

# MyTaxiService - RASD

A series of horizontal lines in teal and light blue colors, with some lines having a 3D effect, spanning the width of the slide.

Andrijana Mirchevska (838622)

Marija Mavcheva (838647)

# Overview

- **Introduction**

- Purpose
- Scope
- Glossary

- **Overall Description**

- Identifying stakeholders and actors
- User characteristics
- Goals and assumptions
- Domain properties

- **Requirements**

- Functional requirements
- Non-functional requirements

- **Scenarios**

- **UML Models**

- Use Case Diagrams
- Class Diagram
- Sequence Diagrams
- State Chart Diagrams

- **Alloy Modeling**

- Alloy Code
- Alloy Worlds

# Scope

- The project is about developing an application that will enable fast and optimized taxi services in the city
- The application will allow users to register and then sign in into the app for using its services
- Also taxi drivers can register and sign into the myTaxiService application with the purpose to manage their availability and duties

# Identifying stakeholders

- ***Company*** - requires project specification and expects it to be delivered while respecting the set deadlines and budget
- ***Developer group*** - in this case group of two people
- ***Taxi driver*** - worker at the company that ordered the software product
- ***Passenger*** - person who need a ride to specific location

# Actors identifying

- **Guest** – can access the system but has never registered or still hasn't logged in
- **User** – already registered and logged into the system. The user can access and manage with all the services that the application offers
- **Taxi Driver** - already registered and logged into the system. The driver can access and manage with all the services that the taxi driver application offers
- **Admin** - is the person that is responsible for handling reports on users and drivers of my taxi service

# Goals

myTaxyService should have these features:

- registering a new user
- sending notification about taxi availability
- confirming about the reserved vehicle, its code and waiting time
- managing user profile
- ban user/driver

for each **user** it should provide:

- logging in to his/her profile
- making reservation
- requesting taxi
- canceling reservation
- reporting driver

for each **taxi driver** it should provide:

- logging in to his/her profile
- confirming/declining a request for taxi call
- reporting user

# Domain properties

- user making reservation from a specific location to a specific destination
- taxis are organized into taxi zones
- the payment process is irrelevant to the system, it`s done between the passenger and driver

# Assumptions

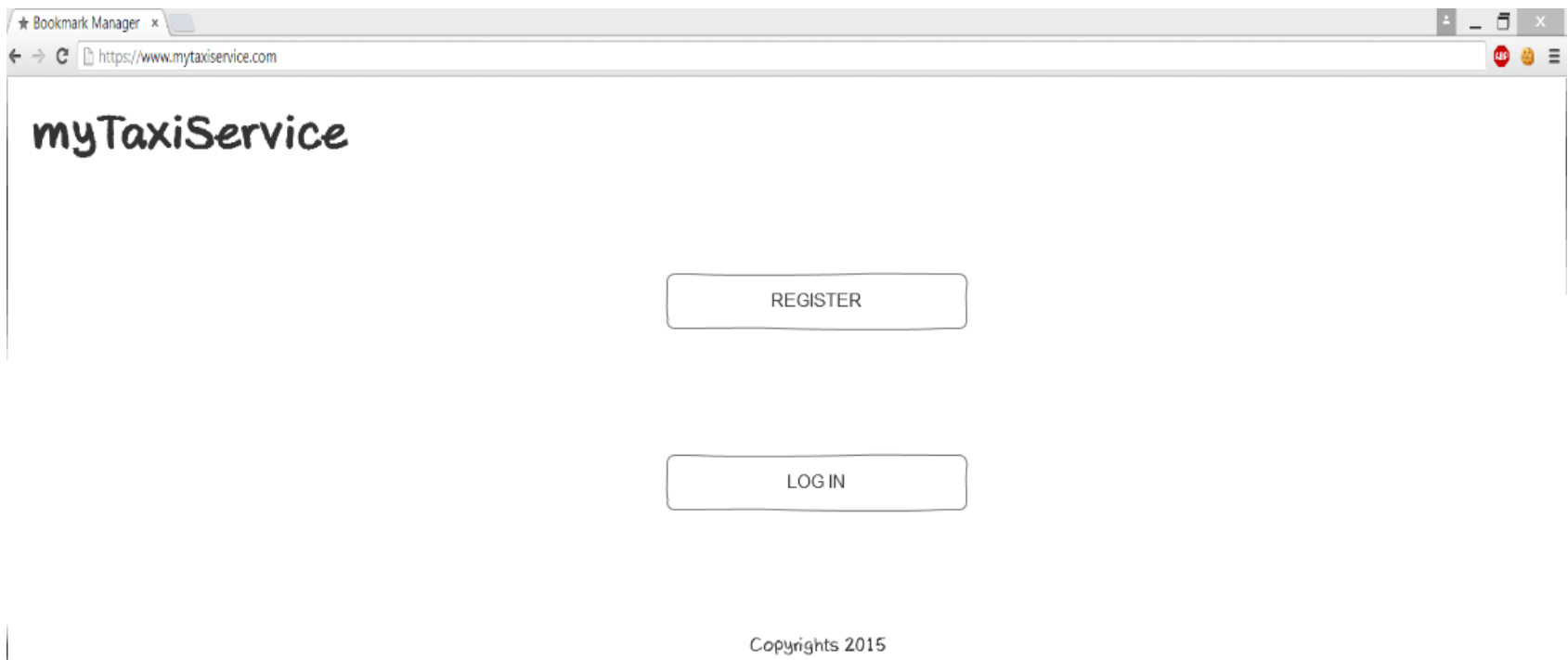
Facts that resolve ambiguities:

- user registers with email and password, and can change them
- user can only have one account
- there is a Terms & Conditions
- we assume that Google Maps service will calculate location accurately
- if the taxi driver does not respect the waiting time, he is banned from the system
- if there are any irregularities with the taxi driver, new vehicle is sent to the passenger



# User-interface

- Sketches for the web version of the application



Initial page

# User-interface



A screenshot of a web browser window displaying the registration page for 'myTaxiService'. The browser's address bar shows the URL 'https://www.mytaxiservice.com'. The page has a simple, clean design with a white background and black text. The title 'myTaxiService' is prominently displayed at the top. Below it, a prompt asks the user to fill in their details. There are five input fields: 'Name', 'Surname', 'Email', 'Password', and 'Repeat password'. Each field is represented by a text label followed by a rectangular input box. At the bottom of the form is a 'REGISTER' button. The footer of the page contains the text 'Copyrights 2015'.

