Software Engineering 2 – A.Y. 2015/2016

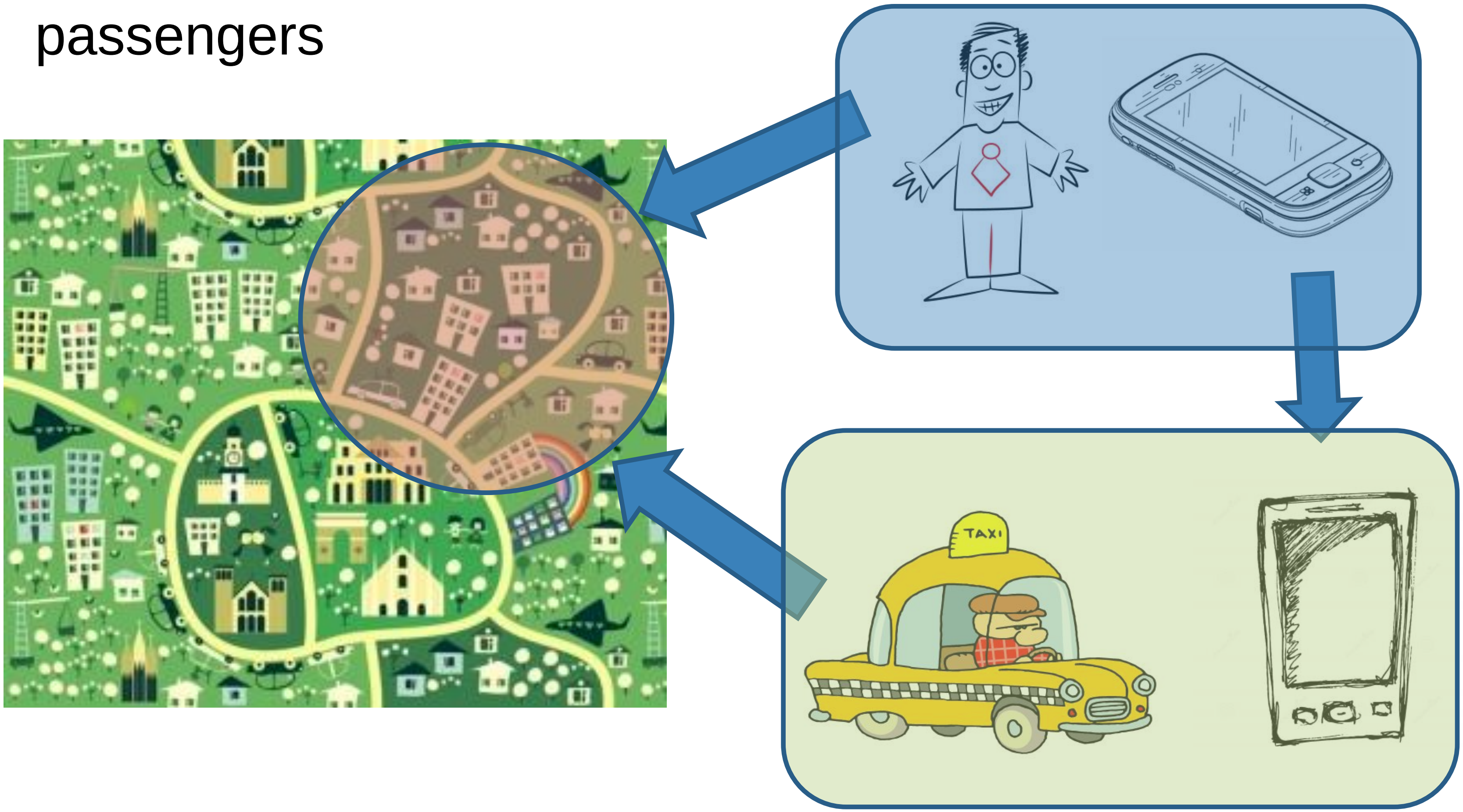
POLITECNICO DI MILANO

RASD

Jovanovic Milica 835953
Vidanovic Pavle 854472

myTaxiService software application

- Application similar to Uber, which makes the process of assigning an available taxi vehicle to possible passengers



Structure of RASD

- Introduction
  - Describe the structure of the document
  - Give a basic overview about the system to be developed
  - Glossary
- Overall description
  - Give a list of stakeholders
  - Describe the user characteristics
  - Lists the actors in the system
  - List of goals, assumptions and constraints
  - Describes the domain of the problem
- Specific Requirements
  - Mockups
  - Scenarios
  - Use cases and sequence diagrams
  - Class diagram and State Chart diagrams
  - Non functional requirements
- Alloy
  - Worlds generated by Alloy Analyzer
  - Prove the model's consistency

Overall description

- Stakeholders:
  - Company
    - provided project specification
    - expect it to be delivered in a way that satisfies given specification
    - respecting the set deadlines and budget
  - Developer group
  - Taxi driver
    - person working for the company that ordered the software product
  - Passenger
    - person who need a ride to specific location

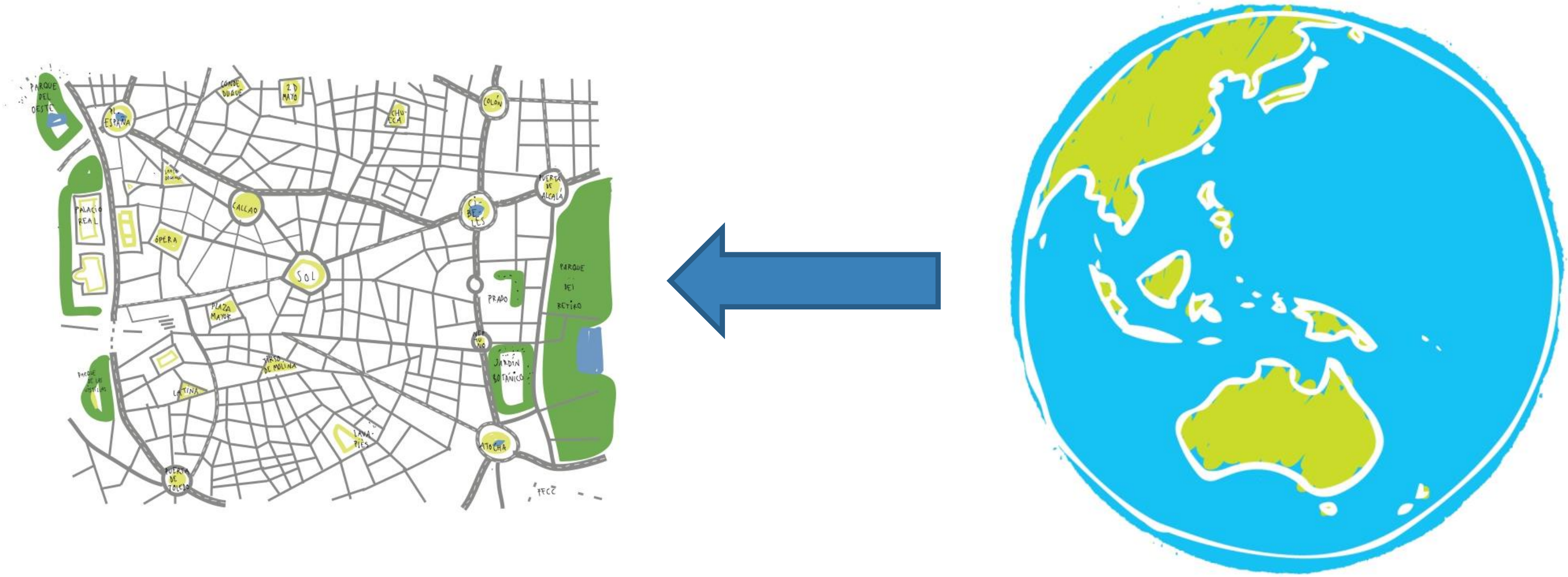
- Actors:
  - Guest - person accessing a system that has either never registered of hasn't logged in yet
  - User - person already registered and logged into the system
  - Taxi driver - same as User, but can access to all features offered by the driver application
  - Admin - person responsible for handling reports, can ban users or driver from system

Overall description

- Goals:
  - [G1] registering new user
  - [G2] login to existing user’s account
  - [G3] managing user's profile
  - [G4] requesting a taxi
  - [G5] reserving a taxi
  - [G6] canceling a ride
  - [G7] checking taxi availability around user
  - [G8] reporting a problem caused by passenger or taxi driver
  - [G9] confirming/declining a ride(taxi driver)

Overall description

- Domain properties - these conditions hold in the analyzed world:
  - the passenger needs a ride to specific location
  - the details of the ride provided by the passenger are accurate
  - money exchange between the passenger and the taxi driver is made independently from the myTaxiService system
  - distinction between the zones are clearly defined



