Software Engineering 2 AA 2015-2016

MyTaxiService - RASD

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

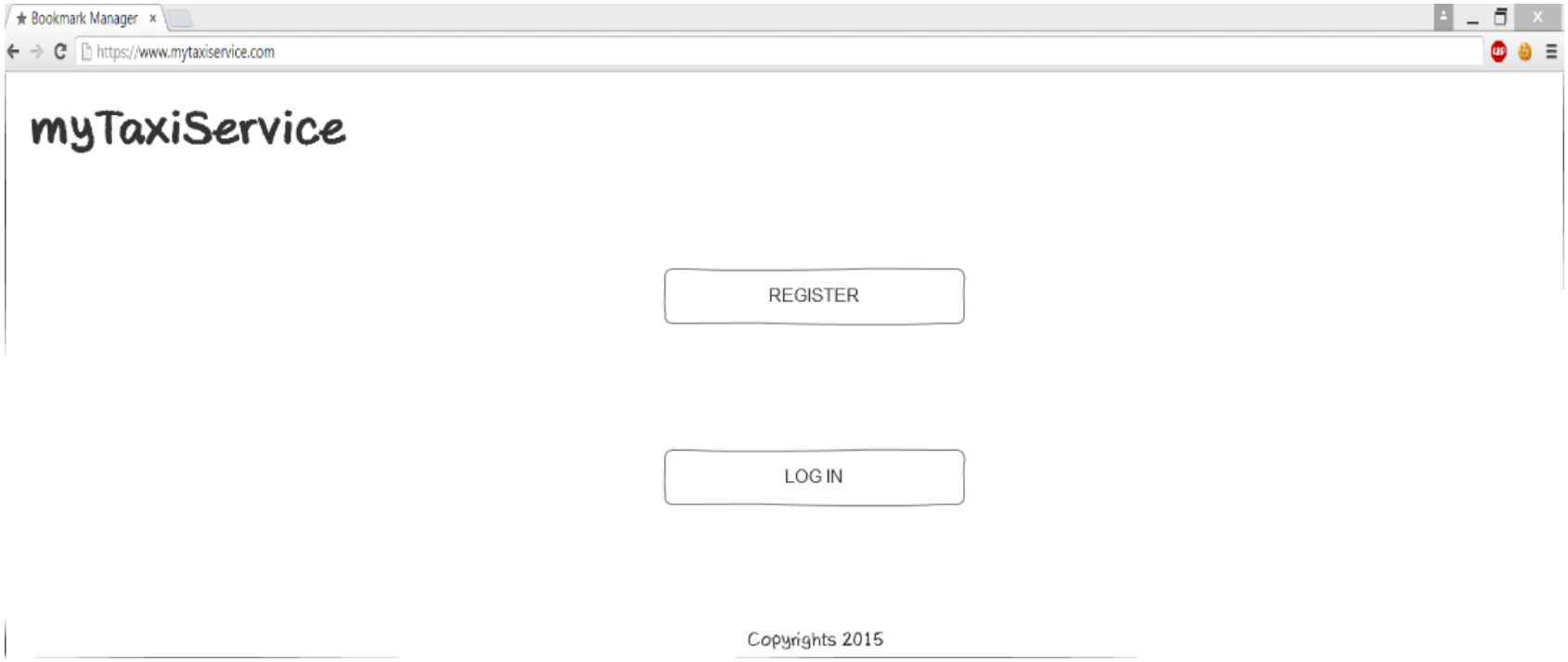
Assumptions

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

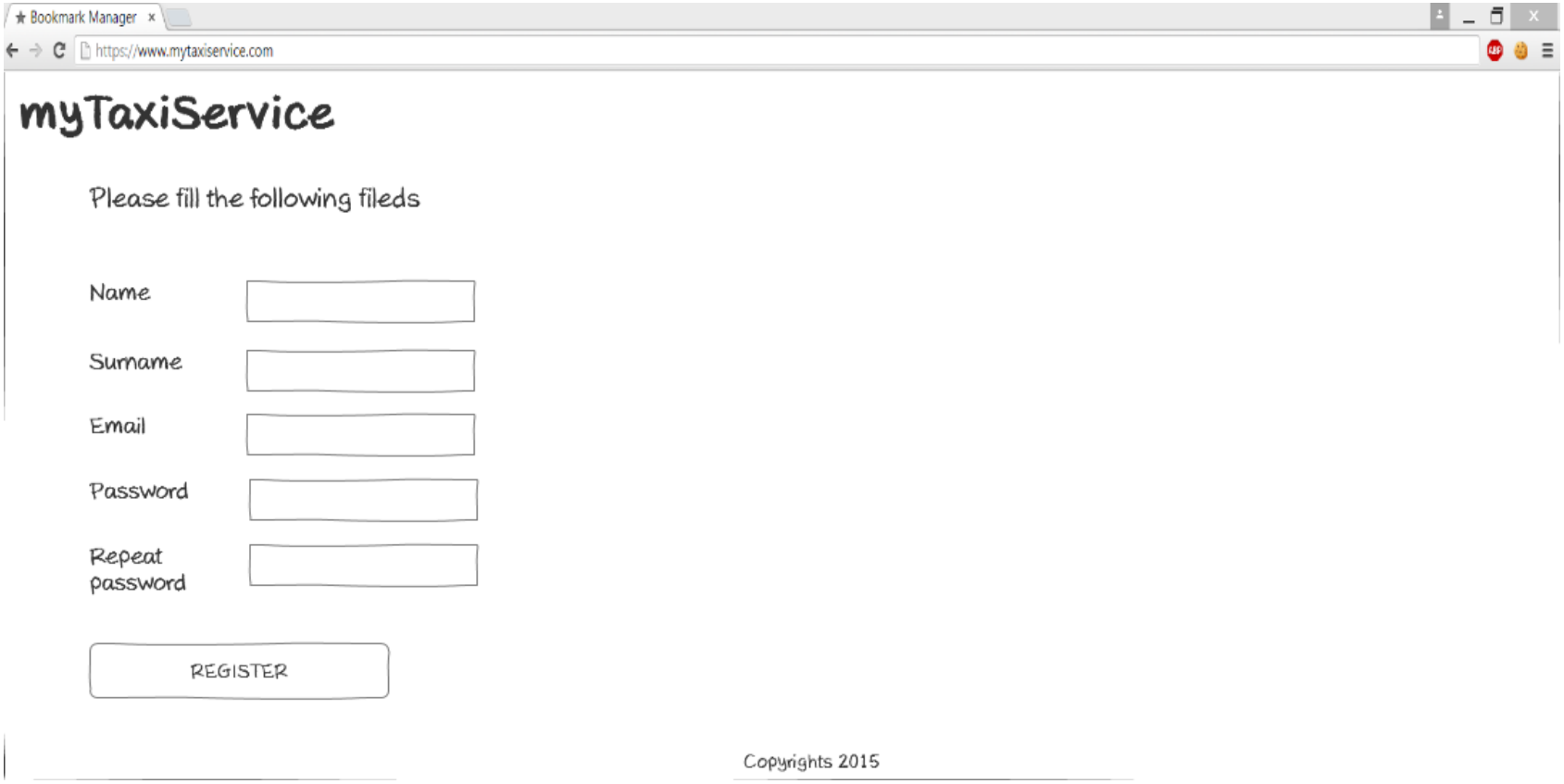
User-interface