★ Bookmark Manager x

← → C https://www.mytaxiservice.com

## myTaxiService

Please fill the following fields

Name

Surname

Email

Password

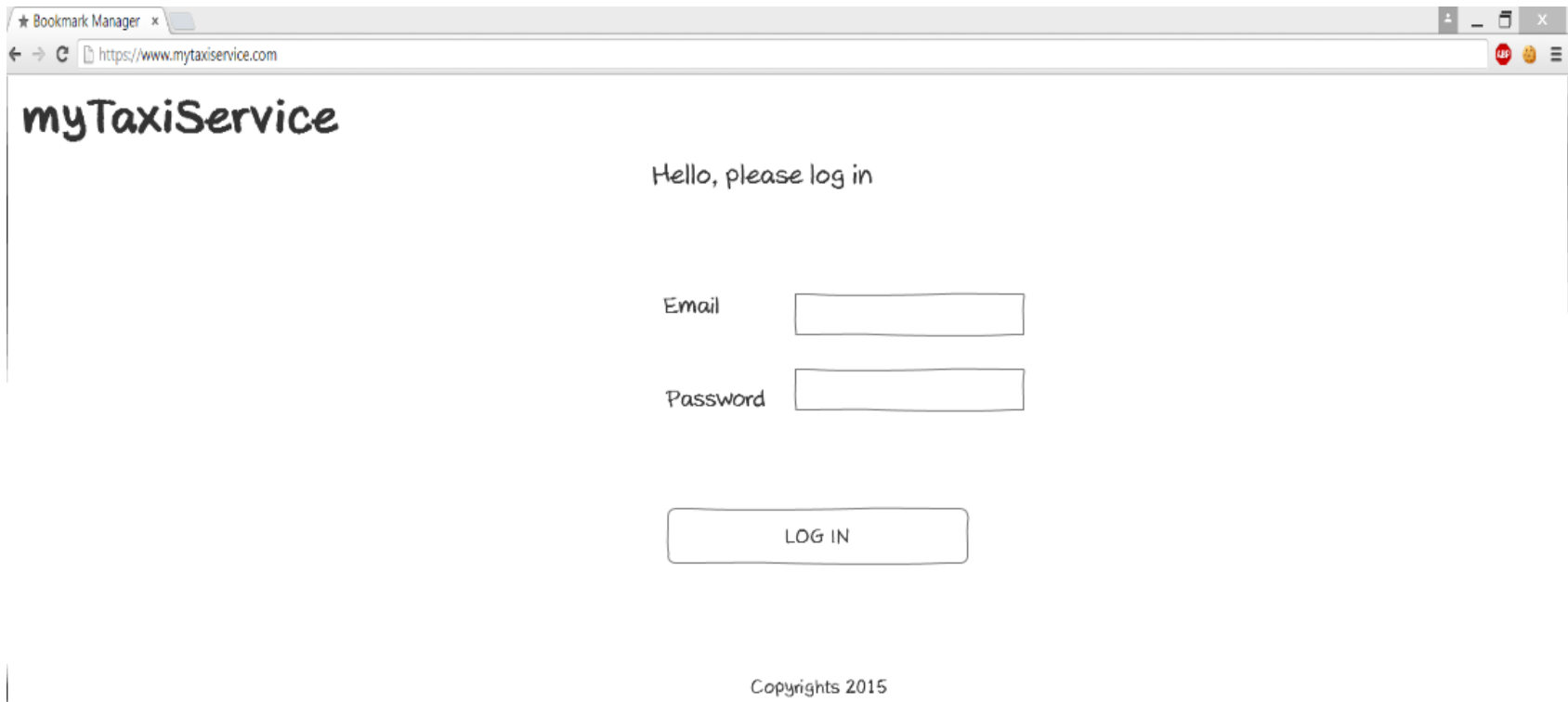
Repeat password

REGISTER

Copyrights 2015

Register page

# User-interface



A screenshot of a web browser window displaying the login page for "myTaxiService". The browser's address bar shows the URL "https://www.mytaxiservice.com". The page content includes the site name "myTaxiService" in a large, bold, black font. Below it, the text "Hello, please log in" is displayed. There are two input fields: one for "Email" and one for "Password", each with a corresponding label to its left. Below these fields is a "LOG IN" button. At the bottom right of the page, the text "Copyrights 2015" is visible.

★ Bookmark Manager x

← → <https://www.mytaxiservice.com>

**myTaxiService**

Hello, please log in

Email

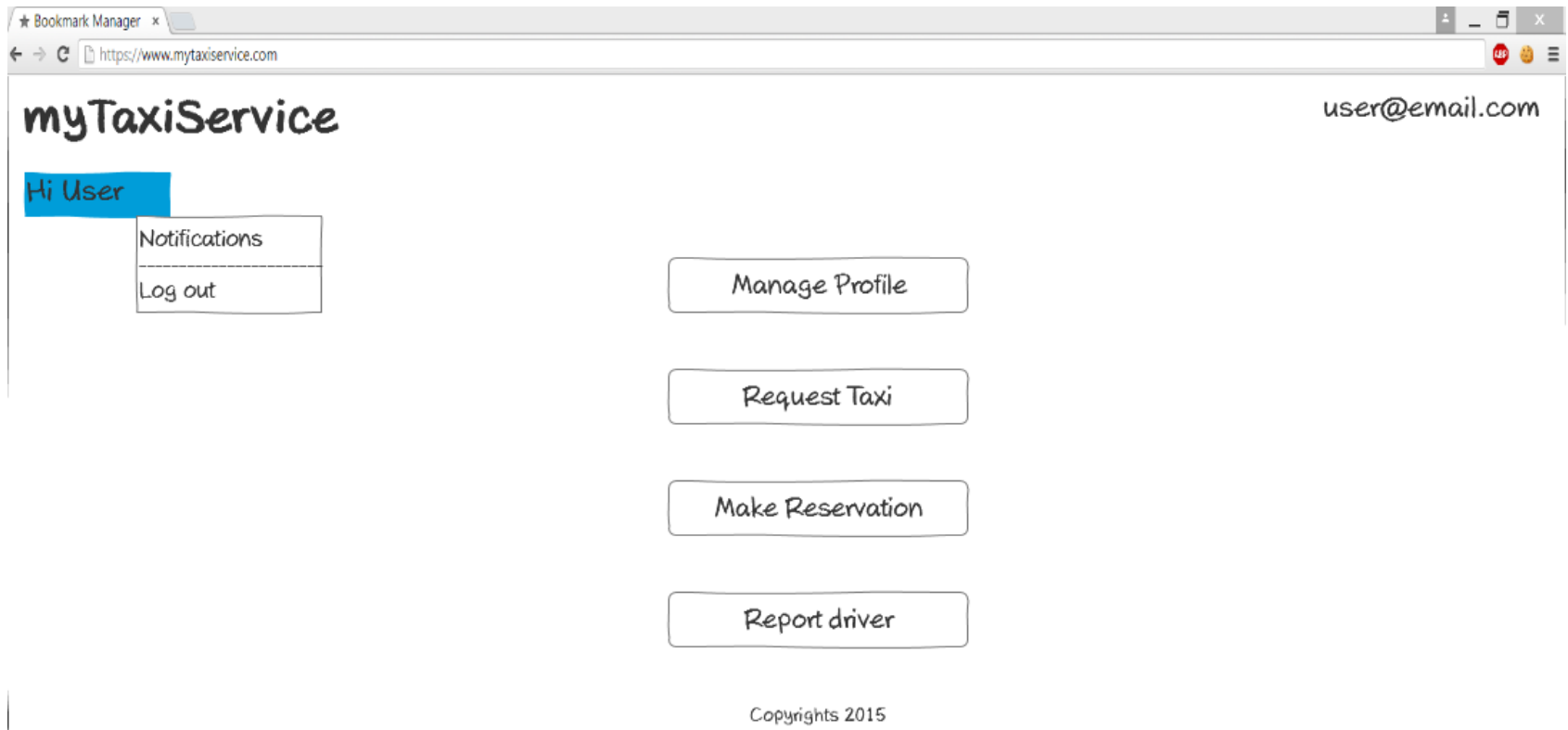
Password

LOG IN

Copyrights 2015

Login page

# User-interface



Homepage

# User-interface

The screenshot shows a web browser window with the address bar displaying "https://www.mytaxiservice.com". The page title is "myTaxiService" and the user is logged in as "user@email.com". The main heading is "Manage Profile". Below this, there are four input fields: "New Email", "New Password", "Repeat New Password", and "Choose Avatar". At the bottom, there are two buttons: "SAVE CHANGES" and "CANCEL". The footer of the page indicates "Copyrights 2015".