Overall description

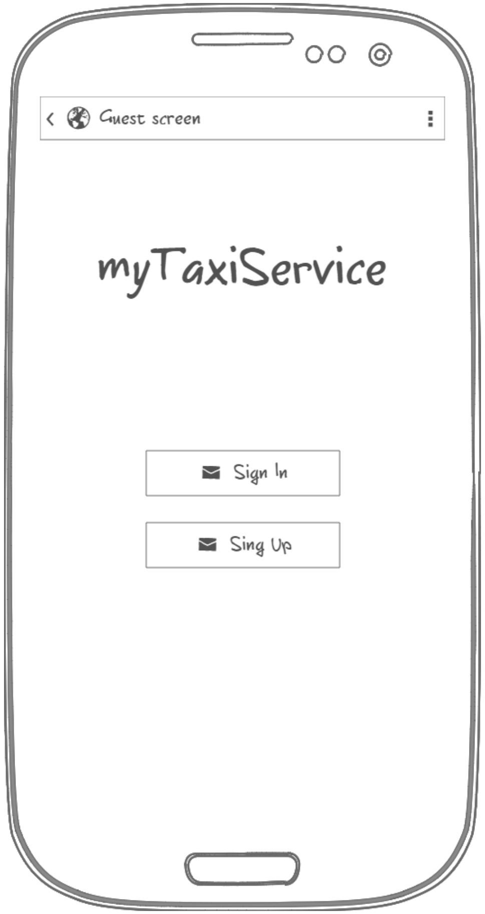
- Constraints

  - **Regulatory policies** - will not take advantage of users personal information and will respect the *privacy policy*
  - **Hardware limitation** - access to *Internet* and own a device with a web browser and *GPS service*
  - **Interfaces to other applications** - Google Maps API, Google Places API and email service in order to make authentication
  - **Parallel operation** - support parallel access to the applications database in a *transparent way*

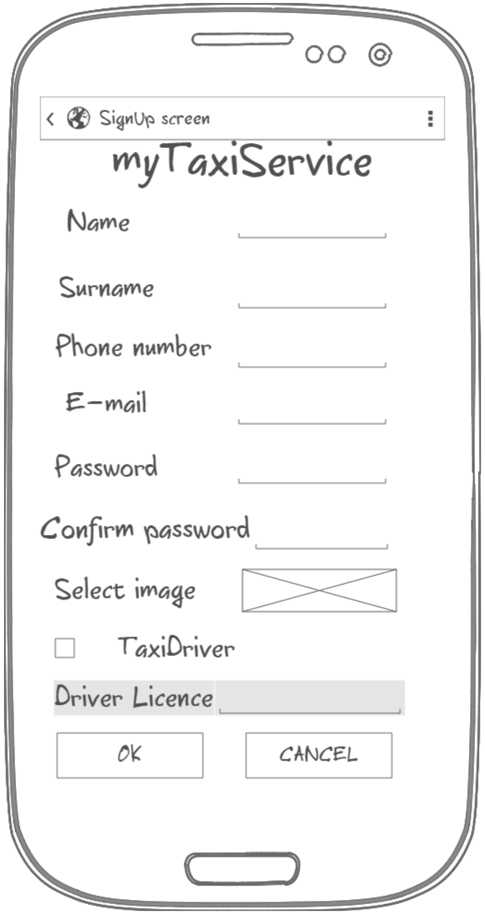
- Assumptions

  - user have only *one account*
  - user provides *accurate information*
  - if users location is not available, the application will show a screen with an option of *typing your current address*
  - we assume that Google Maps service will *calculate location accurately*
  - taxi driver will *respect the ETA*, otherwise they could be banned from the system
  - if a taxi driver has an unexpected issue, the user will be automatically notified by the system and a new vehicle will be assigned to him with new ETA