- Sketches for the web version of the application



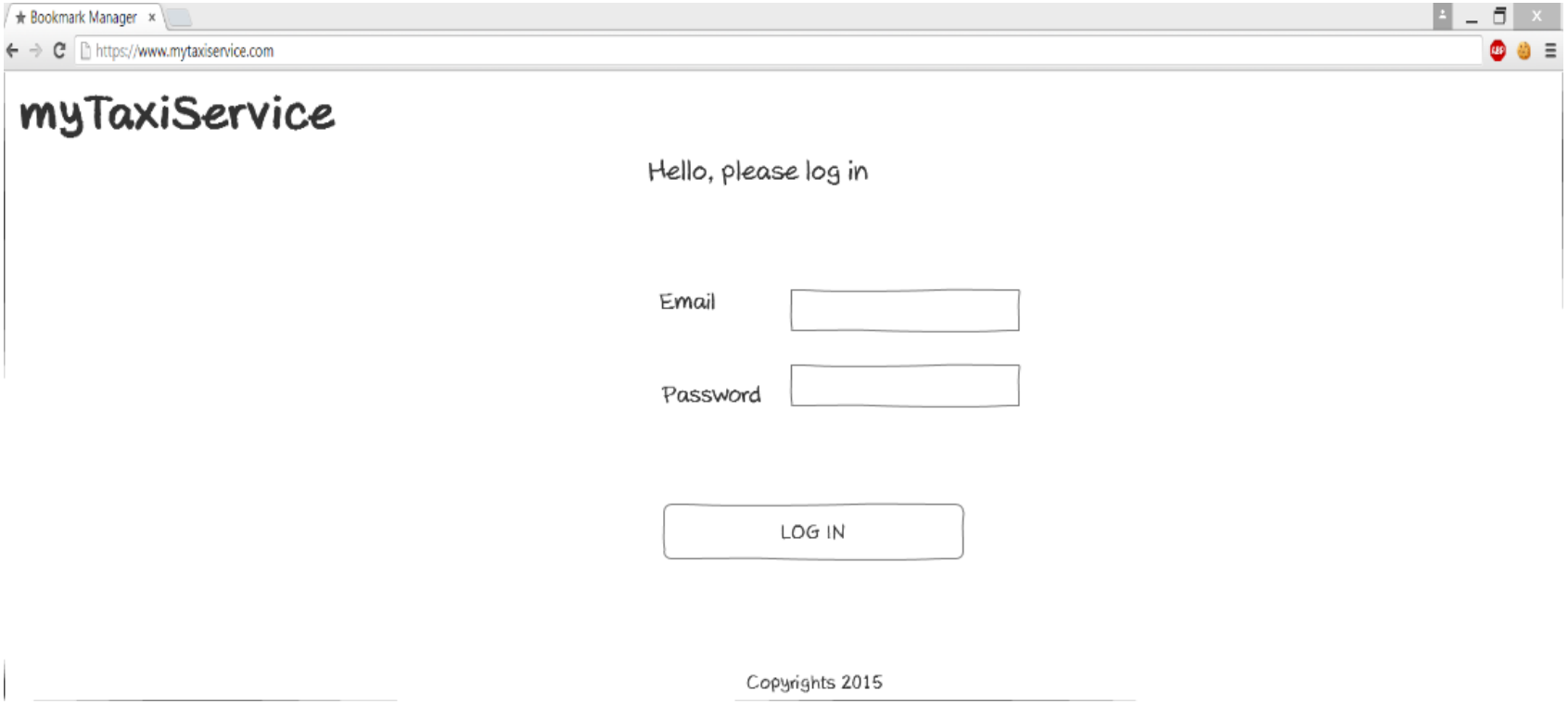
Initial page

User-interface



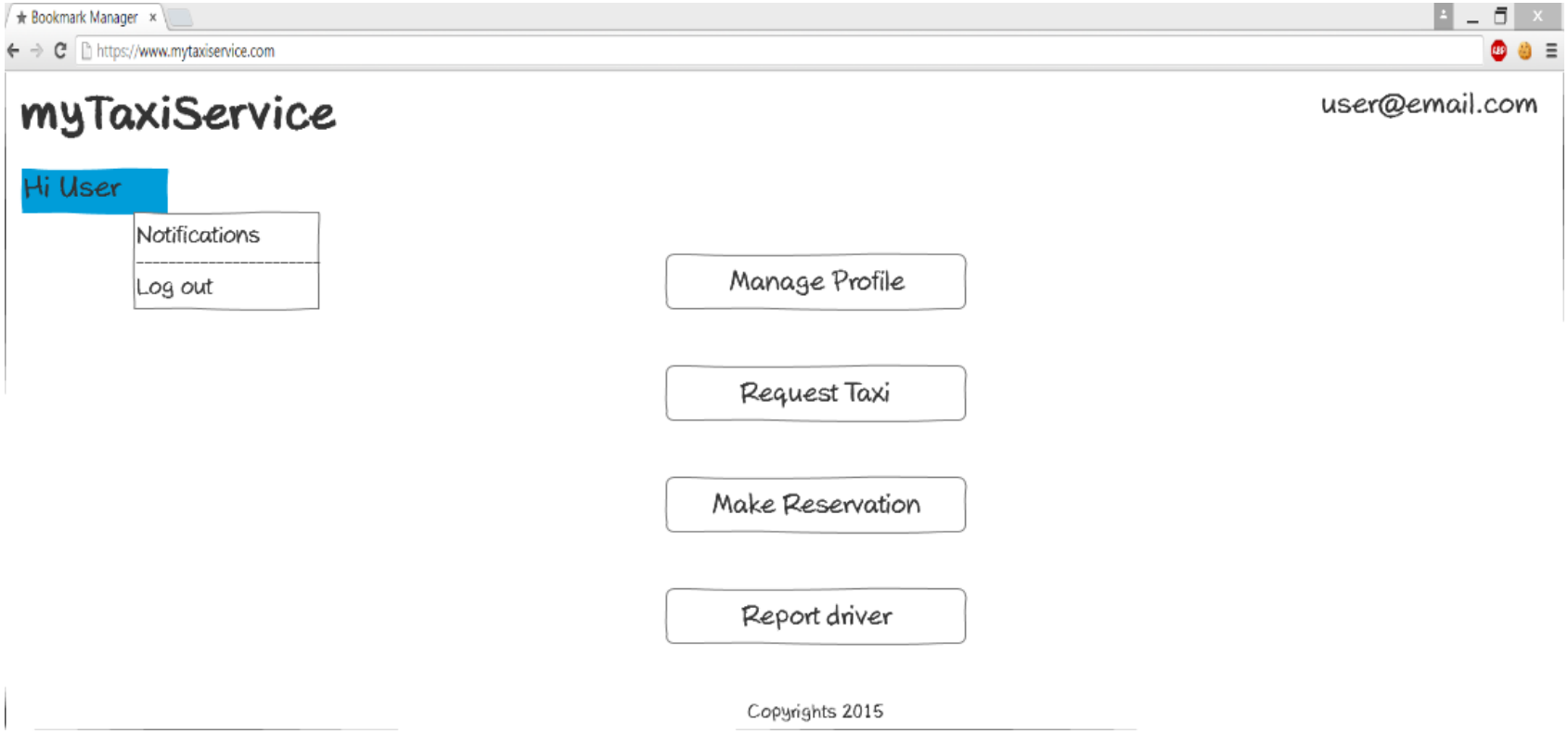
Register page

User-interface



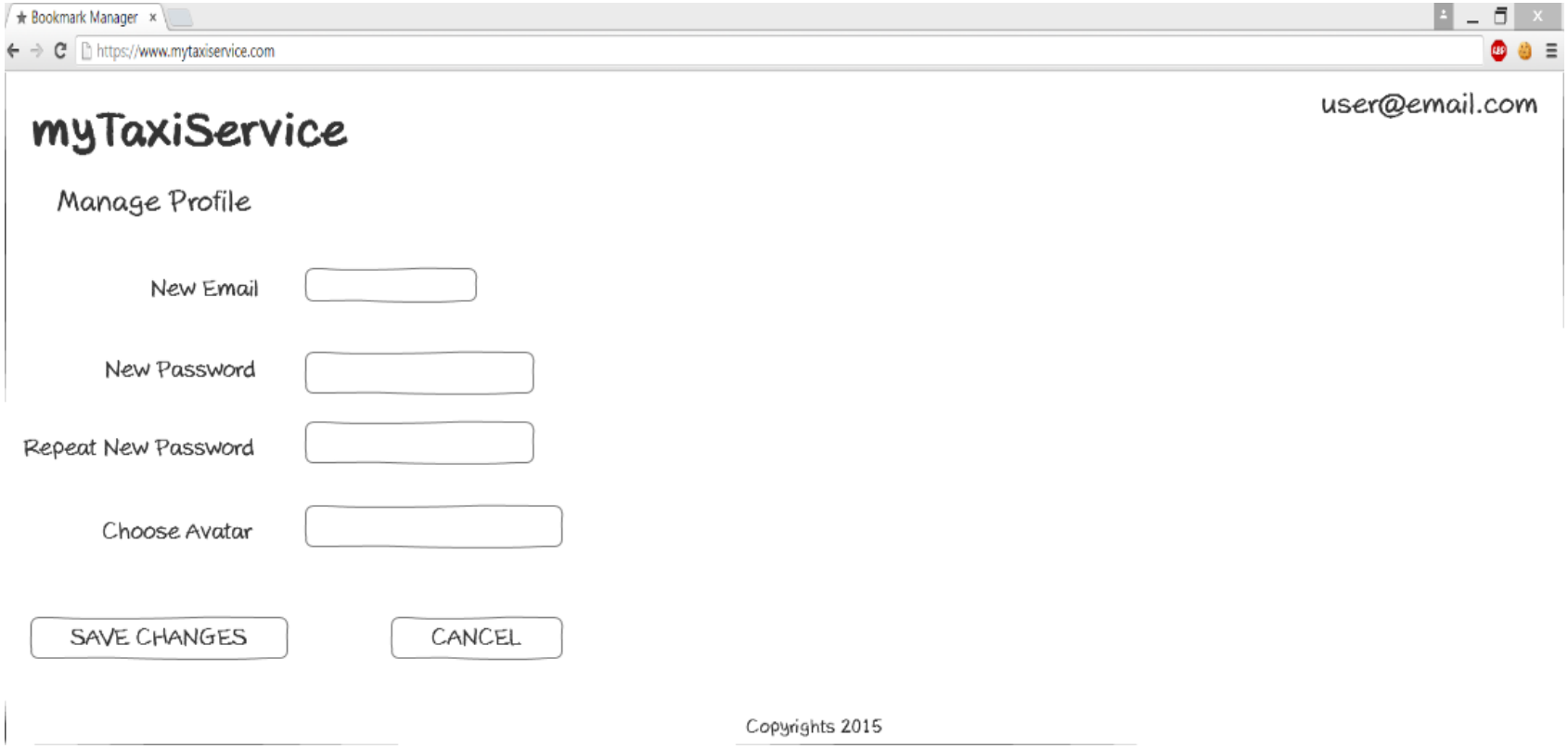
Login page

User-interface



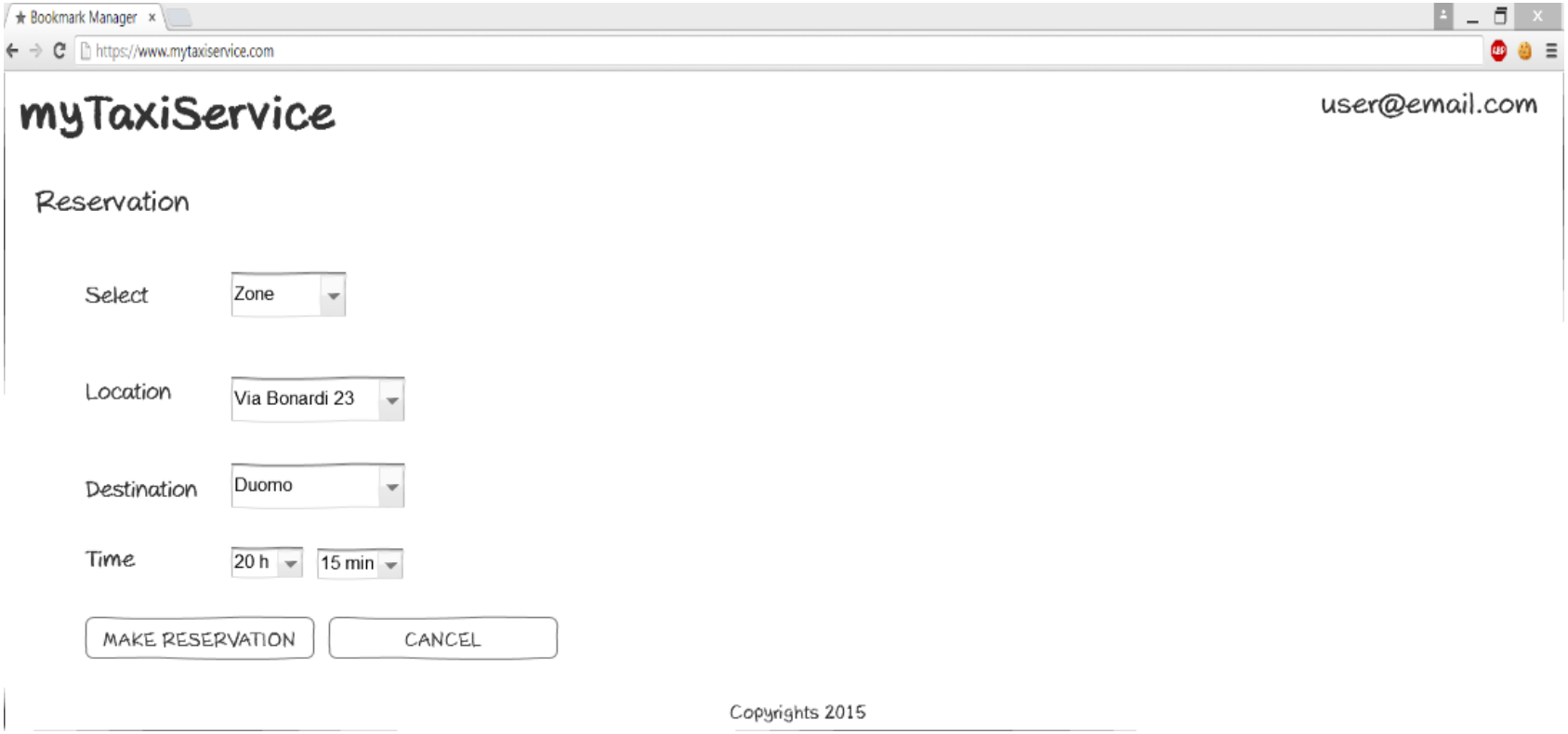