★ Bookmark Manager x

← → ↻ https://www.mytaxiservice.com

myTaxiService

user@email.com

Manage Profile

New Email

New Password

Repeat New Password

Choose Avatar

SAVE CHANGES CANCEL

Copyrights 2015

Manage profile page

# User-interface



The screenshot shows a web browser window with the address bar displaying "https://www.mytaxiservice.com". The page title is "myTaxiService" and the user is logged in as "user@email.com". The main heading is "Reservation". Below this, there are four input fields: "Select" with a dropdown menu showing "Zone", "Location" with a dropdown menu showing "Via Bonardi 23", "Destination" with a dropdown menu showing "Duomo", and "Time" with two dropdown menus showing "20 h" and "15 min". At the bottom, there are two buttons: "MAKE RESERVATION" and "CANCEL".

myTaxiService user@email.com

Reservation

Select Zone

Location Via Bonardi 23

Destination Duomo

Time 20 h 15 min

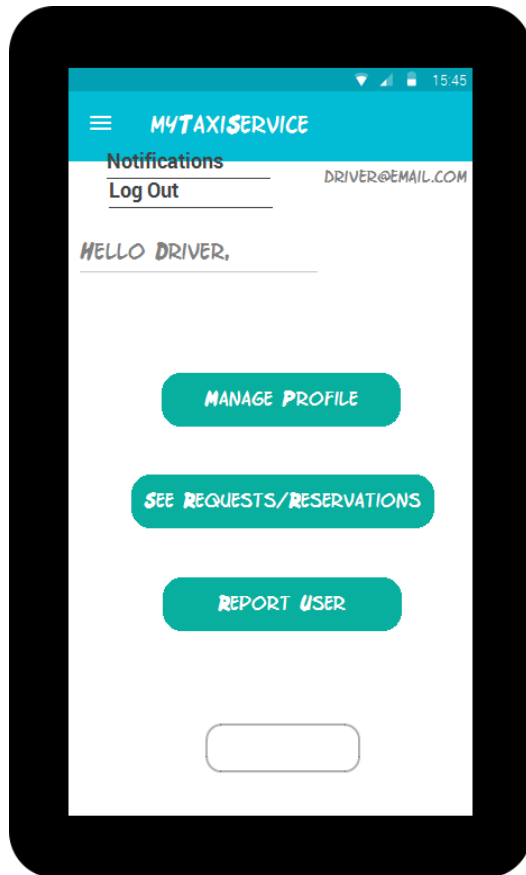
MAKE RESERVATION CANCEL

Copyrights 2015

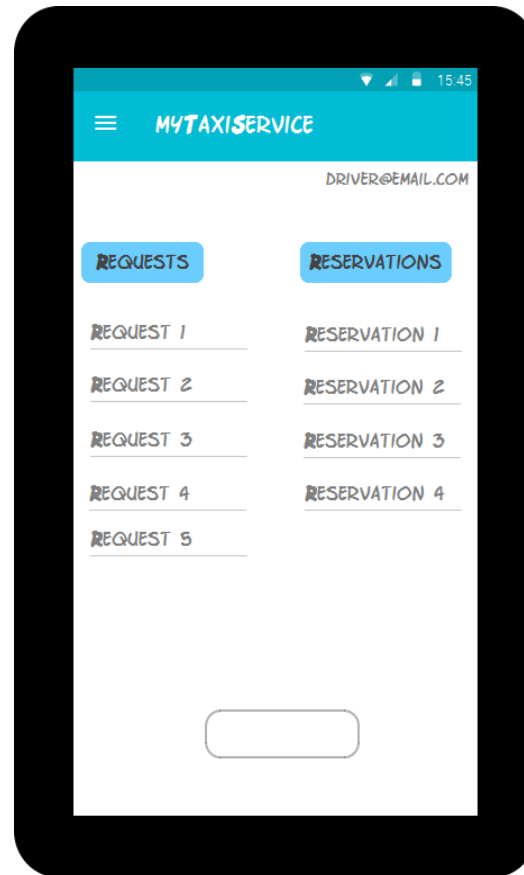
Reservation taxi page

# User-interface

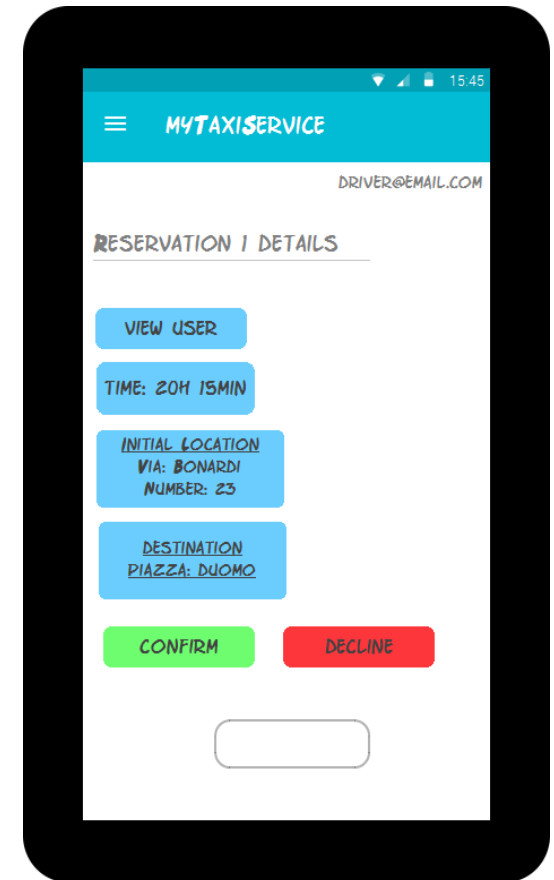
- Sketches for the mobile version of the application