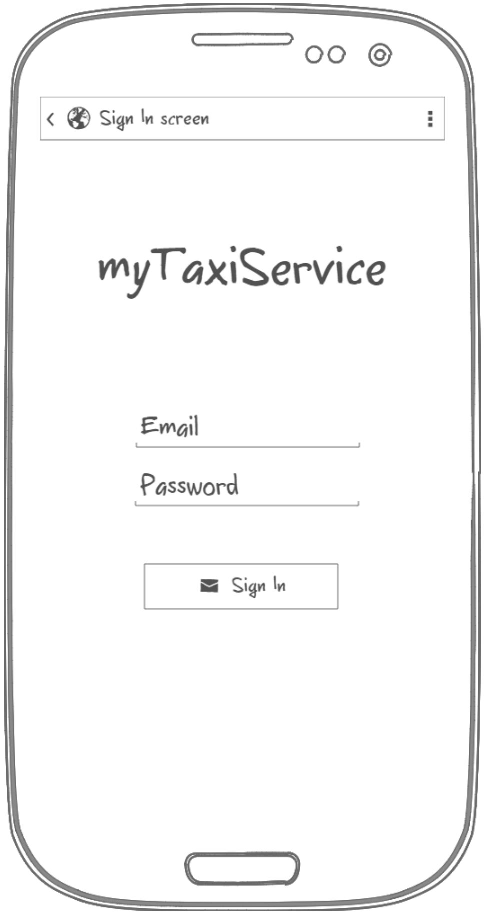
Specific requirements - Mockups



Initial page

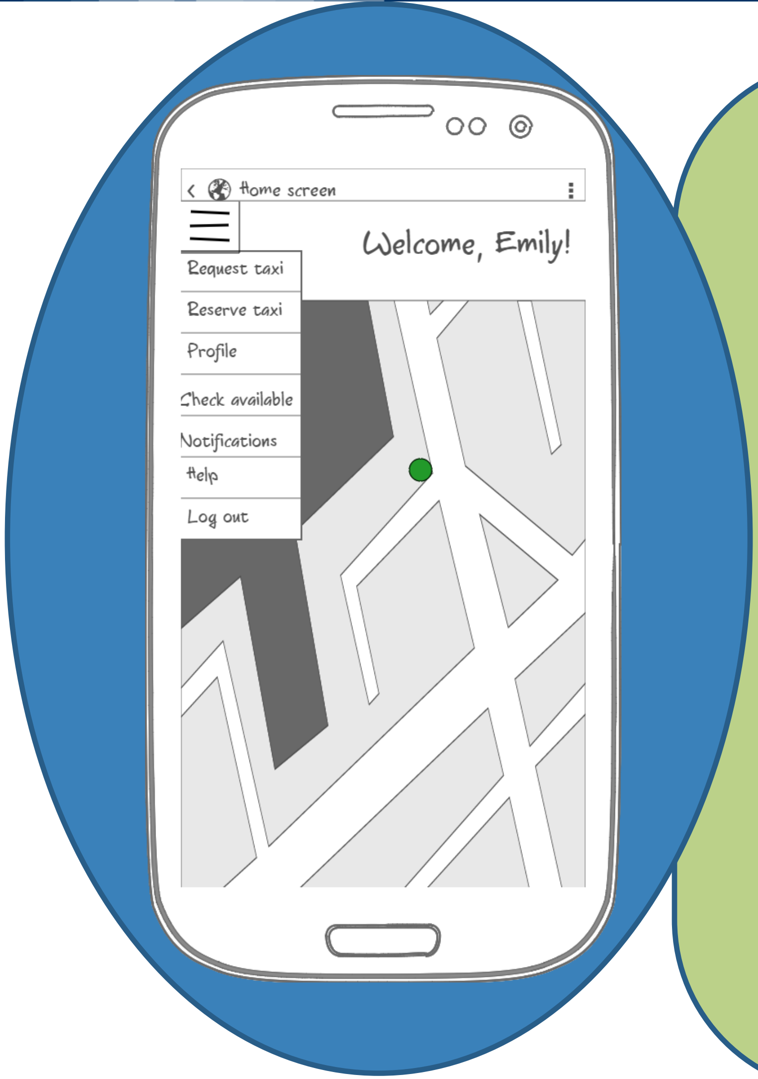


Sign Up

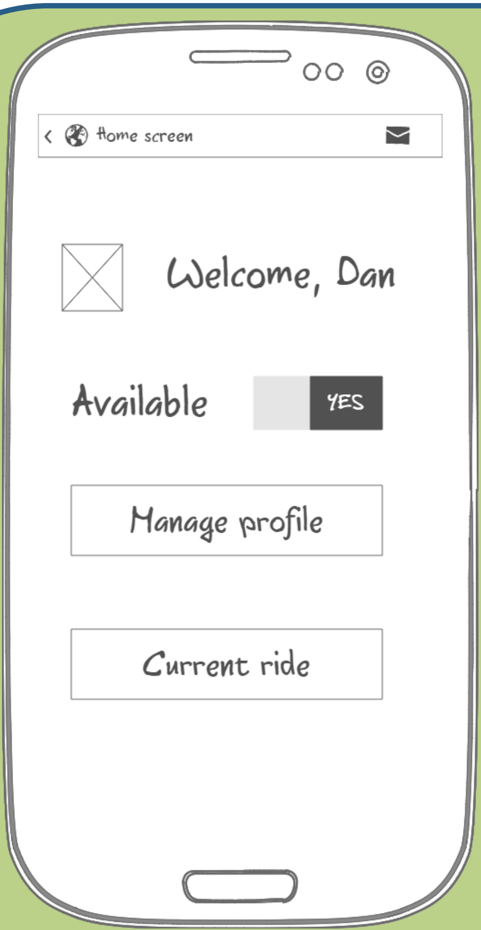


Sign In

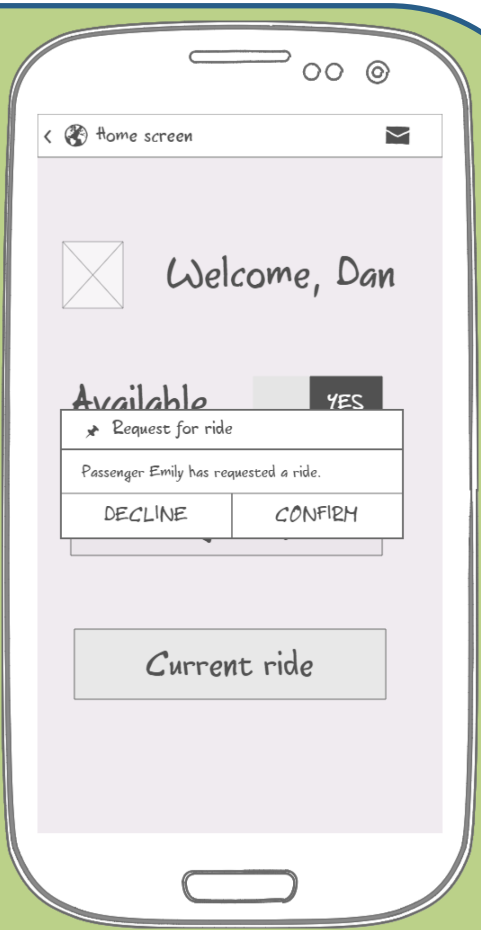
Specific requirements - Mockups



Passenger Home page



Driver Home page

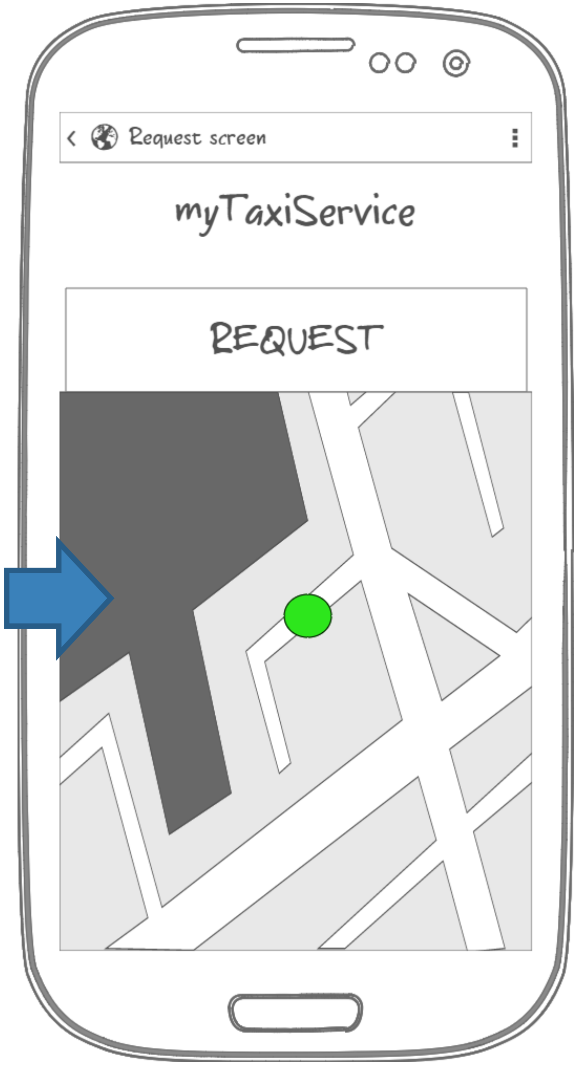


Pop up message

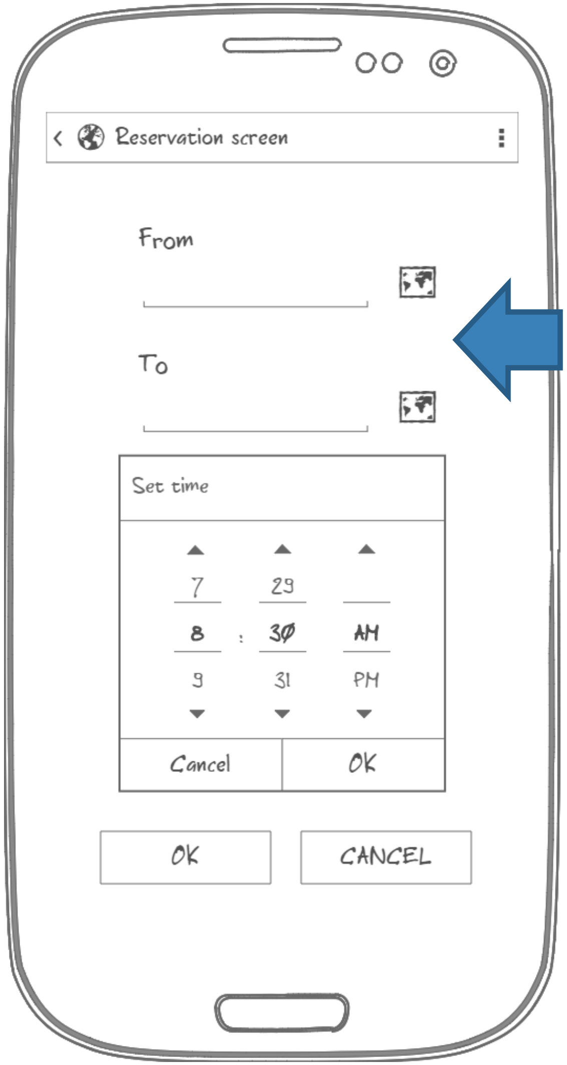
Driver application

Specific requirements - Mockups

User specifies address by clicking on the map



Passenger Request page

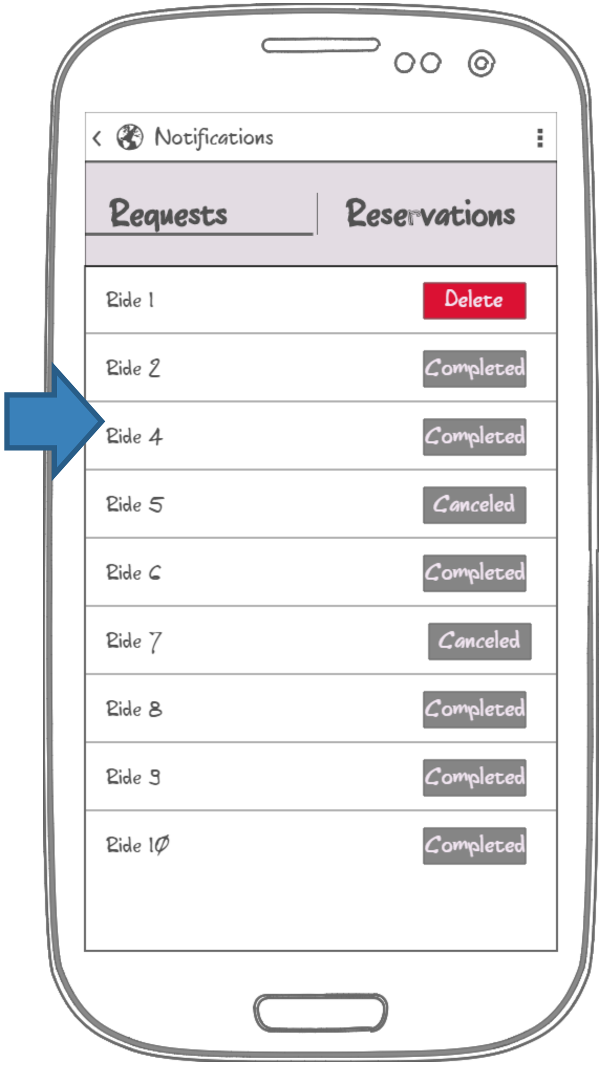


Passenger Reservation page

User specify location either by typing address or by choosing it on the map

Specific requirements - Mockups

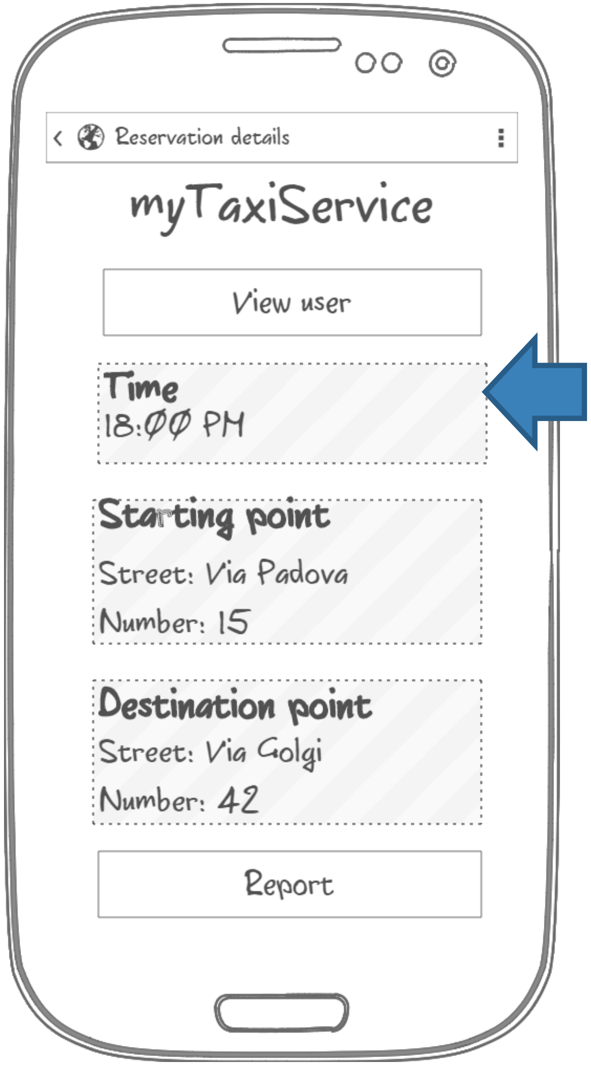
User rides with an option of canceling active ones



Notifications

Requests	Reservations
Ride 1	Delete
Ride 2	Completed
Ride 4	Completed
Ride 5	Canceled
Ride 6	Completed
Ride 7	Canceled
Ride 8	Completed
Ride 9	Completed
Ride 10	Completed