Homepage

User-interface



Manage profile page

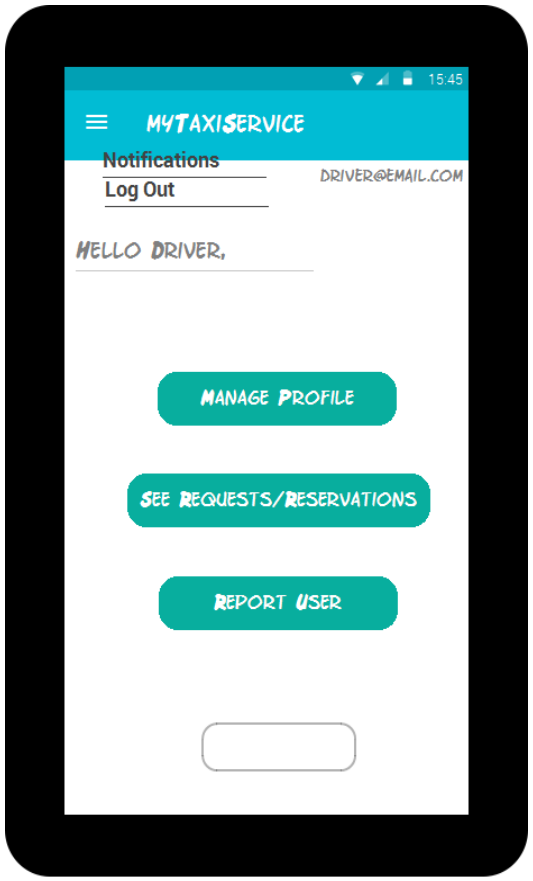
User-interface



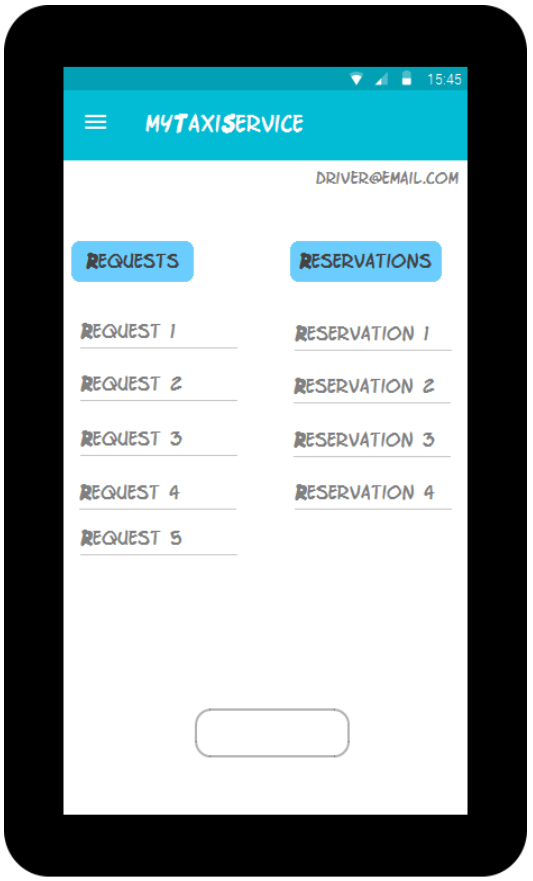
Reservation taxi page

User-interface

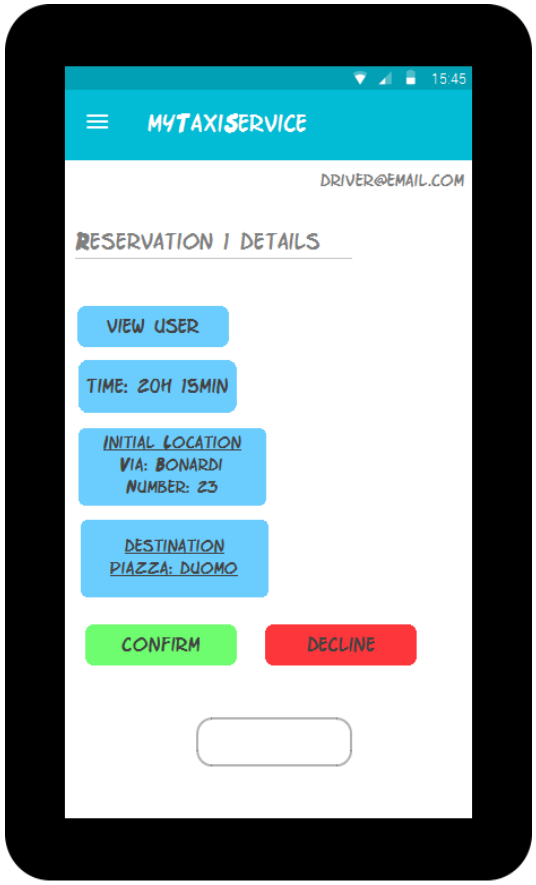
- Sketches for the mobile version of the application