Driver homepage



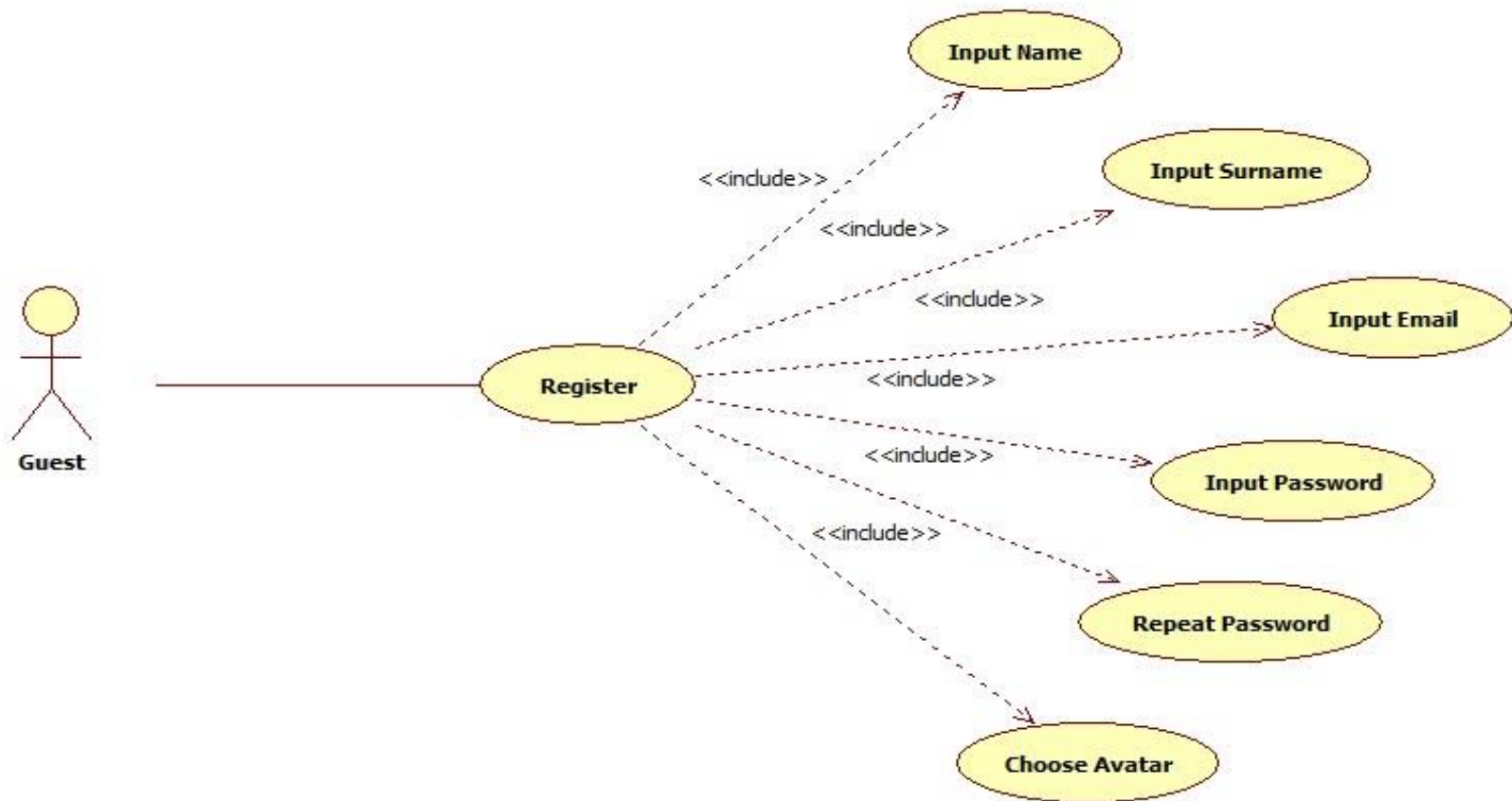
Notification page



Reservation details page

# Use Case Diagrams

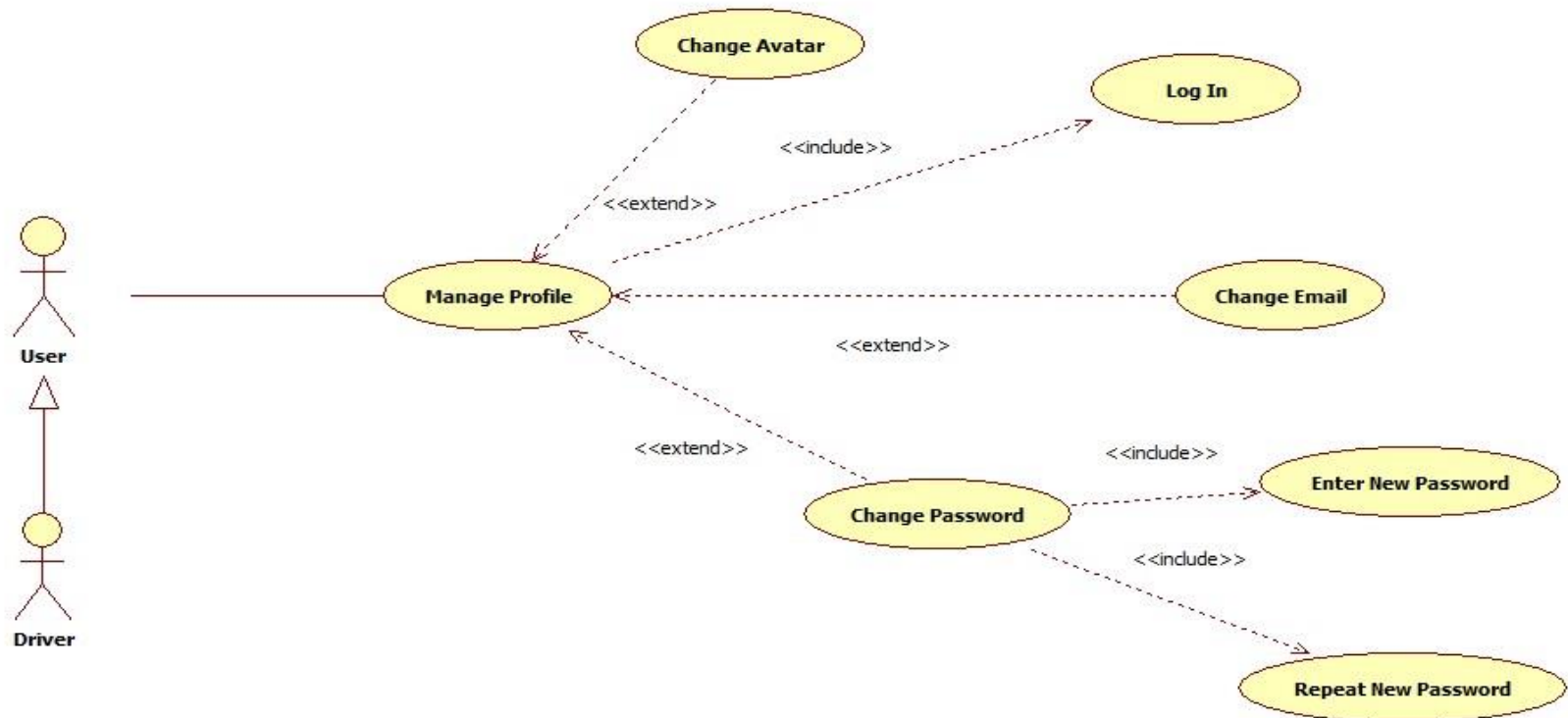
- *Register (accessing the application)*





