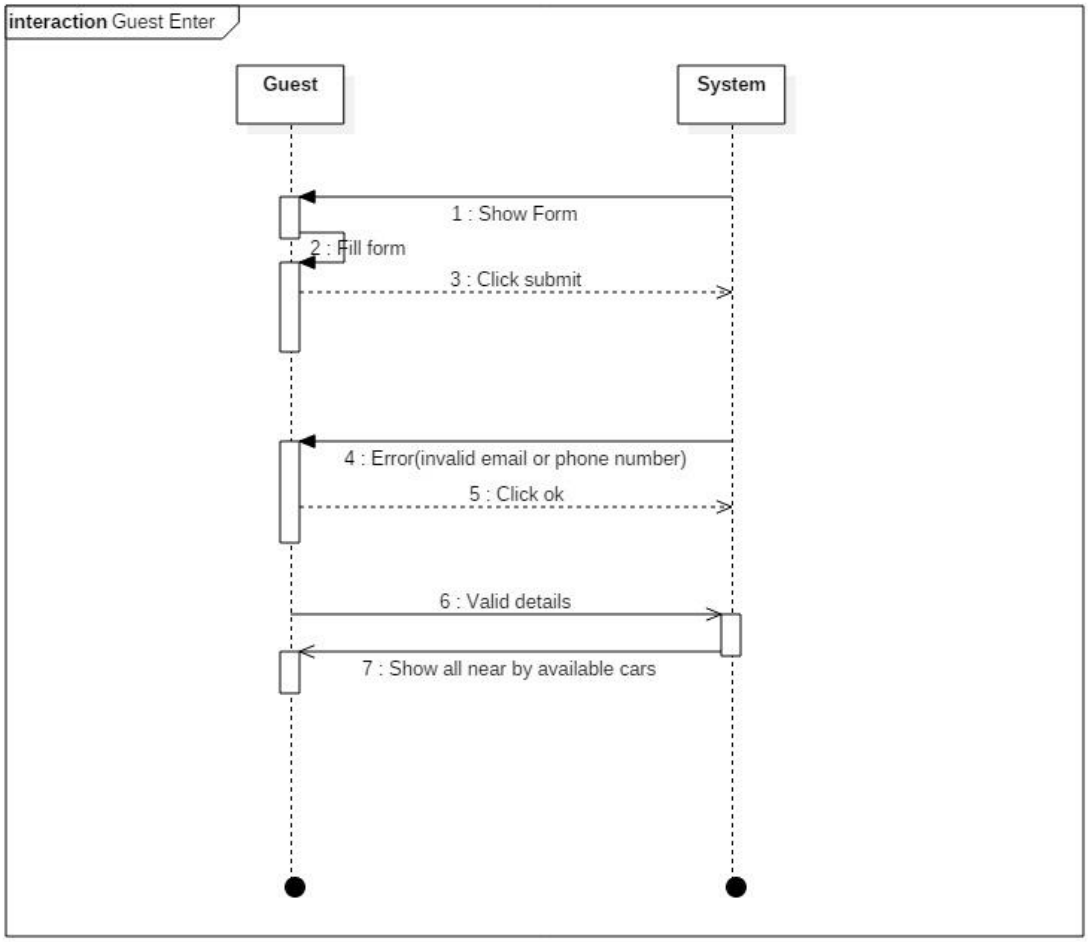

Figure 8: Guest Enter Sequence diagram

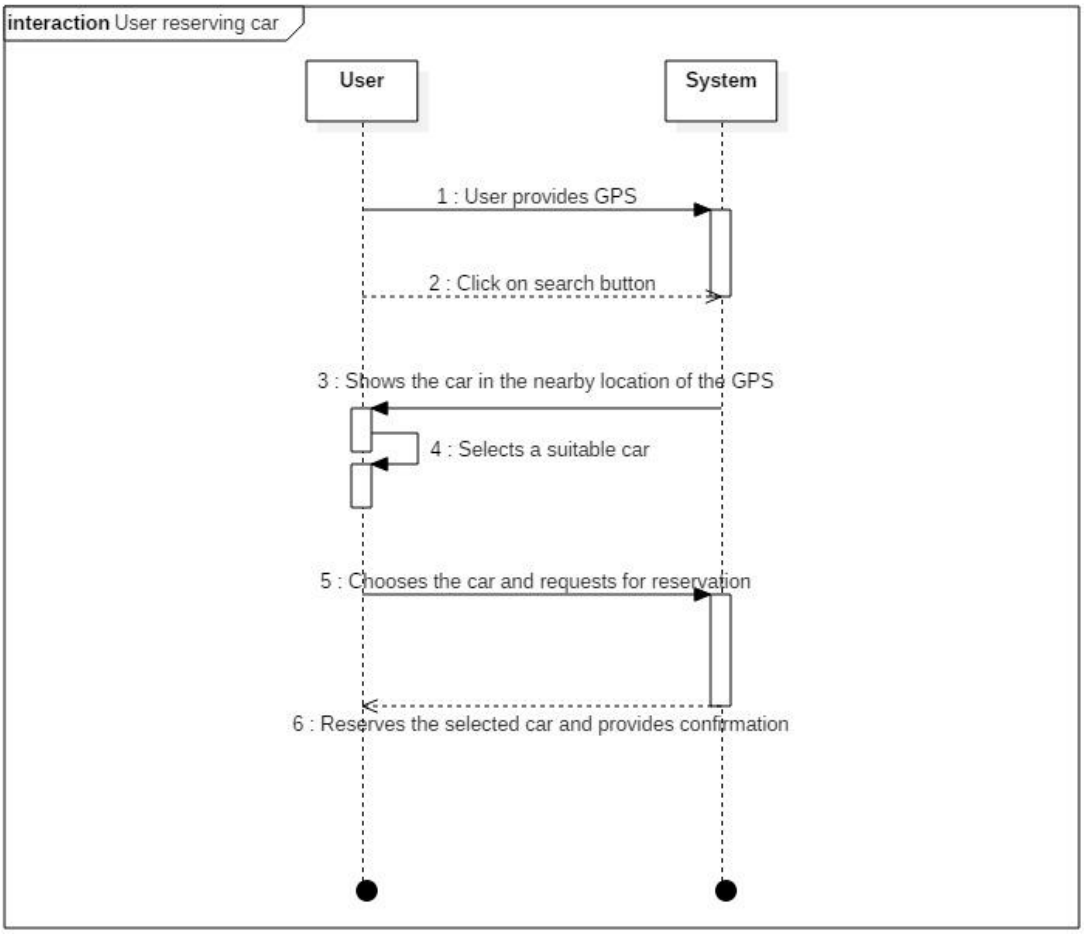

Figure 9: User reserving a car sequence diagram

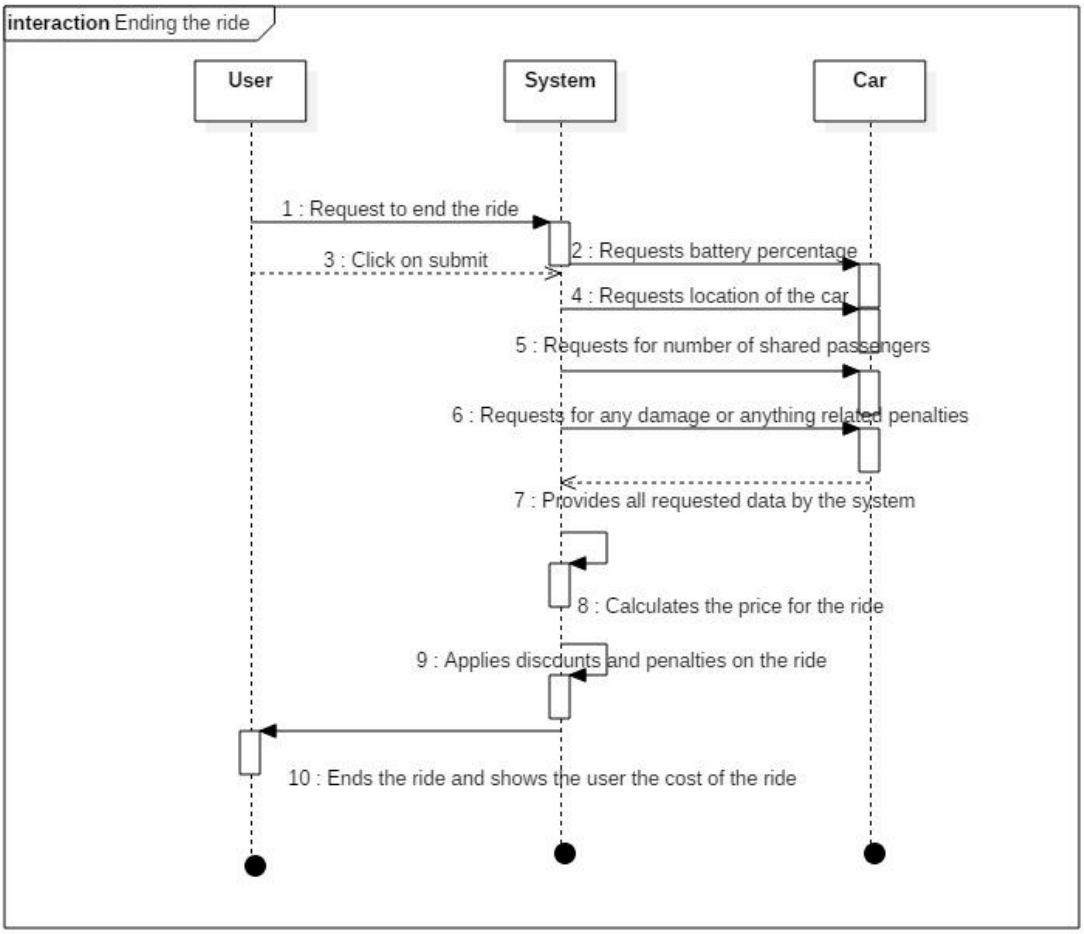

Figure 10: User ending the ride sequence diagram

Activity Diagram

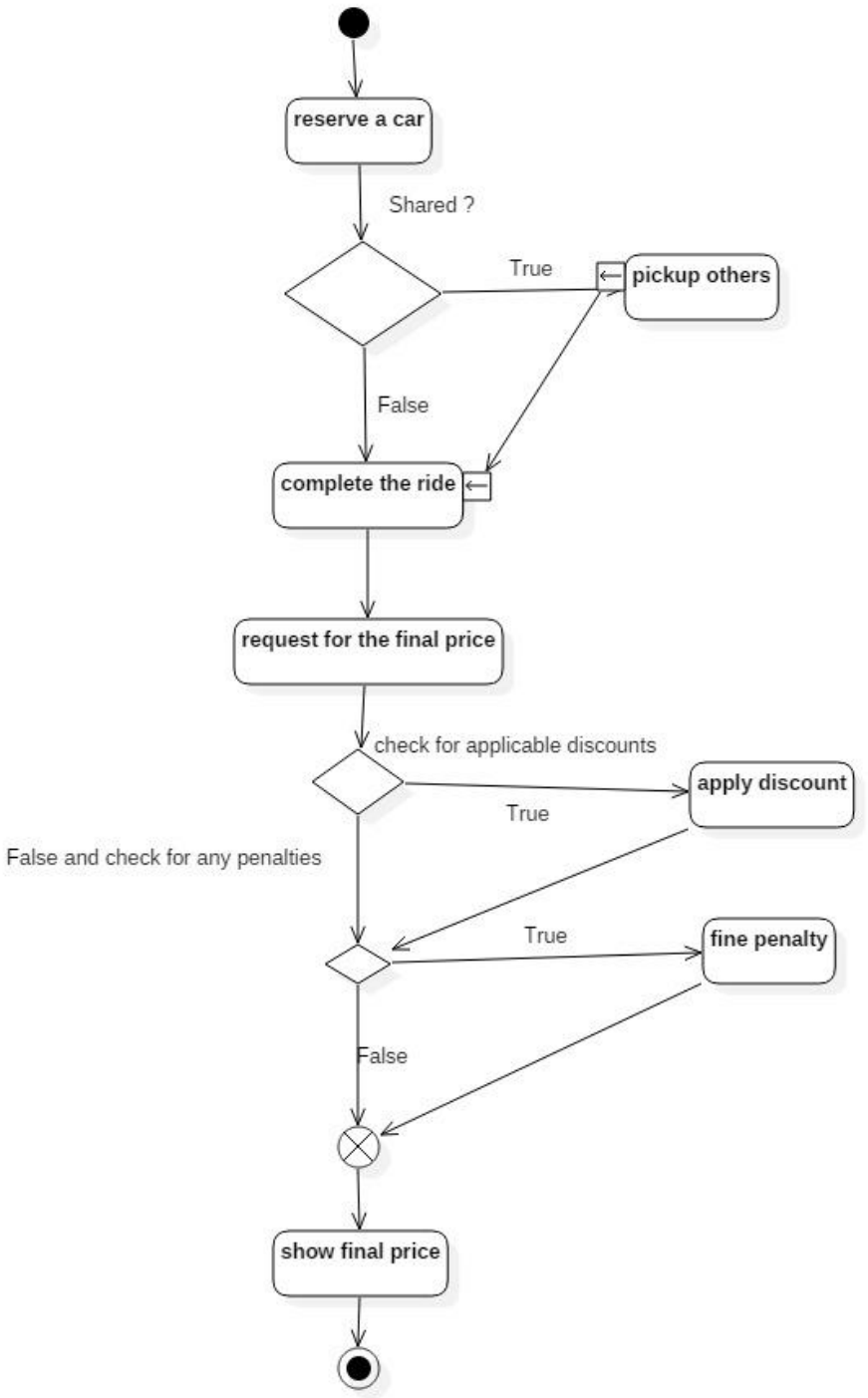

Figure 11: Activity Diagram

State Diagram

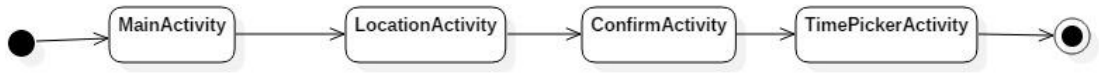

Figure 12: State Diagram

Alloy modelling

Model

```
sig Position{
   latitude : Int,
   longitude : Int
}

sig phoneNumber{}

abstract sig Bool{}
one sig True extends Bool{}
one sig False extends Bool{}

sig User{