Driver homepage



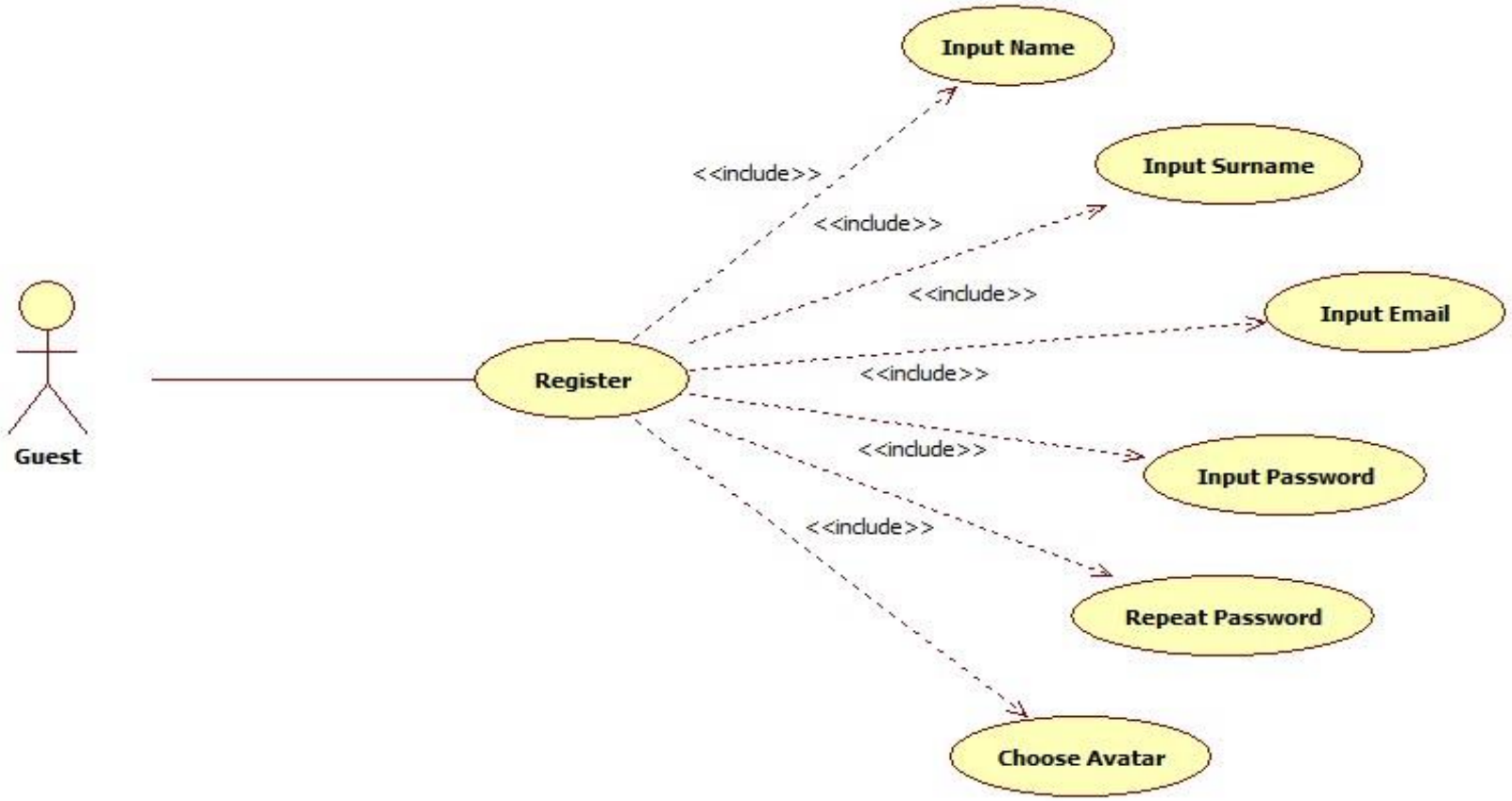
Notification page



Reservation details page

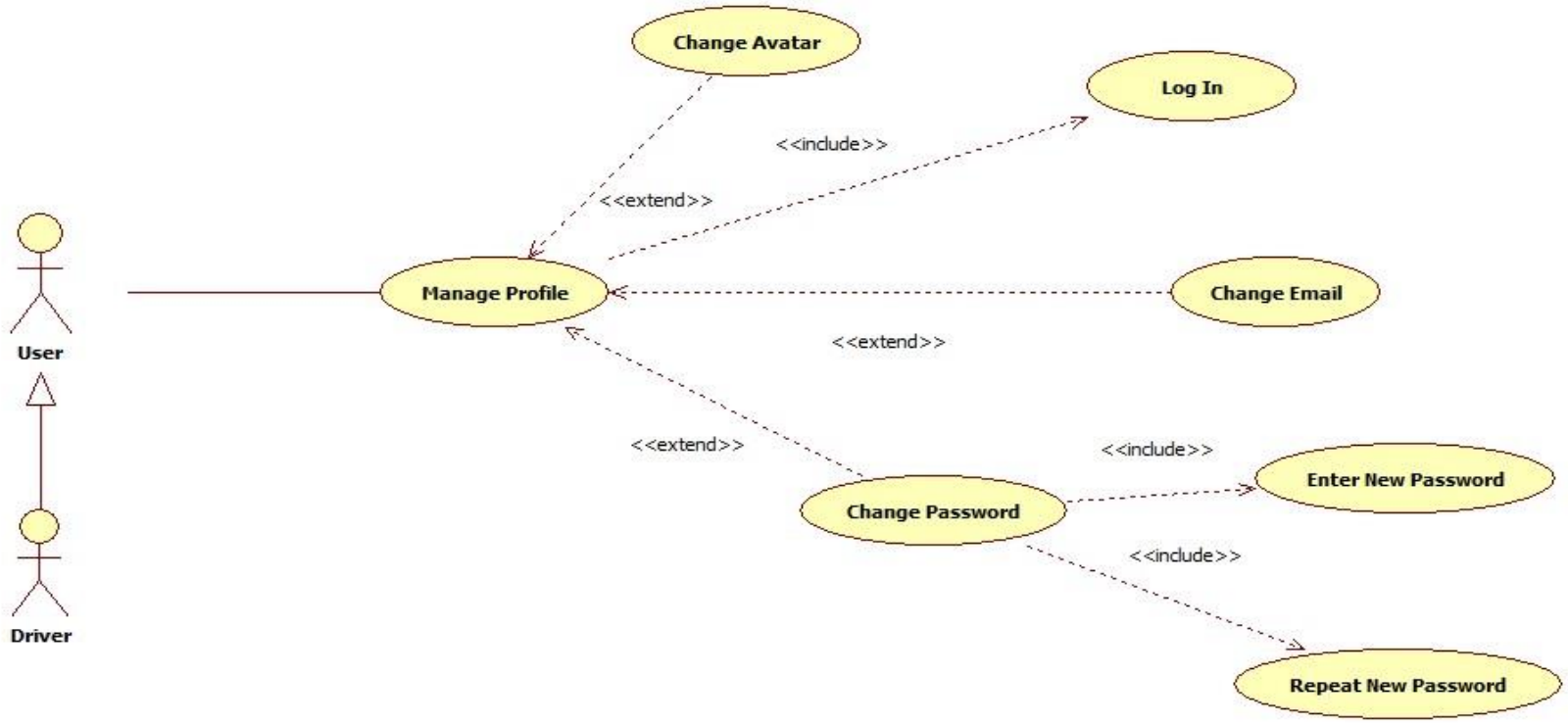
Use Case Diagrams