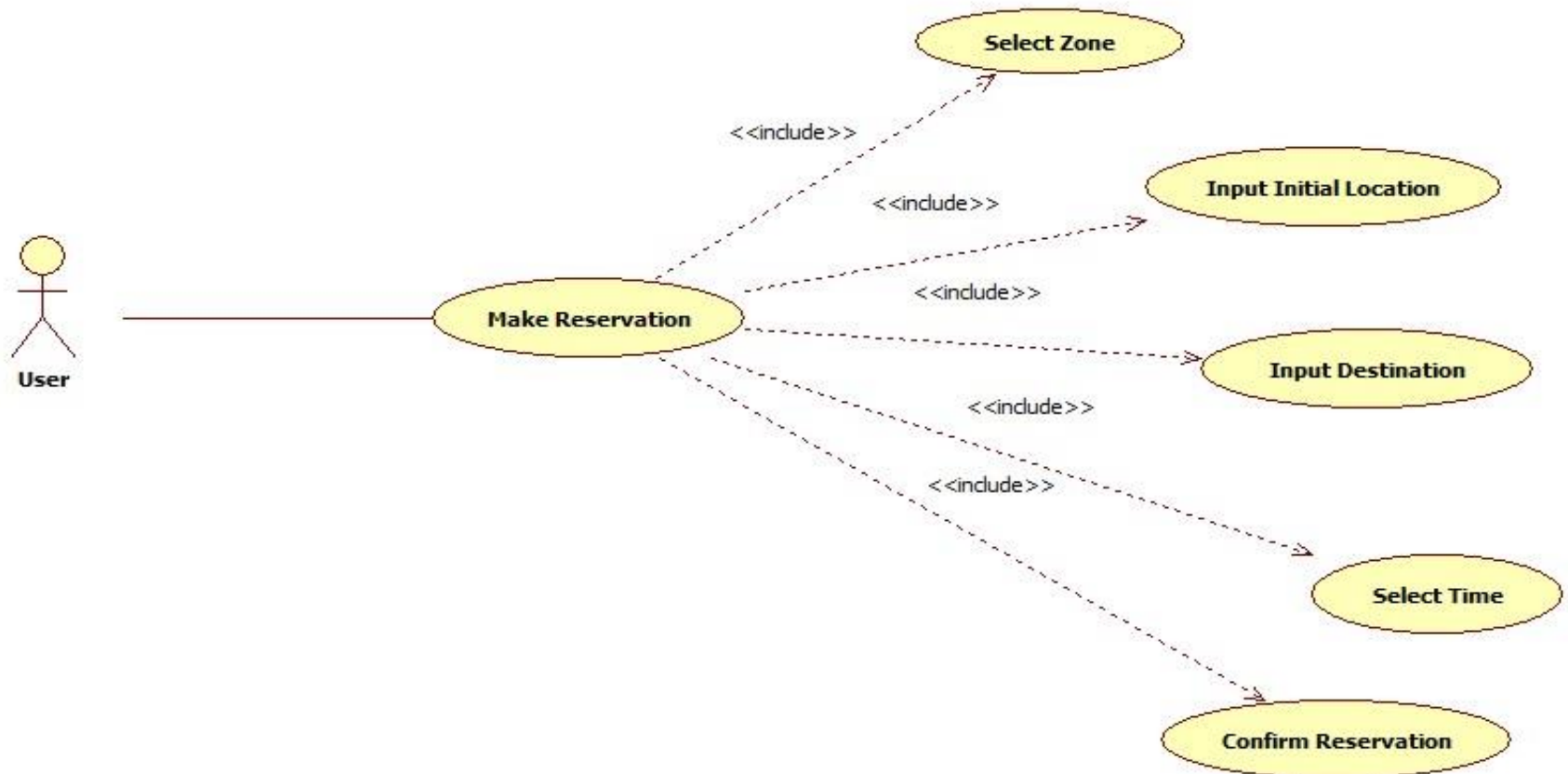
# Use Case Diagrams

- *Manage Profile*



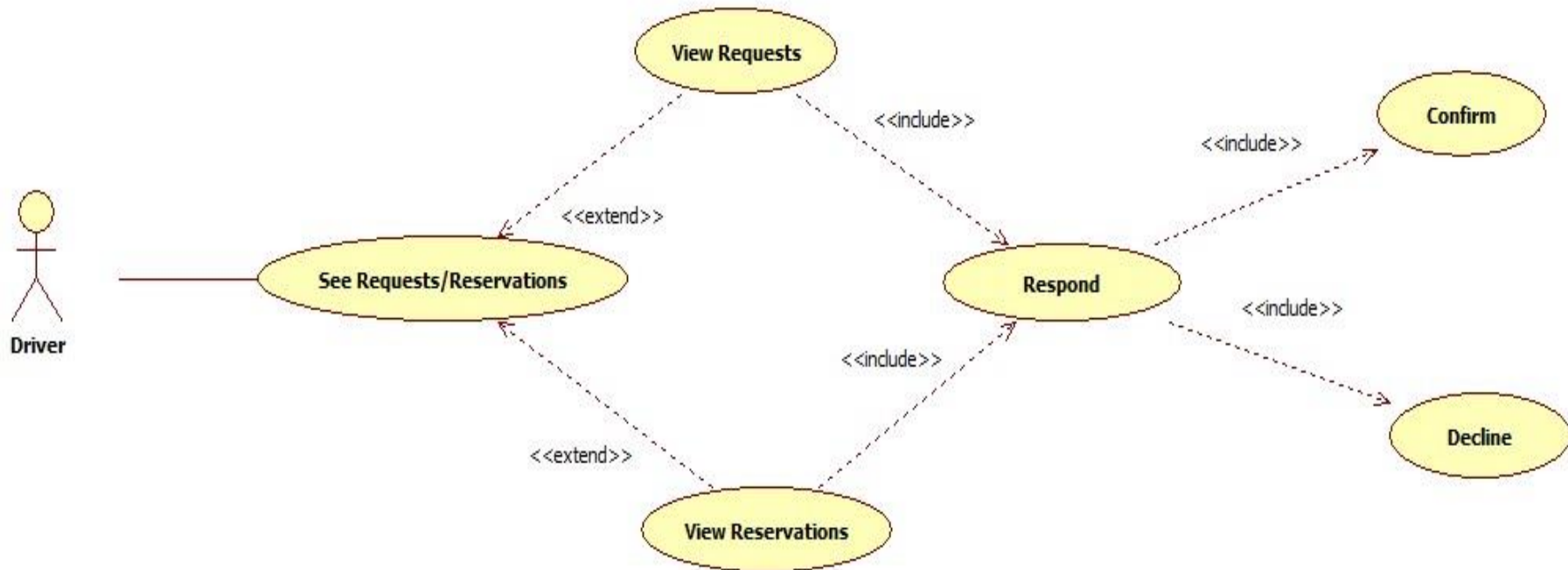
# Use Case Diagrams

- *Make Reservation*