Notification page



Reservation details

myTaxiService

View user

Time
18:00 PM

Starting point
Street: Via Padova
Number: 15

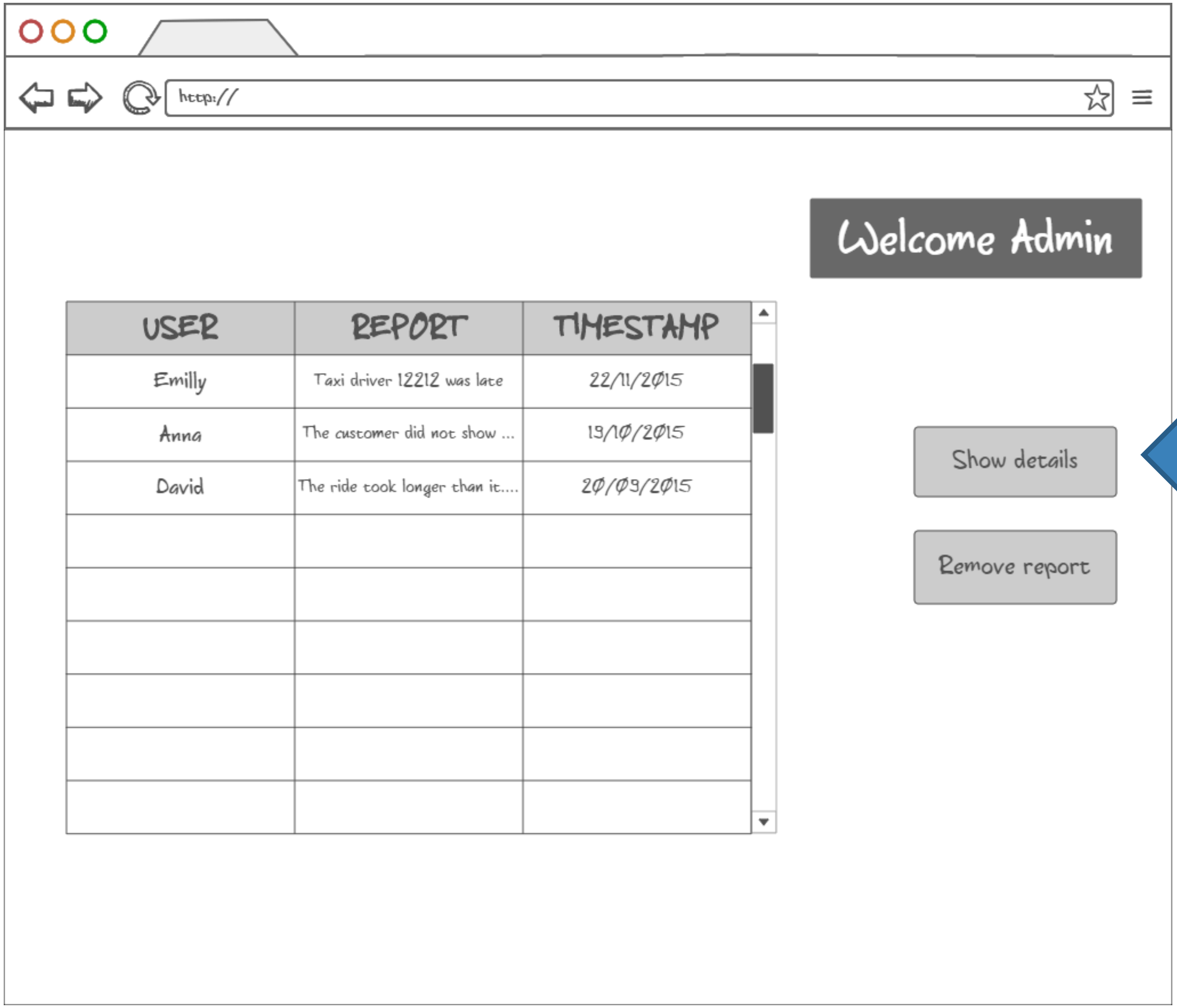
Destination point
Street: Via Golgi
Number: 42

Report

Reservation details

Reservation Details page with data

Specific requirements - Mockups



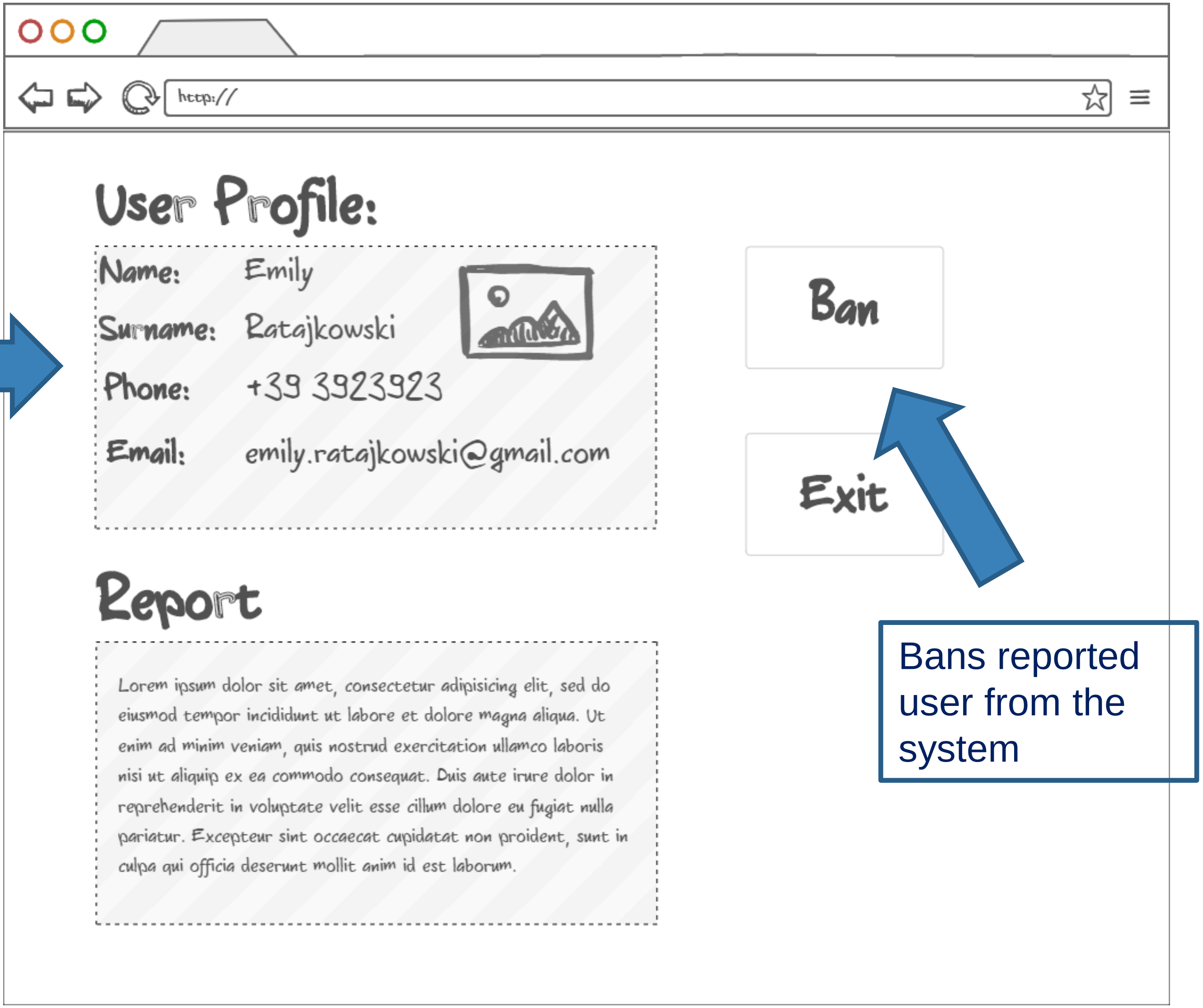
USER	REPORT	TIMESTAMP
Emilly	Taxi driver 12212 was late	22/11/2015
Anna	The customer did not show ...	13/10/2015
David	The ride took longer than it....	20/03/2015

Show details action brings Admin to Report details page

Admin home page

Specific requirements - Mockups

Information about the user that made the repot



User Profile:

Name:	Emily
Surname:	Ratajkowski
Phone:	+39 3923923
Email:	emily.ratajkowski@gmail.com