- *Register (accessing the application)*



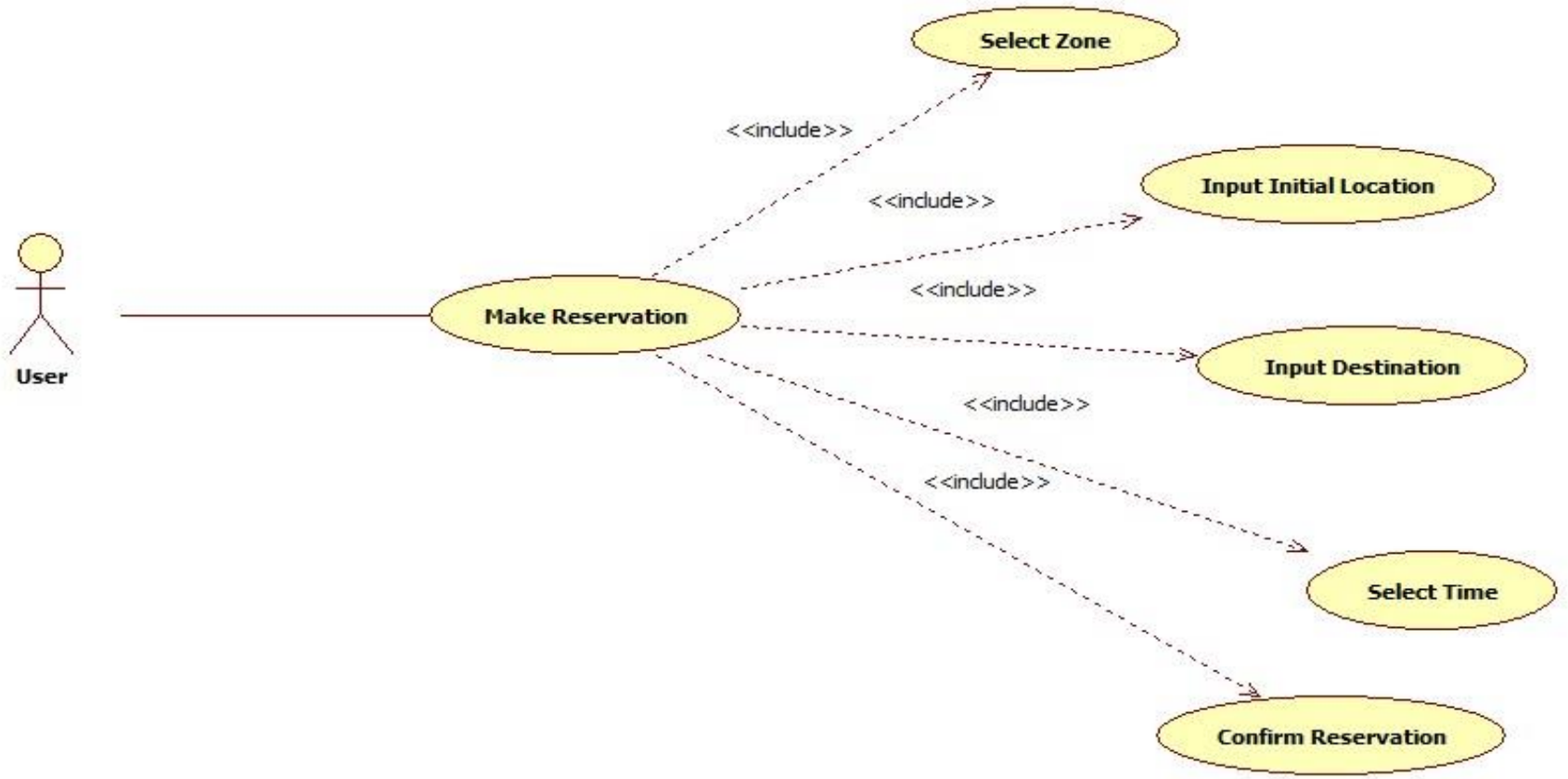
Use Case Diagrams

- *Manage Profile*



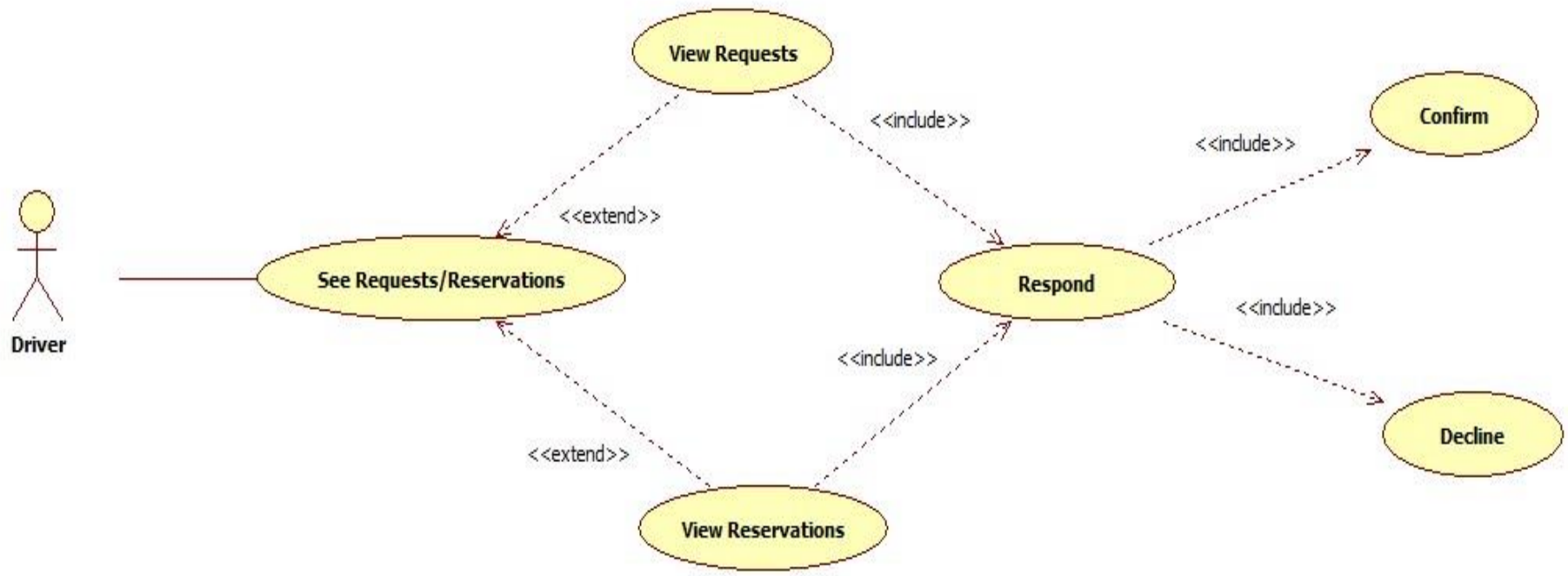
Use Case Diagrams