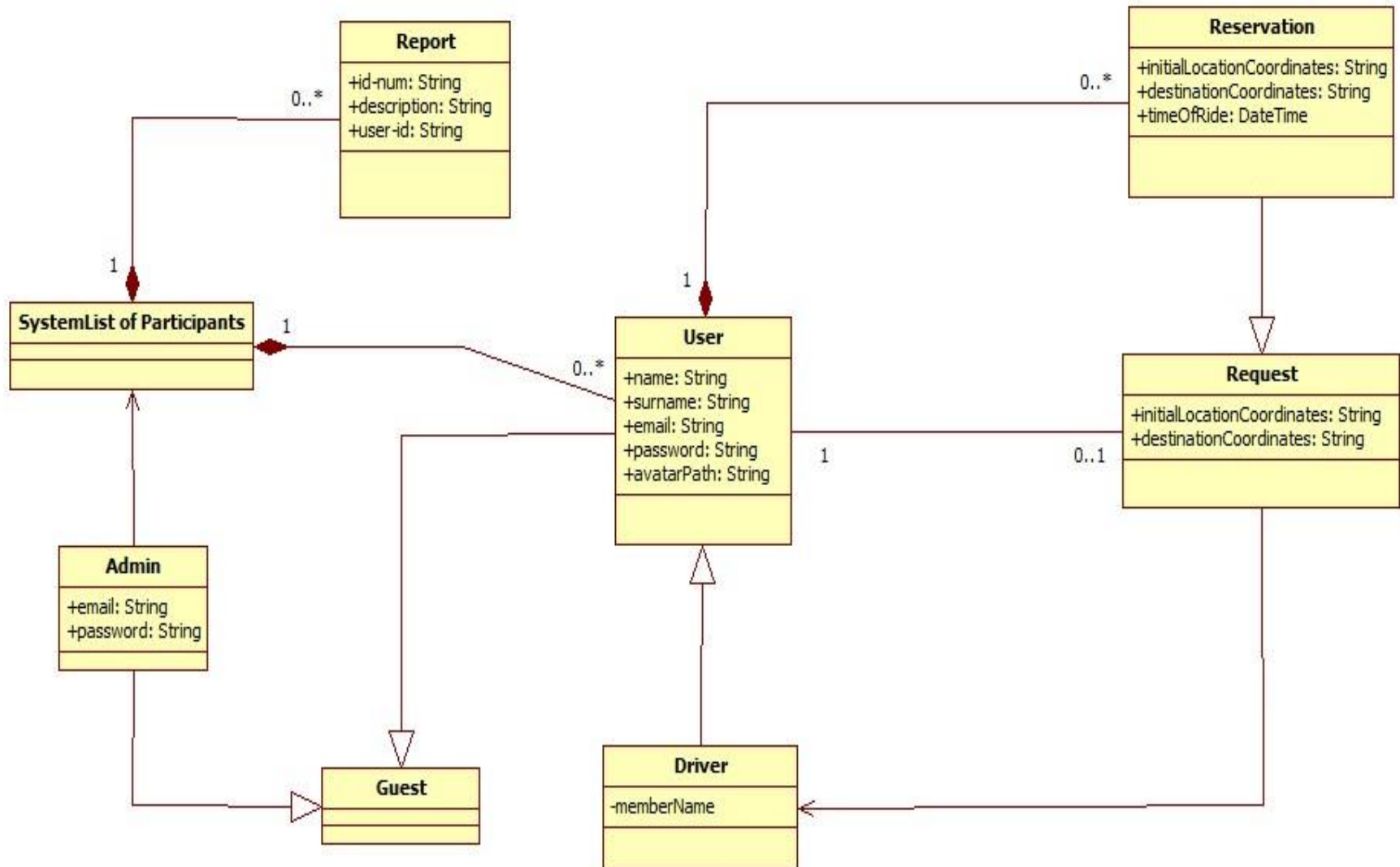


# Use Case Diagrams

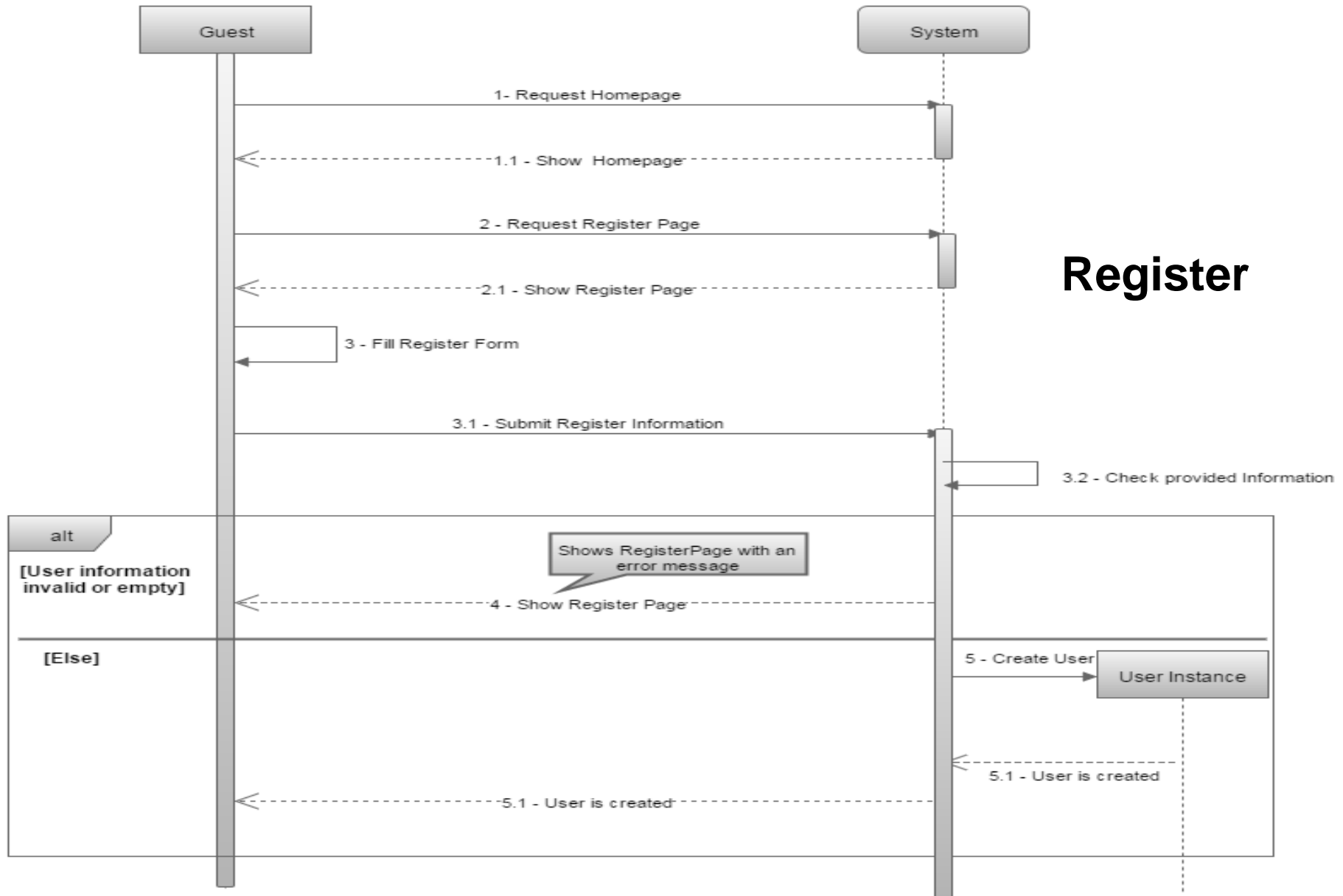
- Driver's Respond to a Request or Reservation*