   currentPosition : Position,
   reservedCar : lone Car,
   currentCar : lone Car,
   phonenumber : phoneNumber,
   sharing : one Bool
}

sig Car{
   currentPosition : Position,
   code : Int,
```

```
    seats : Int,
    lock : one Bool
}{
    code > 0
    seats > 0
    seats <= 4
}

fact CarIsUnlockedWhileUse{
    all c: Car | CarInUse[c] implies c.lock = False
}



fact carNotFilledByTwoPersons{
    no disjoint u, u' : User | u.currentCar = u'.currentCar
}

fact PhoneNumbersAreUnique {
    no disjoint p, p': User | (p != p') => p.phonenumber != p'.phonenumber
}

fact carNotReservedByTwoPeopleUnlessShared{
    no disjoint u, u' : User | ( u = u' and u.reservedCar =  u'.reservedCar) or
(u.reservedCar =  u'.reservedCar and u.sharing = True and u'.sharing=True)
}
```

```
fact userMustReserveAndUse{
   no u : User | #u.reservedCar = 1 and #u.currentCar = 1
}

fact CarCodesAreUnique {
   no disjoint c, c': Car | (c != c') => c.code != c'.code
}

fact carNotShownWhileUsed{
   all c: Car | CarInUse[c] implies !CarReserved[c]
}

fact CarReservedWithProperSeats{
   all c : Car | CarReserved[c] implies c.seats<=4
}

fact CarReservedNotInUse{
   all c : Car | CarReserved[c] implies !CarInUse[c]
}

pred CarInUse [c : Car]{
   one u : User | u.currentCar = c
}
```

```
pred CarReserved[c: Car]{
   one u : User | u.reservedCar = c and c.lock = False
}

pred reserveCar [u : User, c : Car,  u' : User]{
   u.reservedCar = none and u.currentCar = none and
   !(CarReserved[c]) and !(CarInUse[c])
   u'.currentPosition = u.currentPosition and u'.currentCar = none and
u'.reservedCar = c
}

pred useCar[u, u' : User, c : Car] {
   u.currentCar = none and !(CarInUse[c]) and
   (u.reservedCar = c or u.reservedCar = none)
   (u.reservedCar = c implies u'.reservedCar = none) and
   u'.currentPosition = u.currentPosition and u'.currentCar = c
}

pred show{}

run CarInUse

run CarReserved

run reserveCar
```