- *Make Reservation*

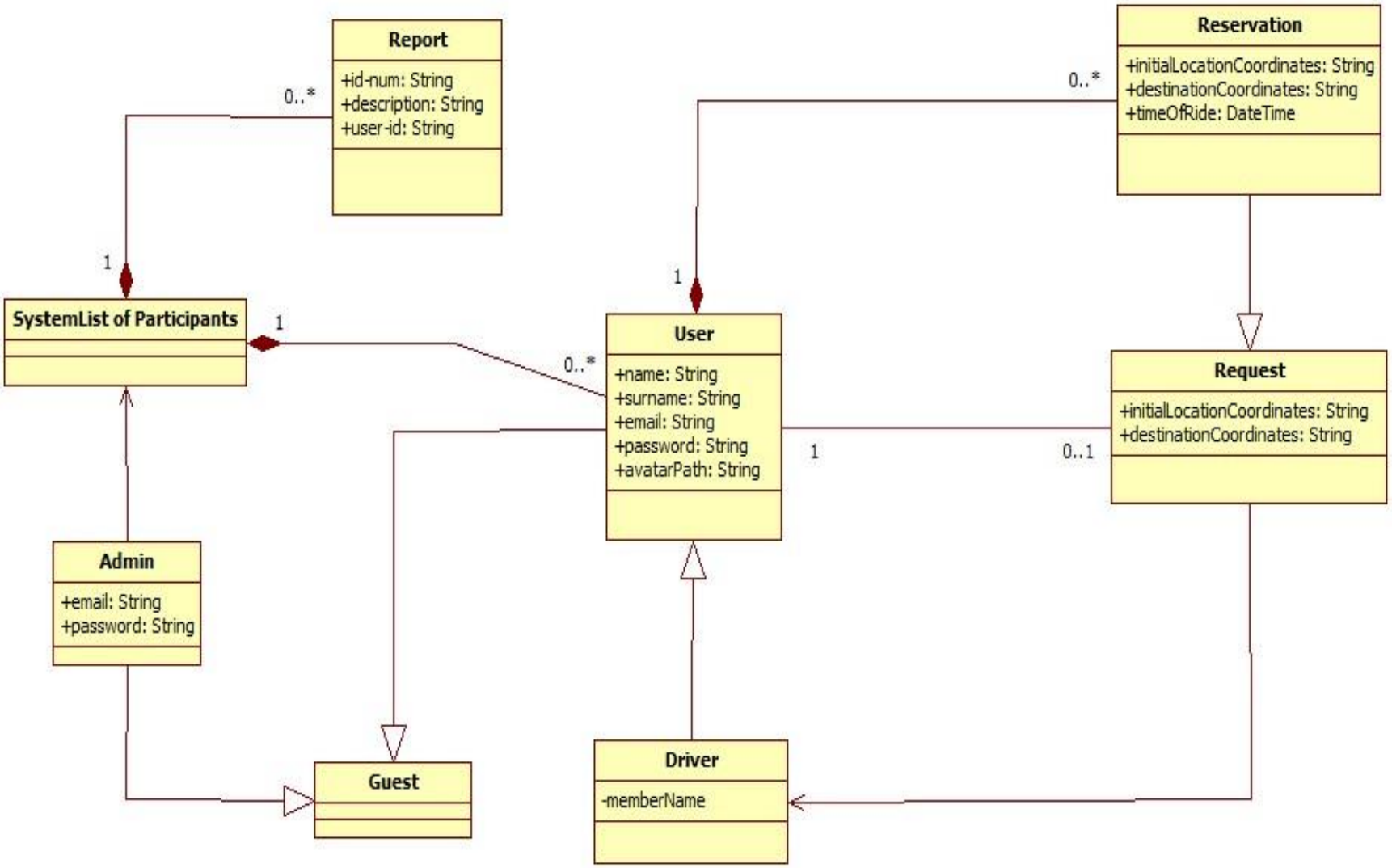


Use Case Diagrams

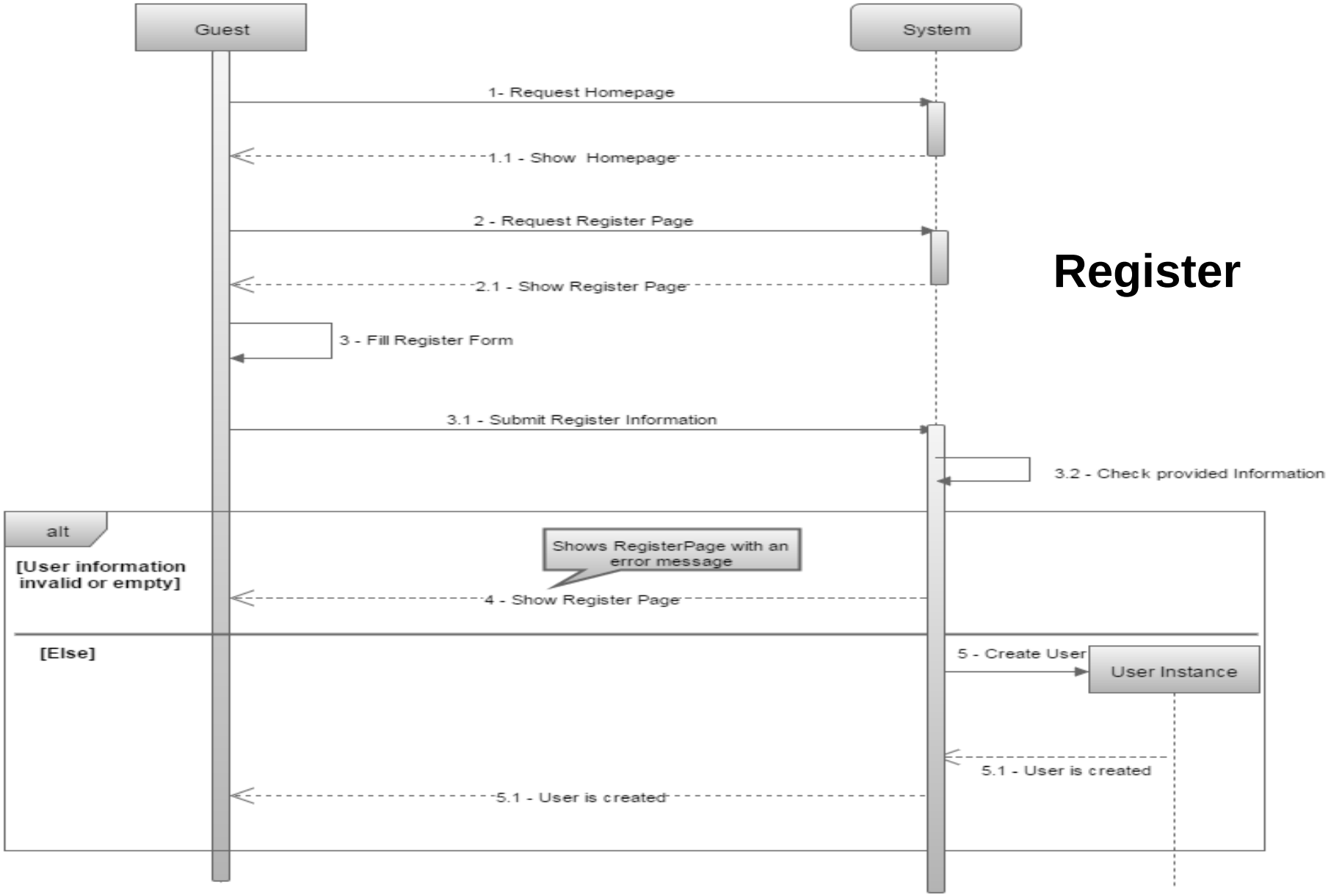
- *Driver`s Respond to a Request or Reservation*