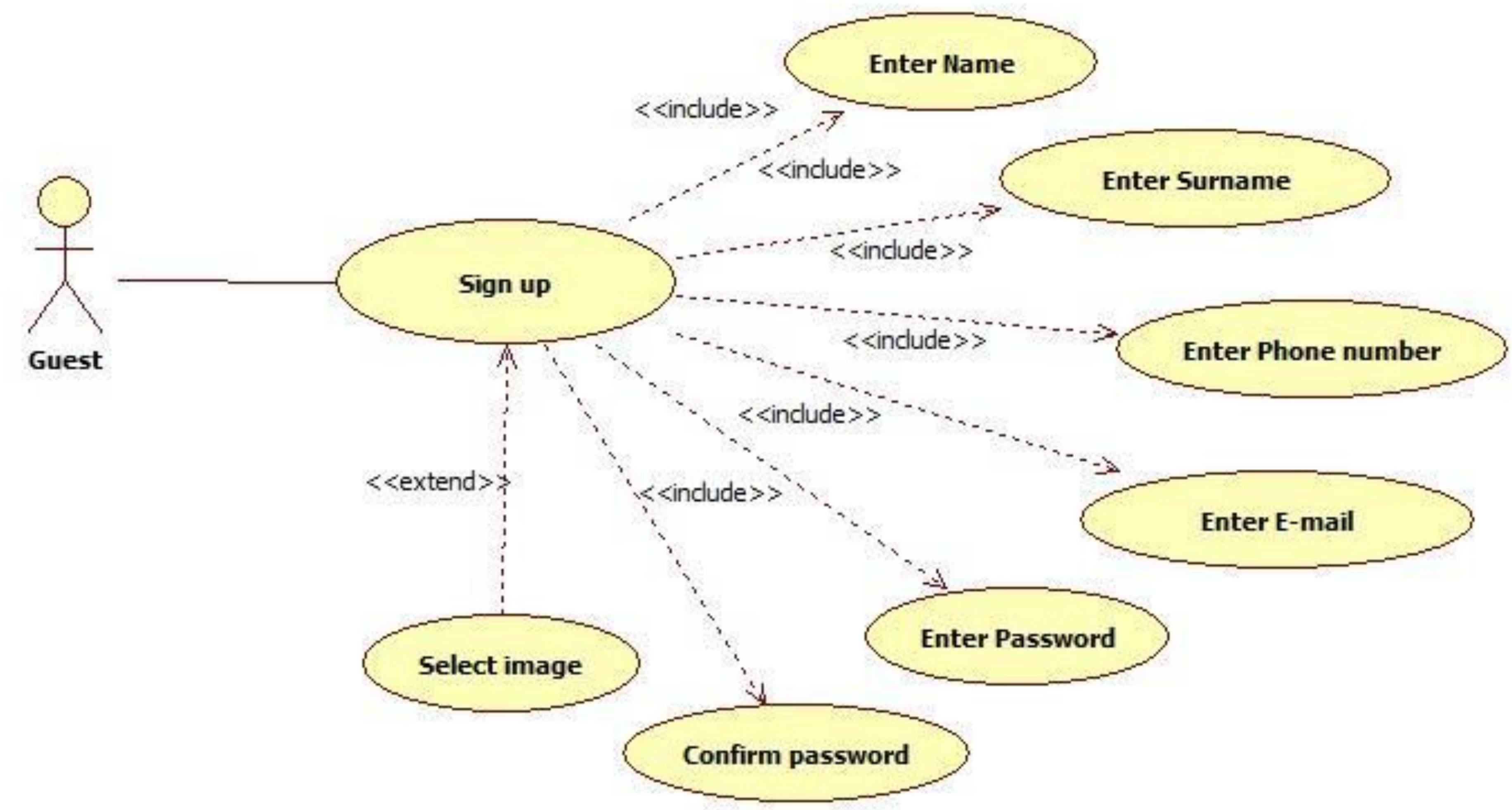
Report

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Bans reported user from the system