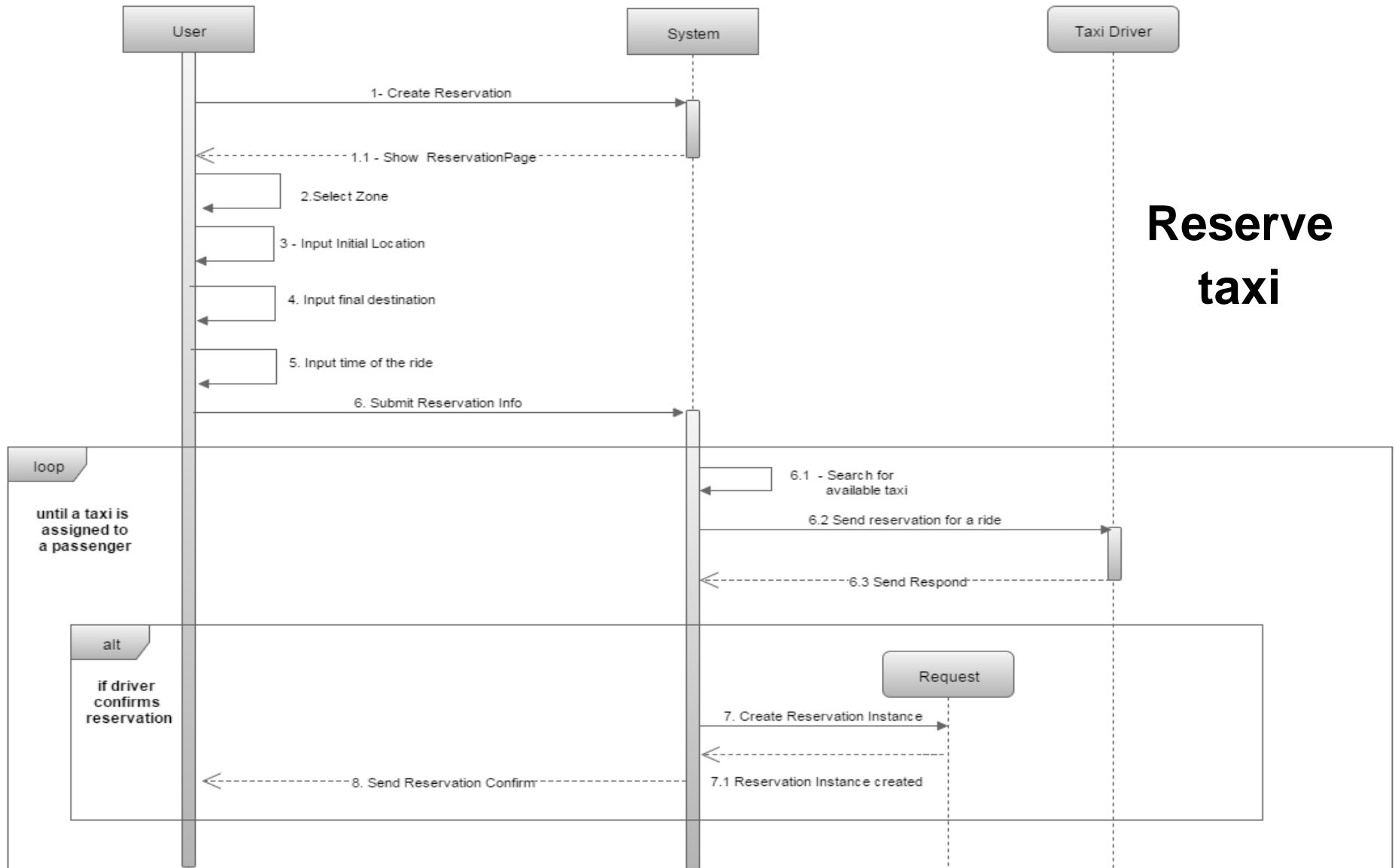
# Class Diagram



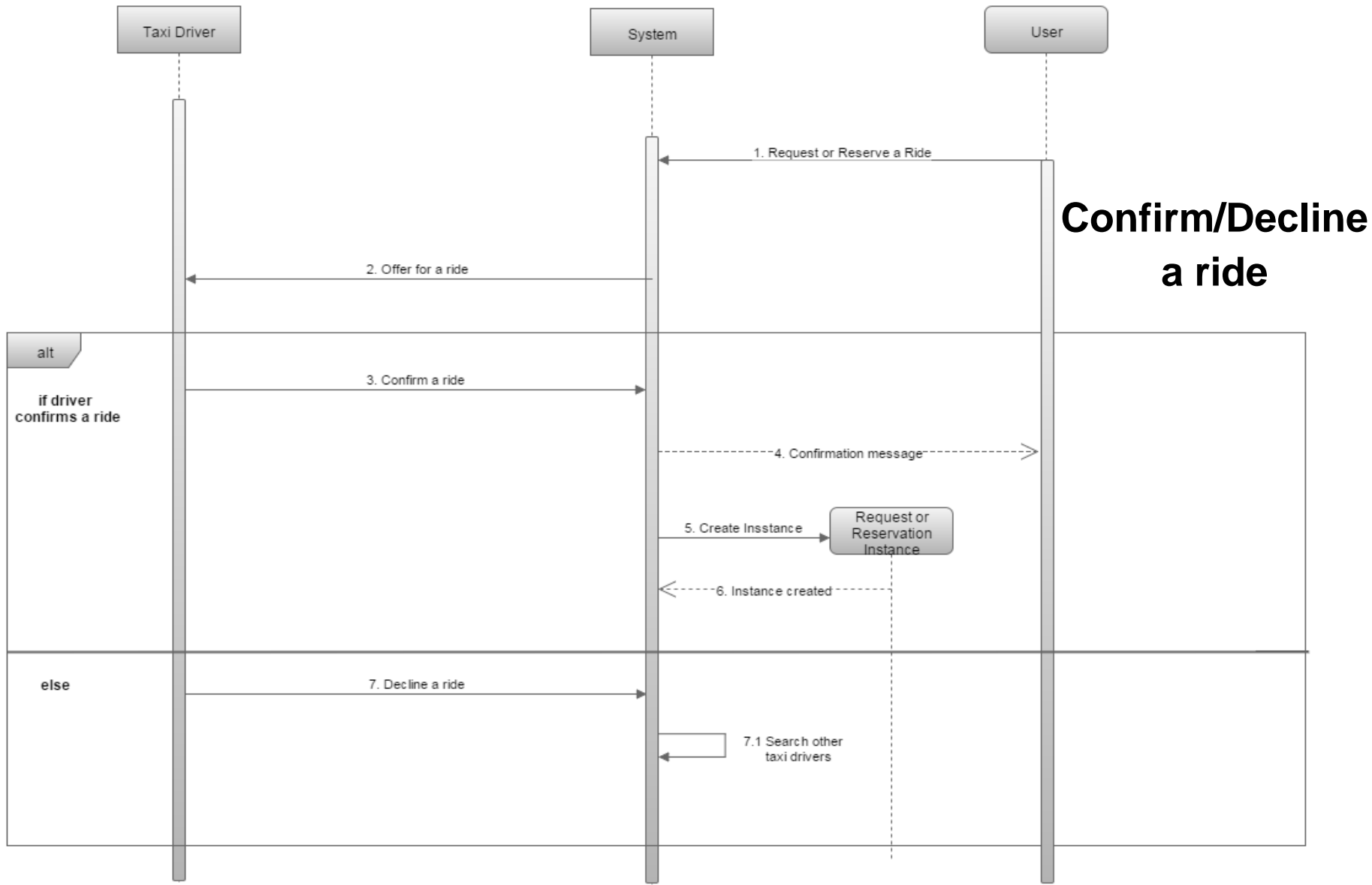
# Sequence Diagrams



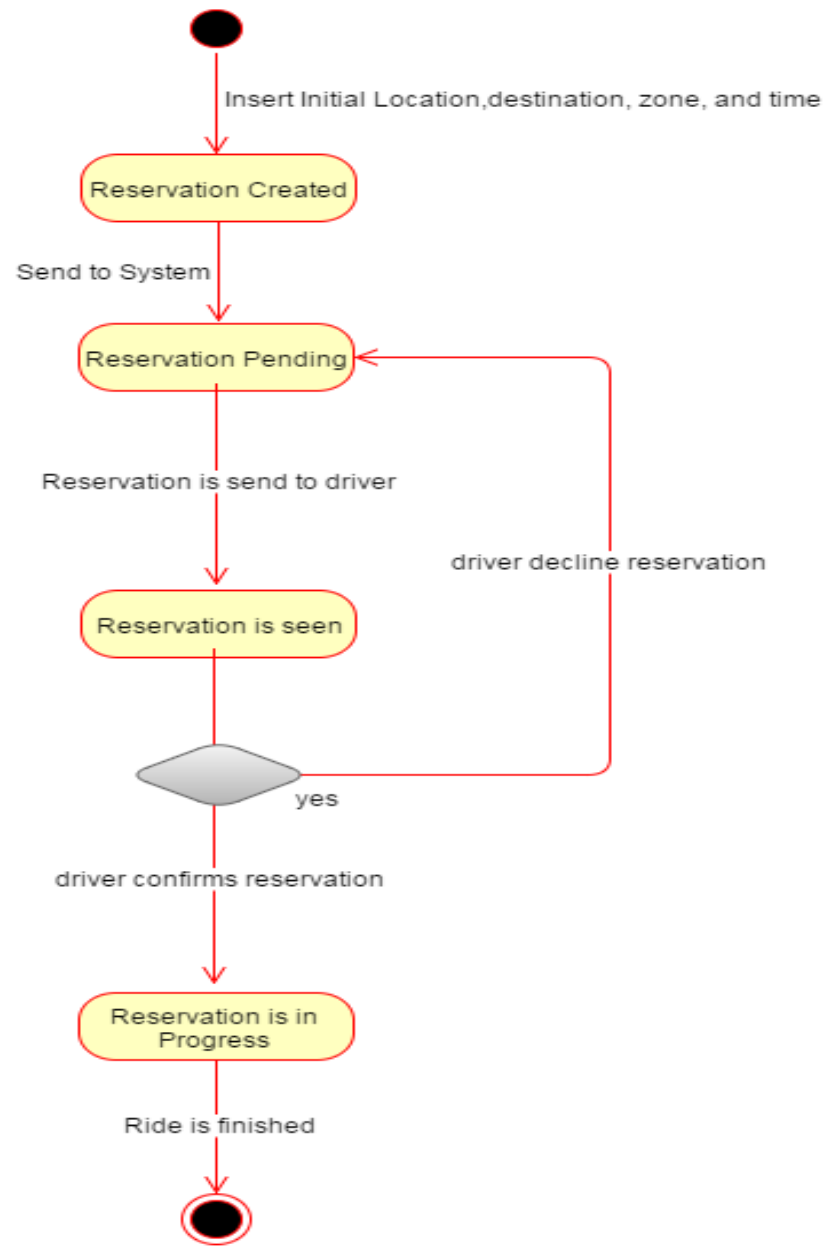
# Sequence Diagrams



# Sequence Diagrams



# State Chart Diagram



## Reservation Lifecycle



# Alloy code

```
module language/myTaxiService
//SIGNATURES
sig Guest{ }

sig User extends Guest {
  reservation: lone Reservation,
  requests: set Request
}

sig Driver extends User{
  car: one TaxiCar
}

sig TaxiCar{
  currentDriver: lone Driver
}

sig Reservation{
  driver: one Driver,
  passenger: one User
}

sig Request extends Reservation{ }

sig Zone{
  drivers: set Driver
}
```