Class Diagram

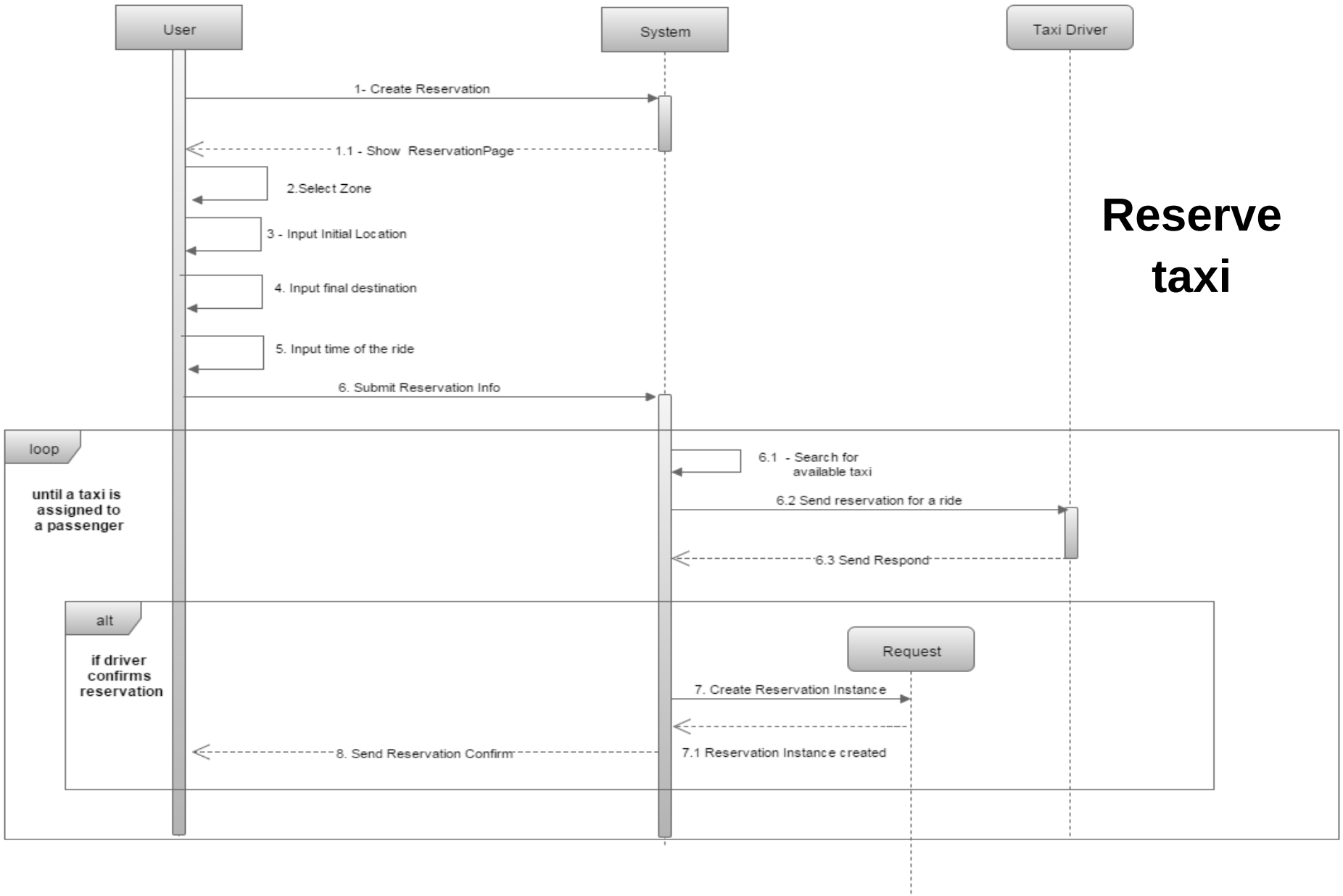


Sequence Diagrams

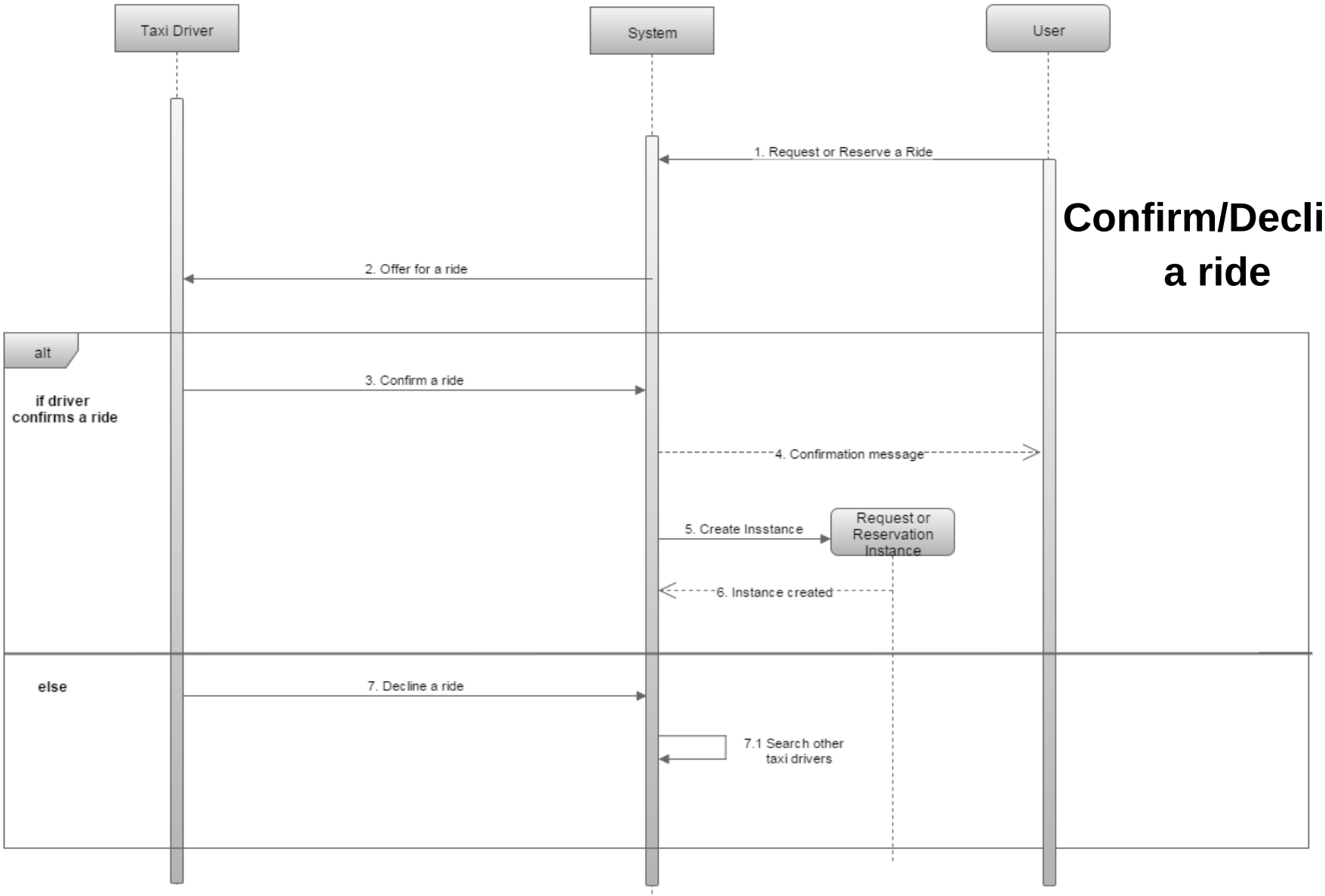


Register

Sequence Diagrams

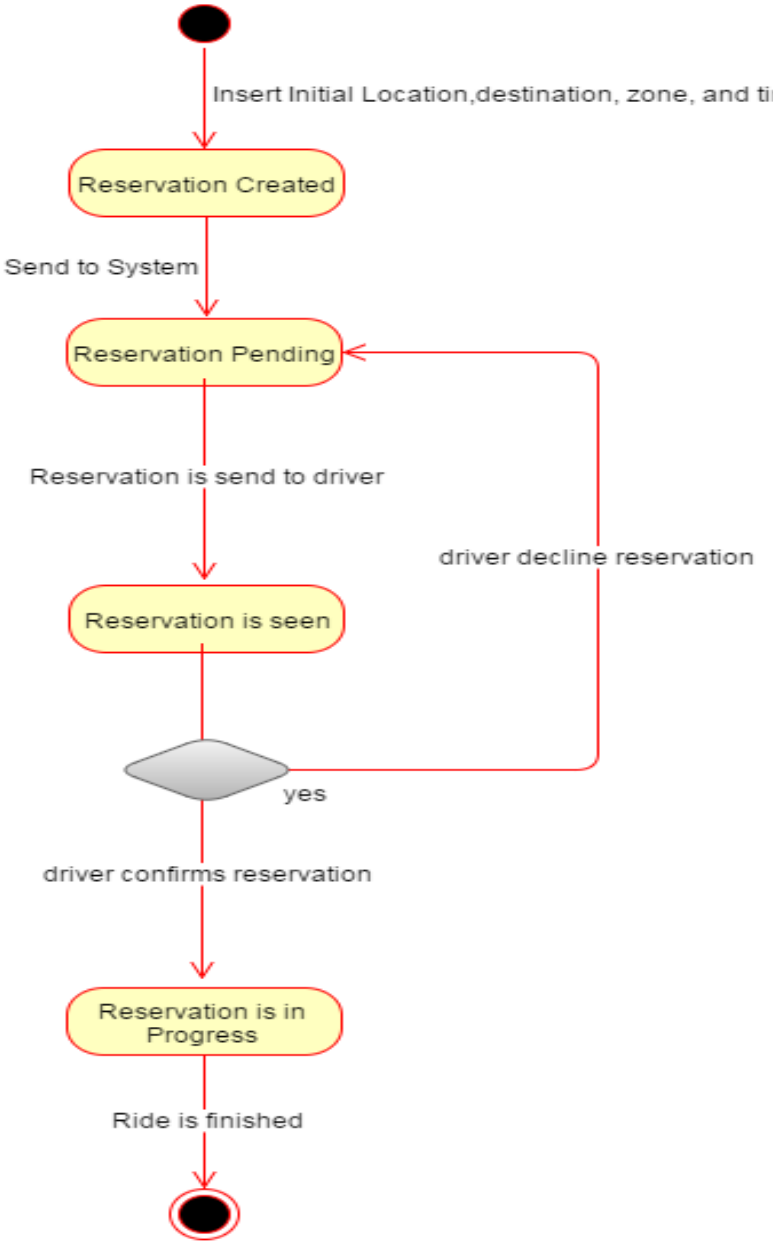


Sequence Diagrams



Confirm/Decline a ride

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
}
```