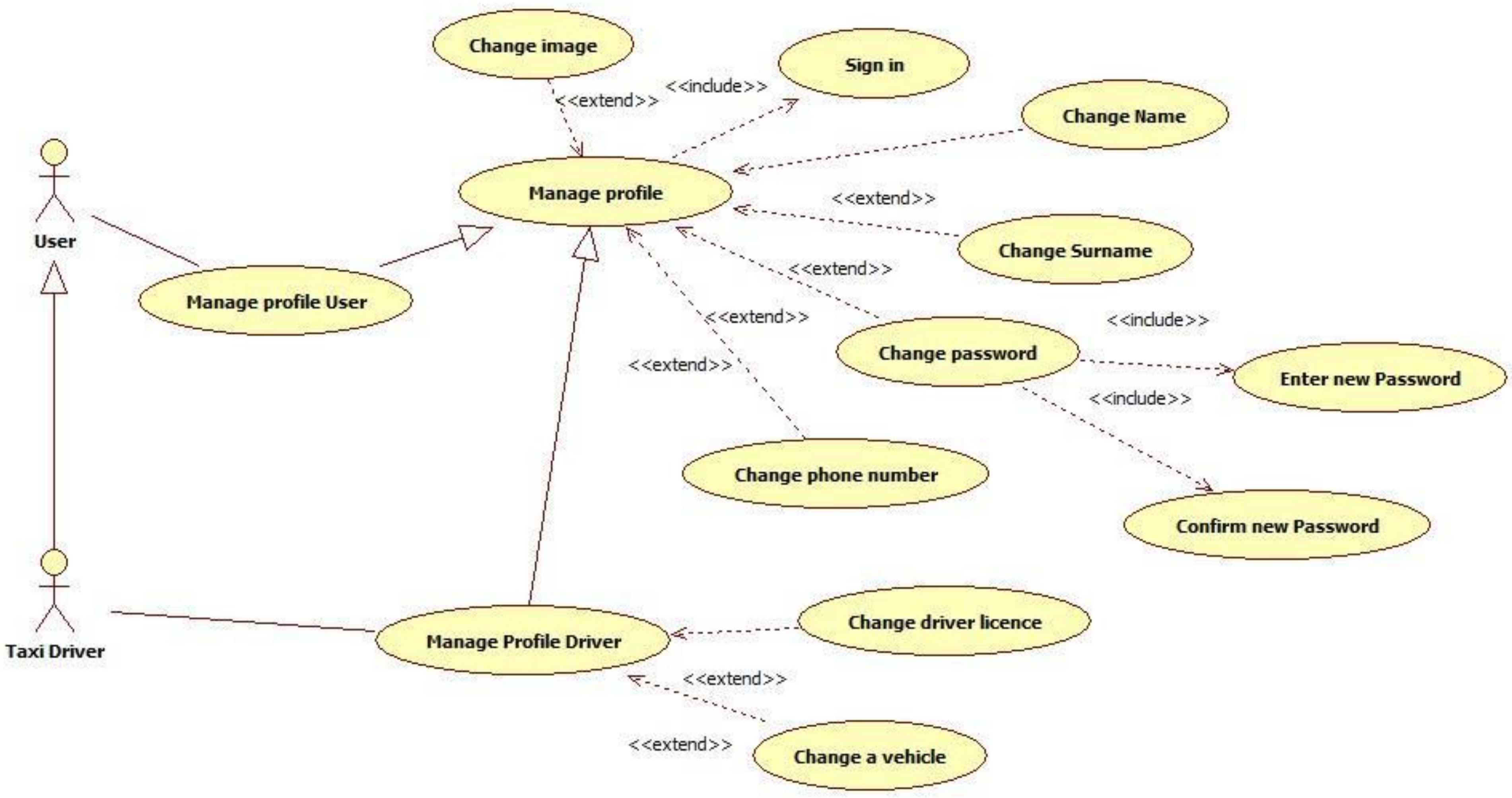
Admin report details page

Specific requirements - Use cases



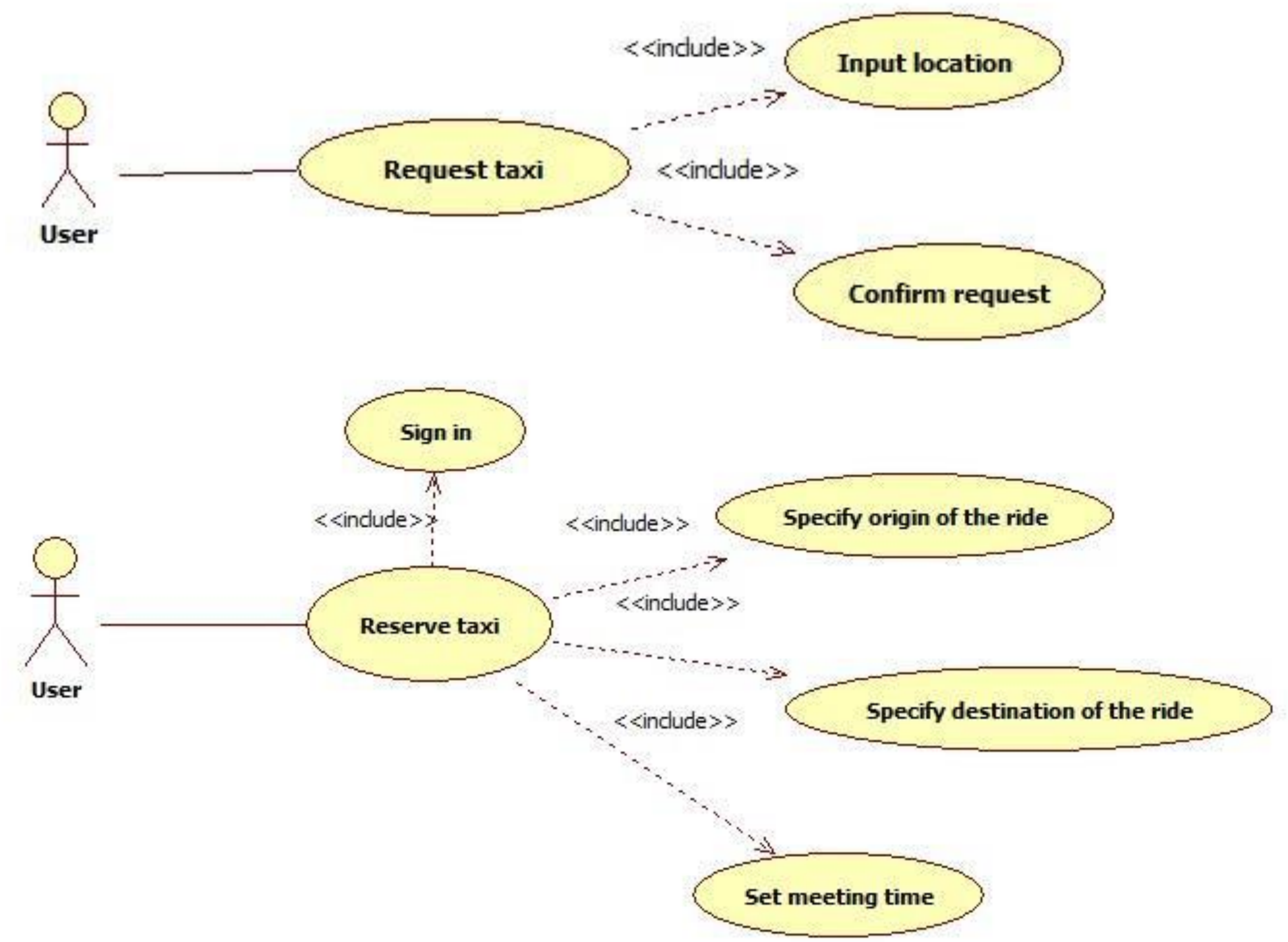
Sign Up use case

Specific requirements - Use cases



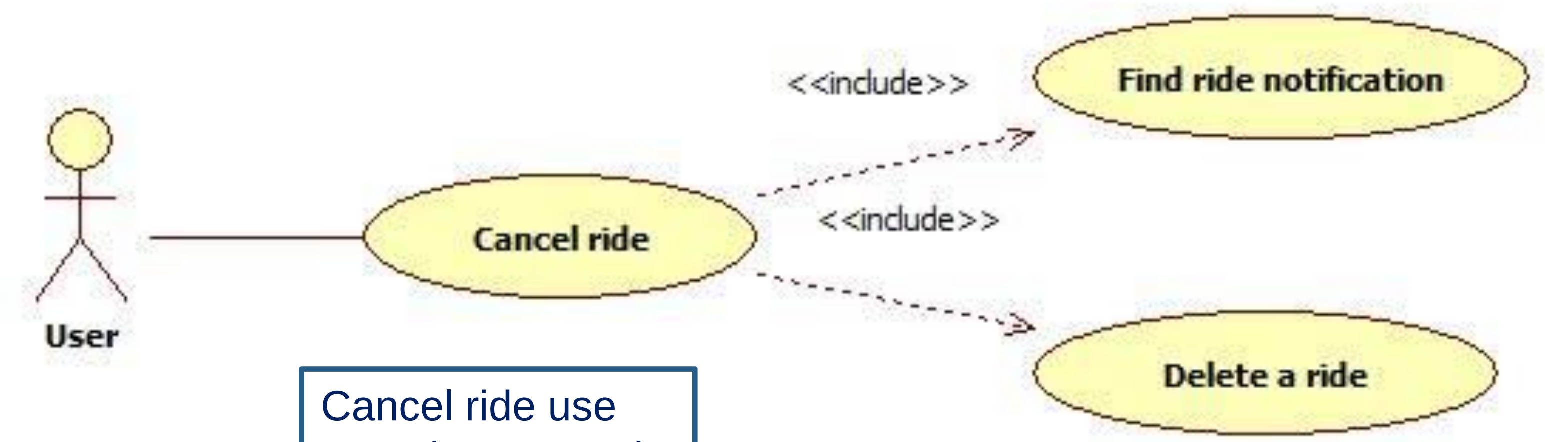
Manage profile use case

Specific requirements - Use cases

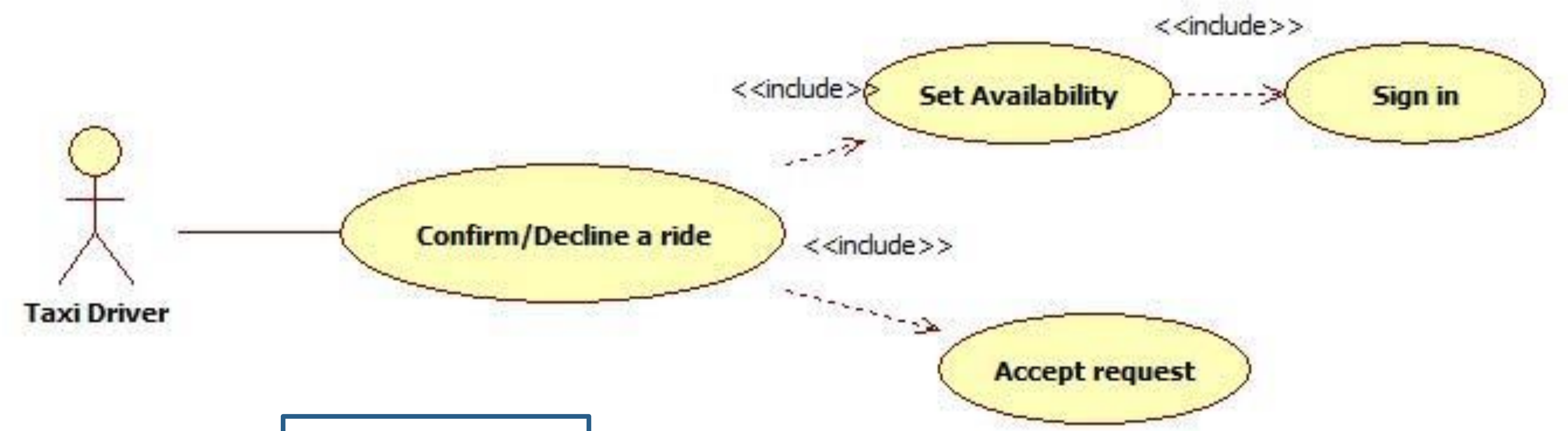


Request / Reserve a taxi use case

Specific requirements - Use cases

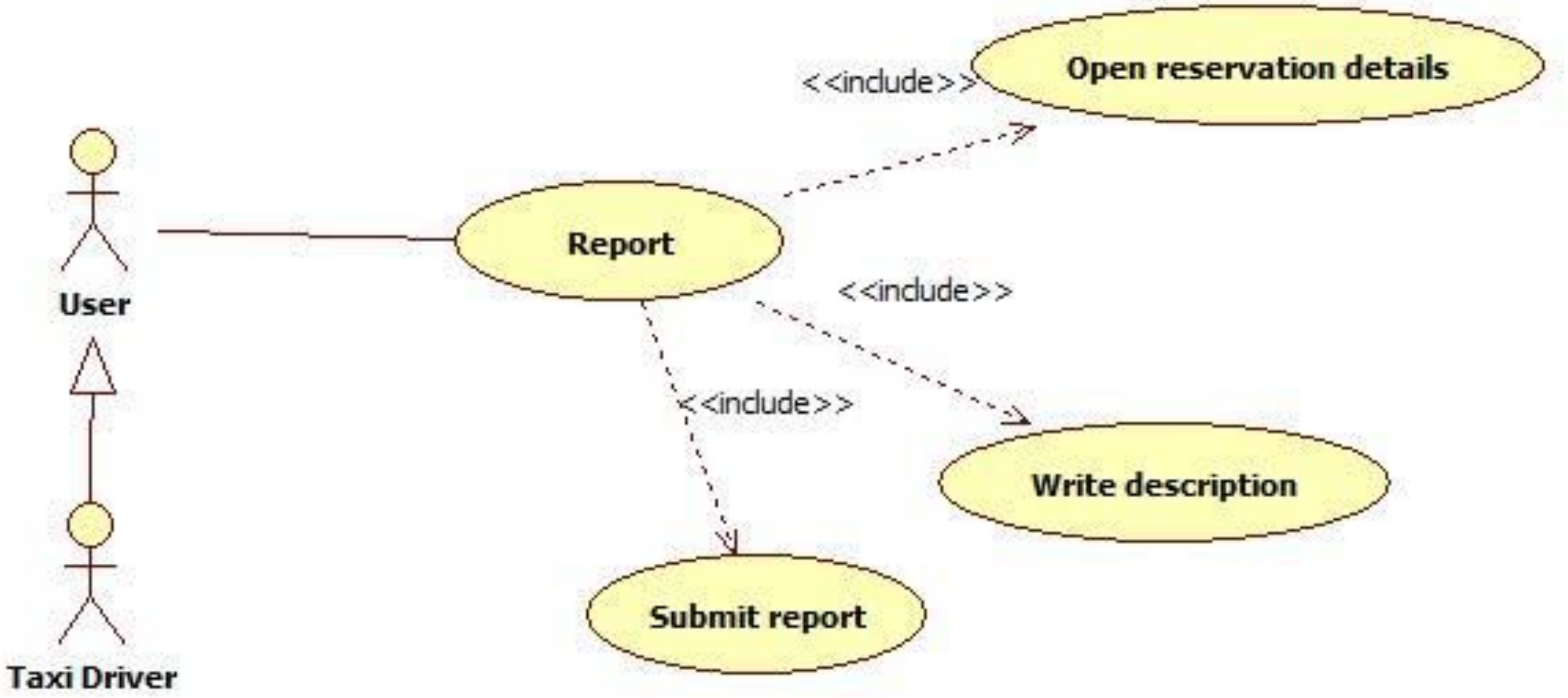


Cancel ride use case (passenger)