run useCar

Alloy result

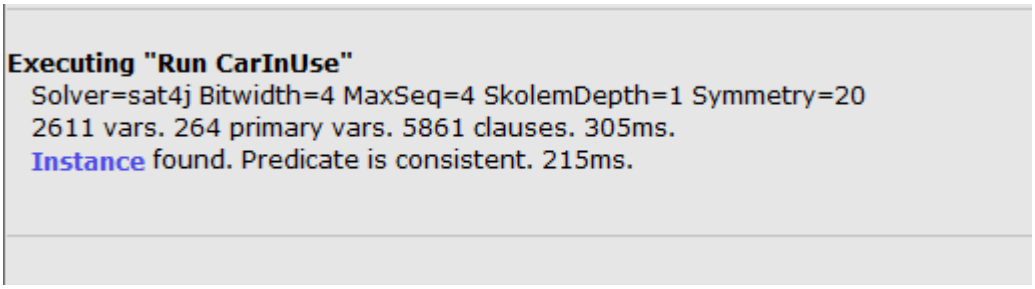

Figure 13: Alloy Result

MetaModel generated

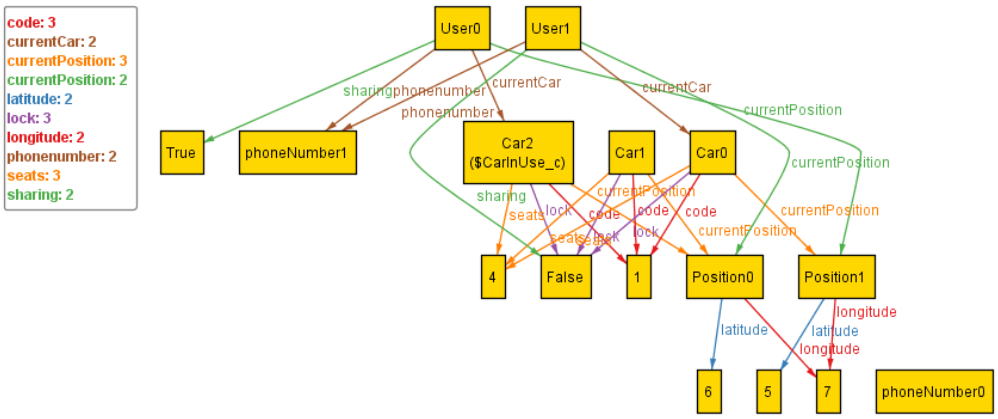

Figure 14: Alloy MetaModel

Used Tools

The tools used to create the above RSAD document are:

- Github: version controller for the project
- Google docs: To share the docs among the team members
- Moqups: for online mock up diagrams
- Word: For the local edits of the document and create pdf
- Alloy JAVA applet: To run and compile alloy code and create metamodel of alloy
- Star UML: To create UML diagrams
- Genmymodel: To create UML diagrams online

Hours of Work

Prasanth R: 37 hrs

Fathima B:

Lipika L:

Change log

- V 1.1
  - Added all the written part.
  - Made basic structure of the document.
- V 1.2
  - Modified few goals.
  - Added few new goals.
- V 1.3
  - Added alloy diagrams.
  - Some minor fixes in the goals.
- V 1.4
  - Added UML diagrams.
- V 1.5
  - Cleaned up the document and made it look perfect.