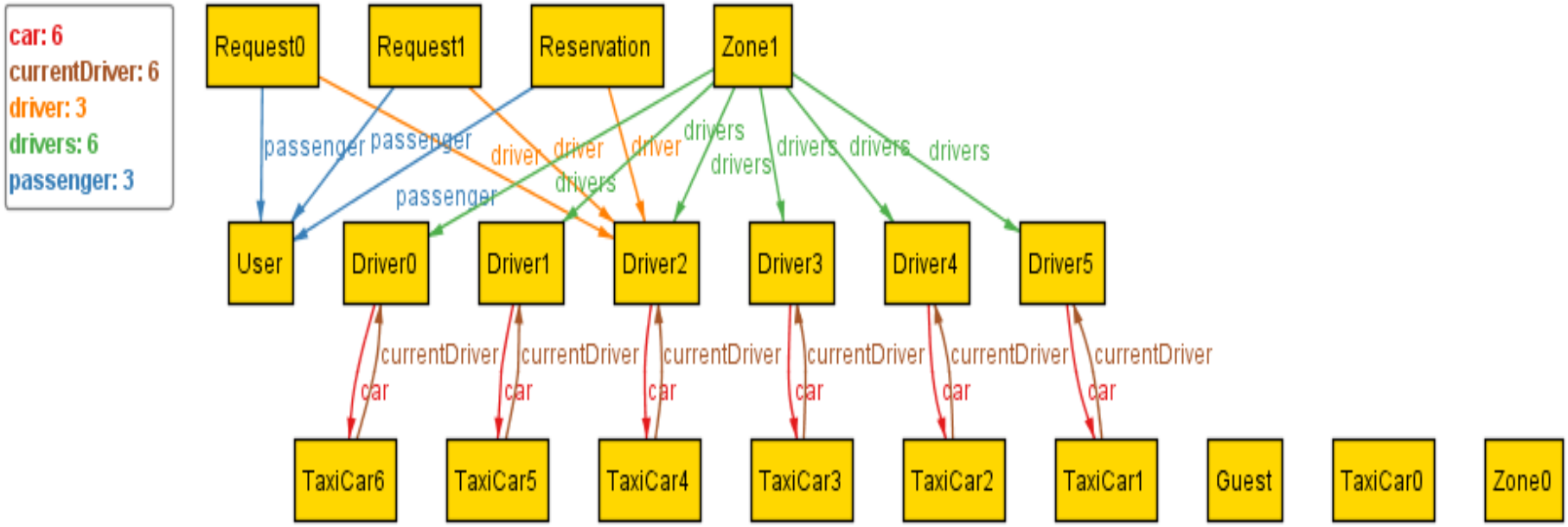
Alloy code

```
//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
}
```

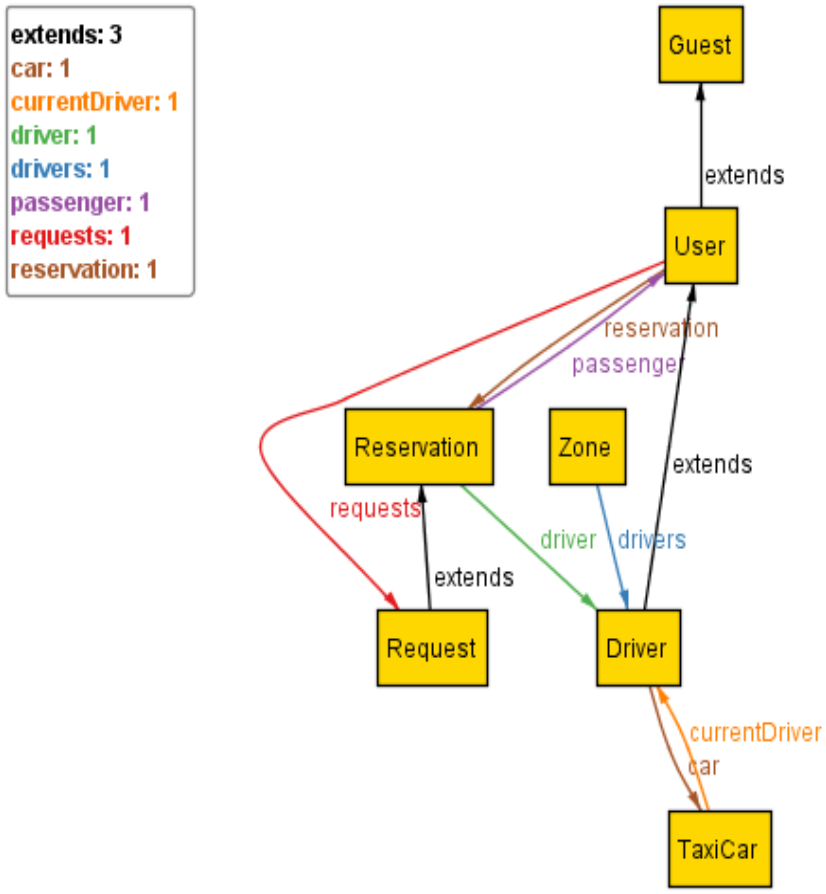
Alloy code

```
//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 Analizer 4.2: to prove the consistency of our model

Questions

Thank you for your attention!