# Alloy code

```
//FACTS
```

```
fact noSameDriverPerCar{
    no d:Driver | some t1,t2: TaxiCar |
    t1!=t2 and d in t1.currentDriver and d in t2.currentDriver
}

fact connectionCarDriver{
    all t:TaxiCar | all d:Driver | t in d.car => t.currentDriver=d
}

fact differentCarsTwoDrivers{
    no t:TaxiCar | some d1,d2: Driver | d1!=d2 and d1.car=t and d2.car=t
}

fact differentCarsTwoDrivers1{
    all d:Driver | all t:TaxiCar | d in t.currentDriver => t in d.car
}

fact NoDriverAPassenger{
    no u:Driver | some res:Reservation | u in res.passenger
}

fact diffDriversPerReservation{
    all res1, res2:Reservation | some d1,d2:Driver | (d1 in res1.driver and d2 in res2.driver) => (d1!=d2)
}

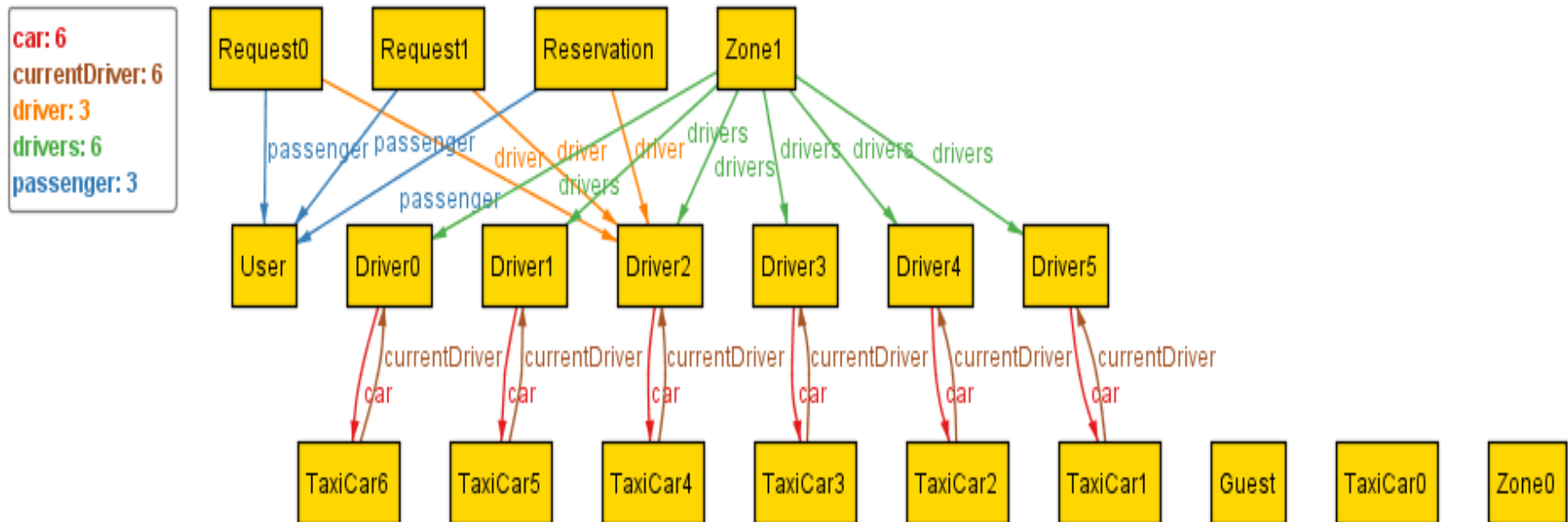
fact diffUsersPerReservation{
    all res1,res2:Reservation | some u1,u2:User | (u1 in res1.passenger and u2 in res2.passenger) => (u1!=u2)
}

fact oneZonePerDriver{
    all d:Driver | one z:Zone | d in z.drivers
}
```

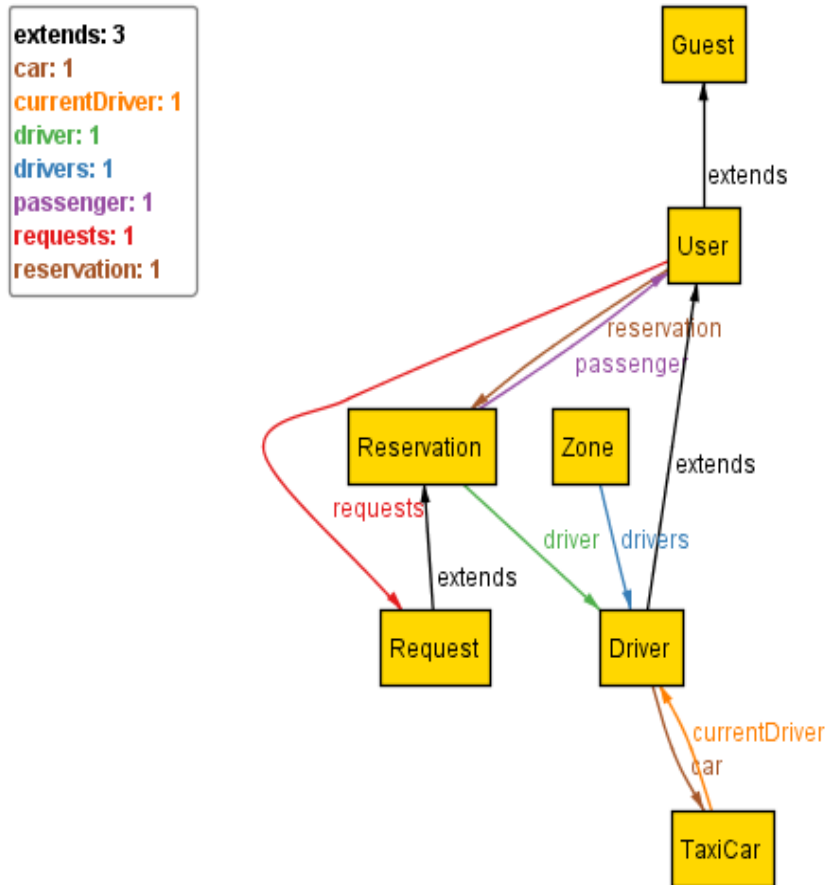
# Alloy code

```
//PREDICATES
pred show{
  #Guest=8
  #User=7
  #Driver=6
  #Request=2
  #Reservation=3
  #Zone=2
  #TaxiCar=7
}
run show for 20
```

# World Generated



# Metamodel



Main constraints of the application:

- Driver belongs only to one zone
- Request belongs only to one user
- Reservation belongs only to one user
- User belongs only to one reservation
- Reservation belongs to one driver
- Taxi car belongs to one driver
- Driver have only one taxi car

# Used tools

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

**Axure RP Pro 7.0:** to create the sketches for the interface of the web version of the application

**JustInMind Prototyper 6.9.1:** to create the sketches for the interface of the mobile version of the application

**Draw IO and StarUML:** to create the State Charts, the Class Diagram, the Sequence Diagrams and the Use Case Diagram

**Alloy Analyzer 4.2:** to prove the consistency of our model

# Questions



**Thank you for your attention!**