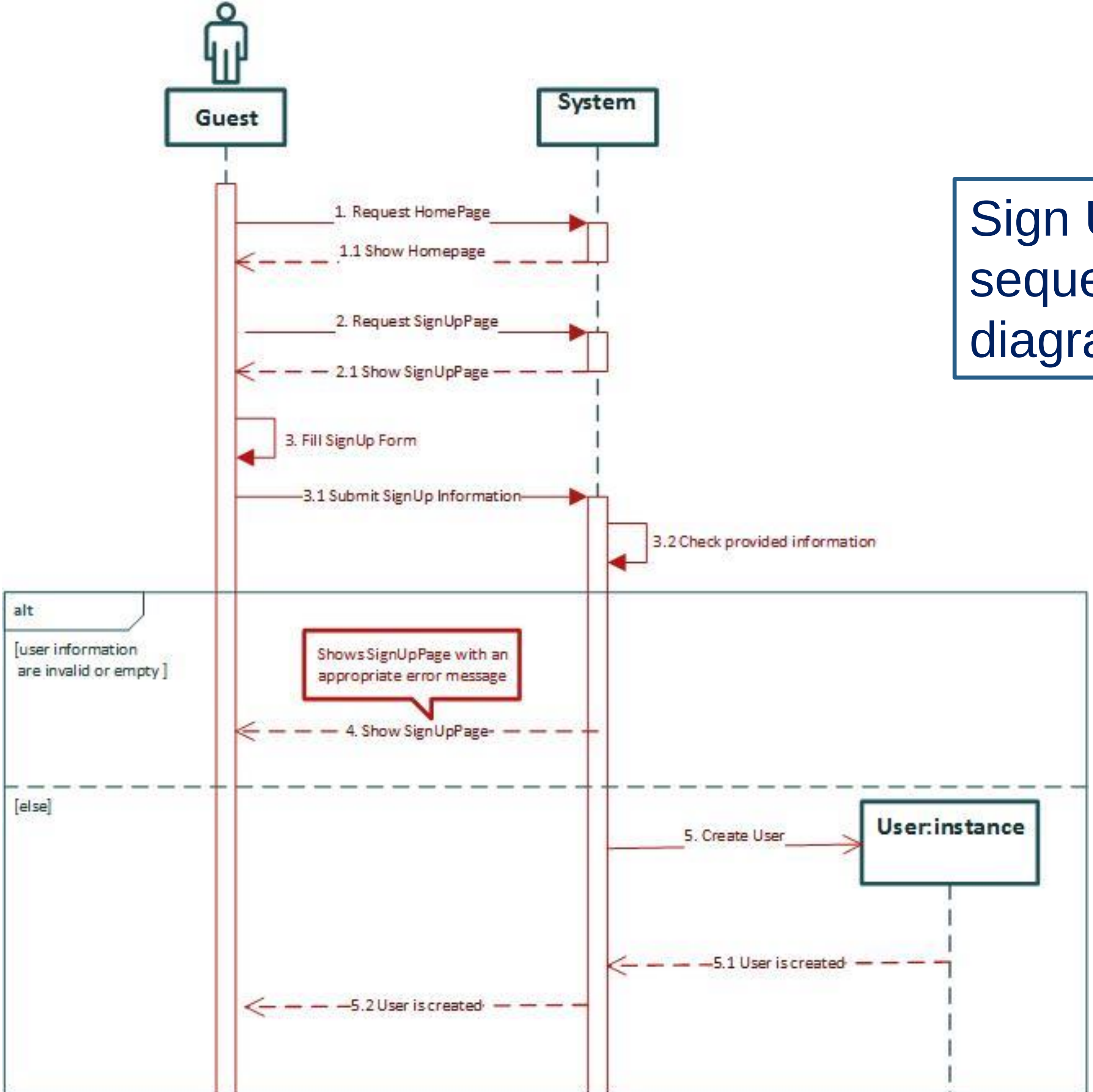
Confirm/Decline a ride use case

Specific requirements - Use cases

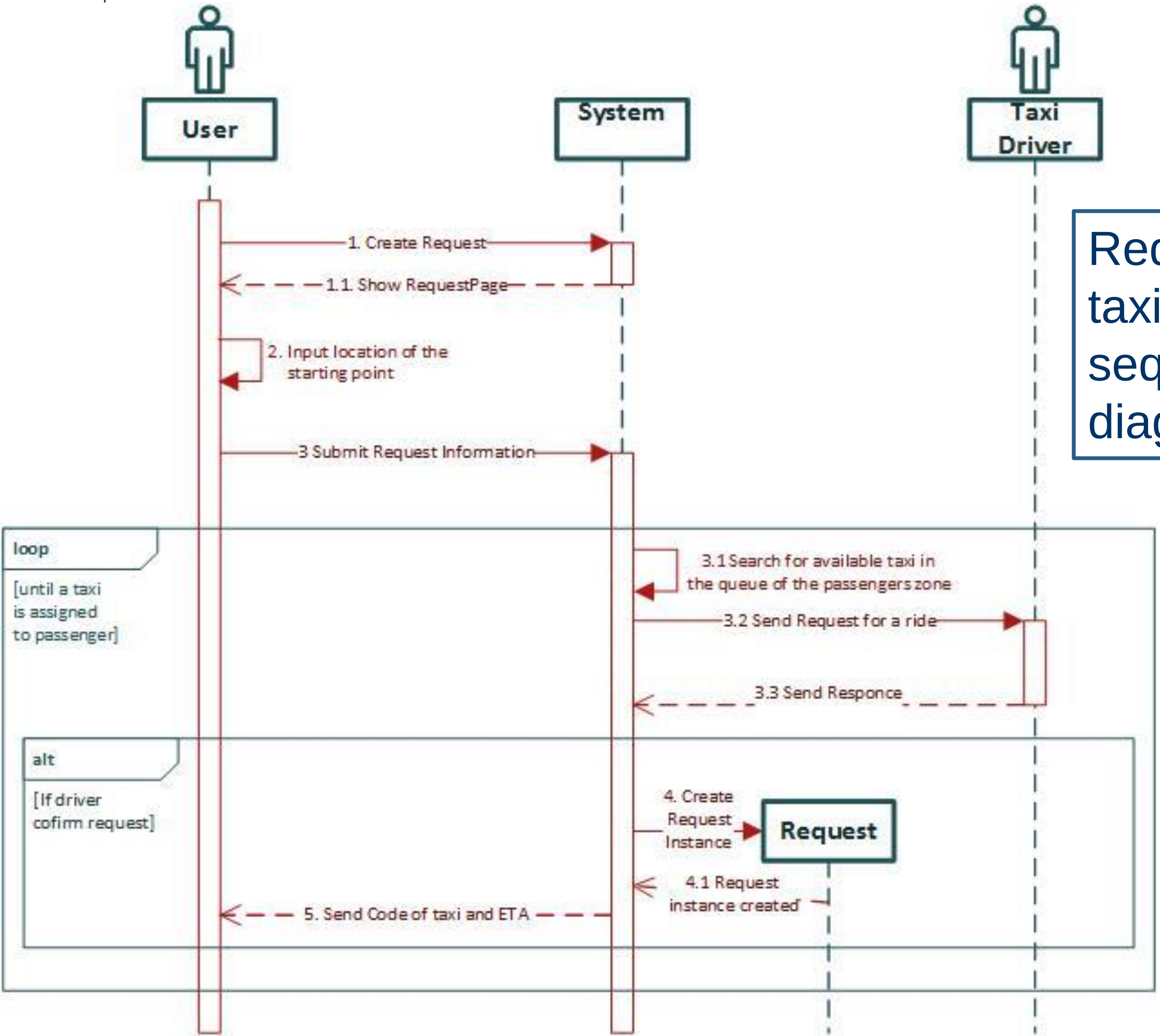


Report use case

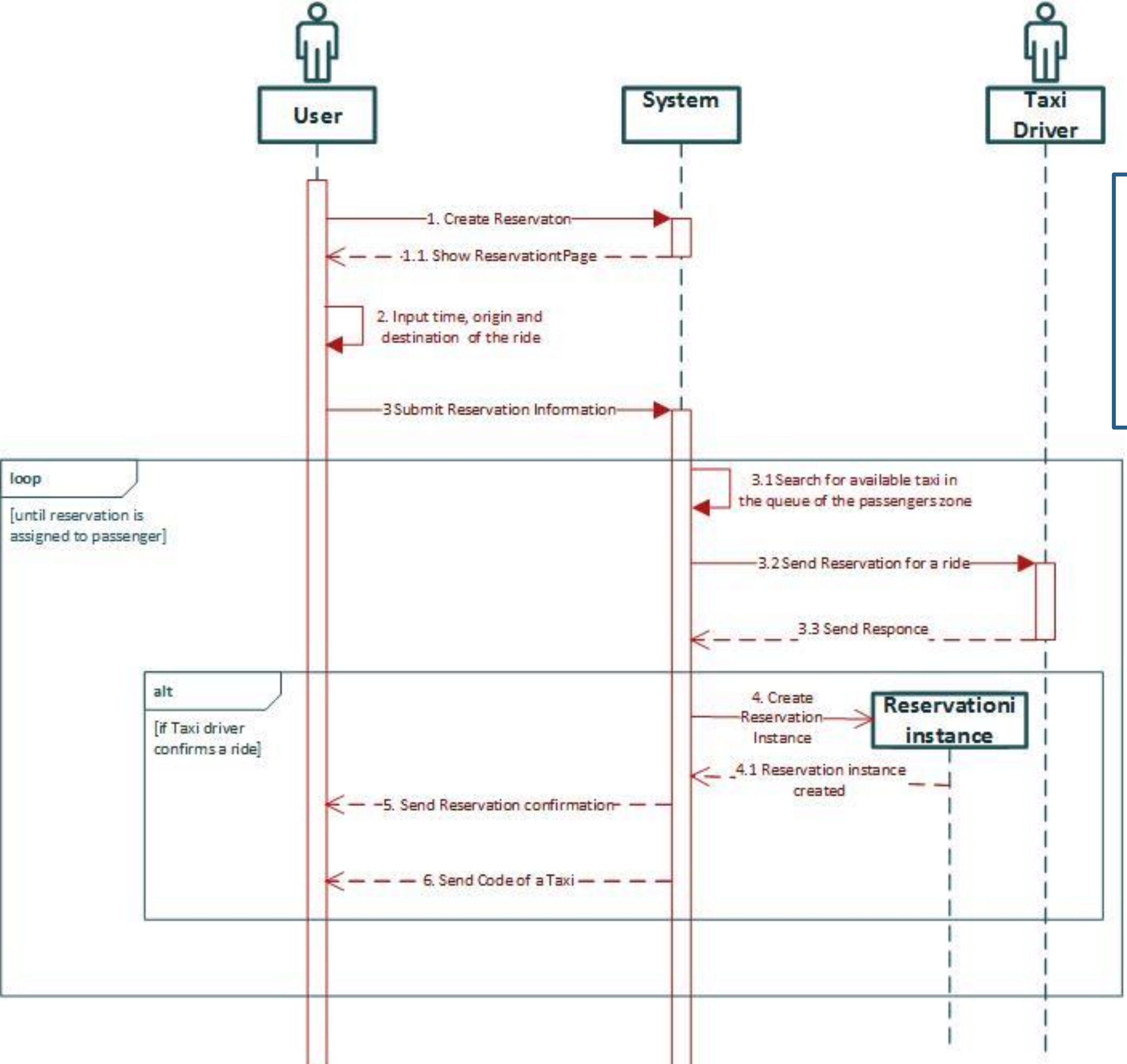
Sign Up sequence diagram



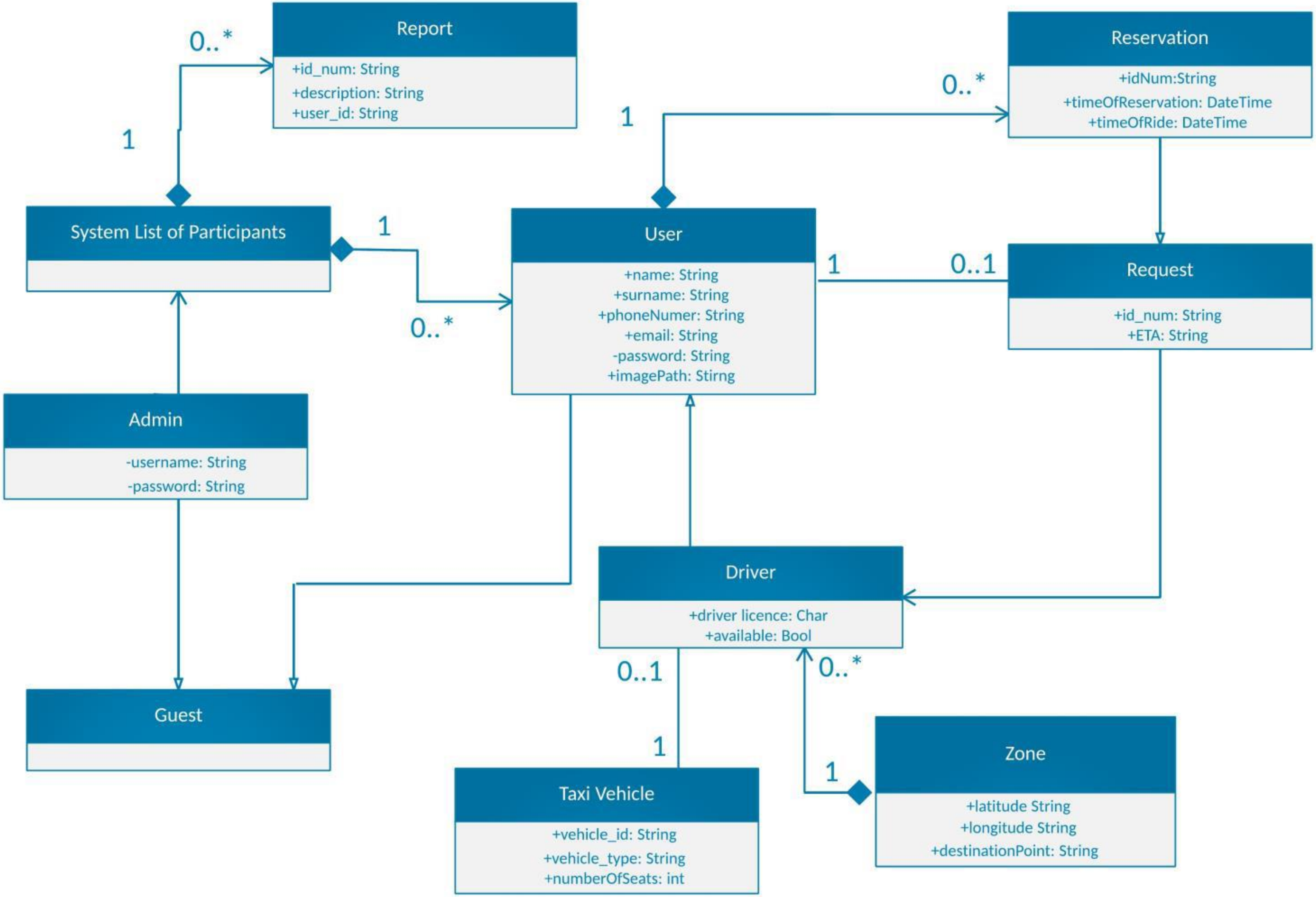
Request a taxi sequence diagram


User
System
Taxi Driver
1. Create Request
1.1. Show RequestPage
2. Input location of the starting point
3 Submit Request Information
loop
[until a taxi is assigned to passenger]
3.1 Search for available taxi in the queue of the passengers zone
3.2 Send Request for a ride
3.3 Send Responce
alt
[If driver cofirm request]
4. Create Request Instance
Request
4.1 Request instance created
5. Send Code of taxi and ETA

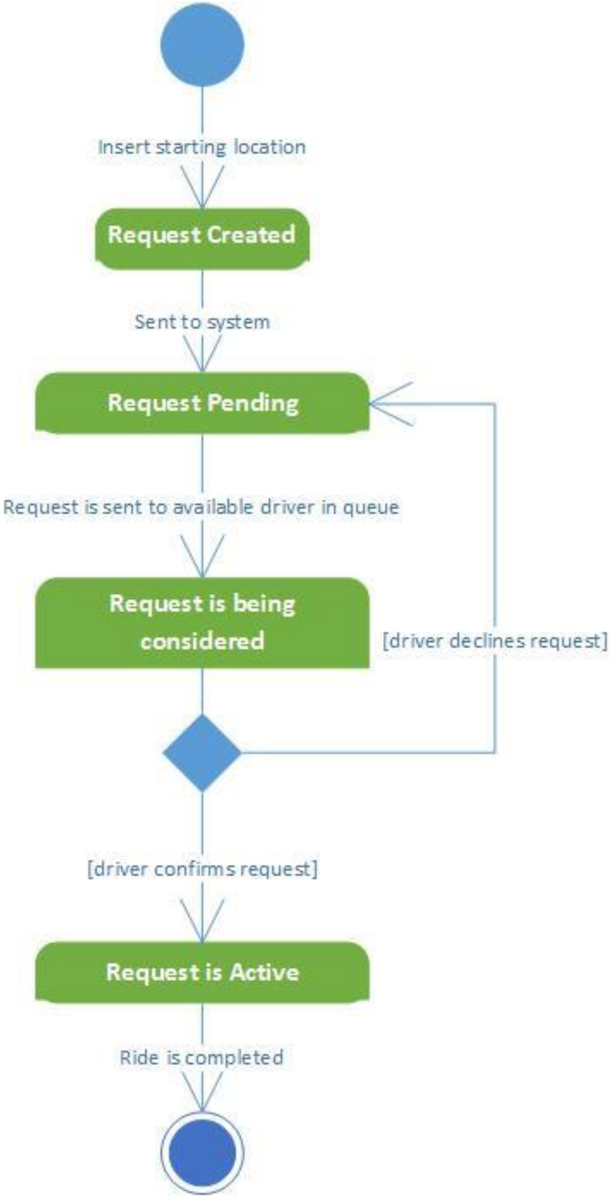
Reserve a taxi sequence diagram


User
System
Taxi Driver
1. Create Reservaton
1.1. Show ReservationtPage
2. Input time, origin and destination of the ride
3 Submit Reservation Information
loop
[until reservation is assigned to passenger]
3.1 Search for available taxi in the queue of the passengers zone
3.2 Send Reservation for a ride
3.3 Send Responce
alt
[if Taxi driver confirms a ride]
4. Create Reservation Instance
Reservationi instance
4.1 Reservation instance created
5. Send Reservation confirmation
6. Send Code of a Taxi

Specific requirements – Class diagram



Specific requirements – State Chart diagrams


Insert starting location
Request Created
Sent to system
Request Pending
Request is sent to available driver in queue
Request is being considered
[driver declines request]
[driver confirms request]
Request is Active
Ride is completed

Request lifecycle

Specific requirements – Non functional requirements

- Availability

  -Application should be available to handle user's request at all times using any device with an installed web browser

- Maintainability

  - The software system provide specific API for enabling future developers with option to add more services or fix bugs in the system

- Portability

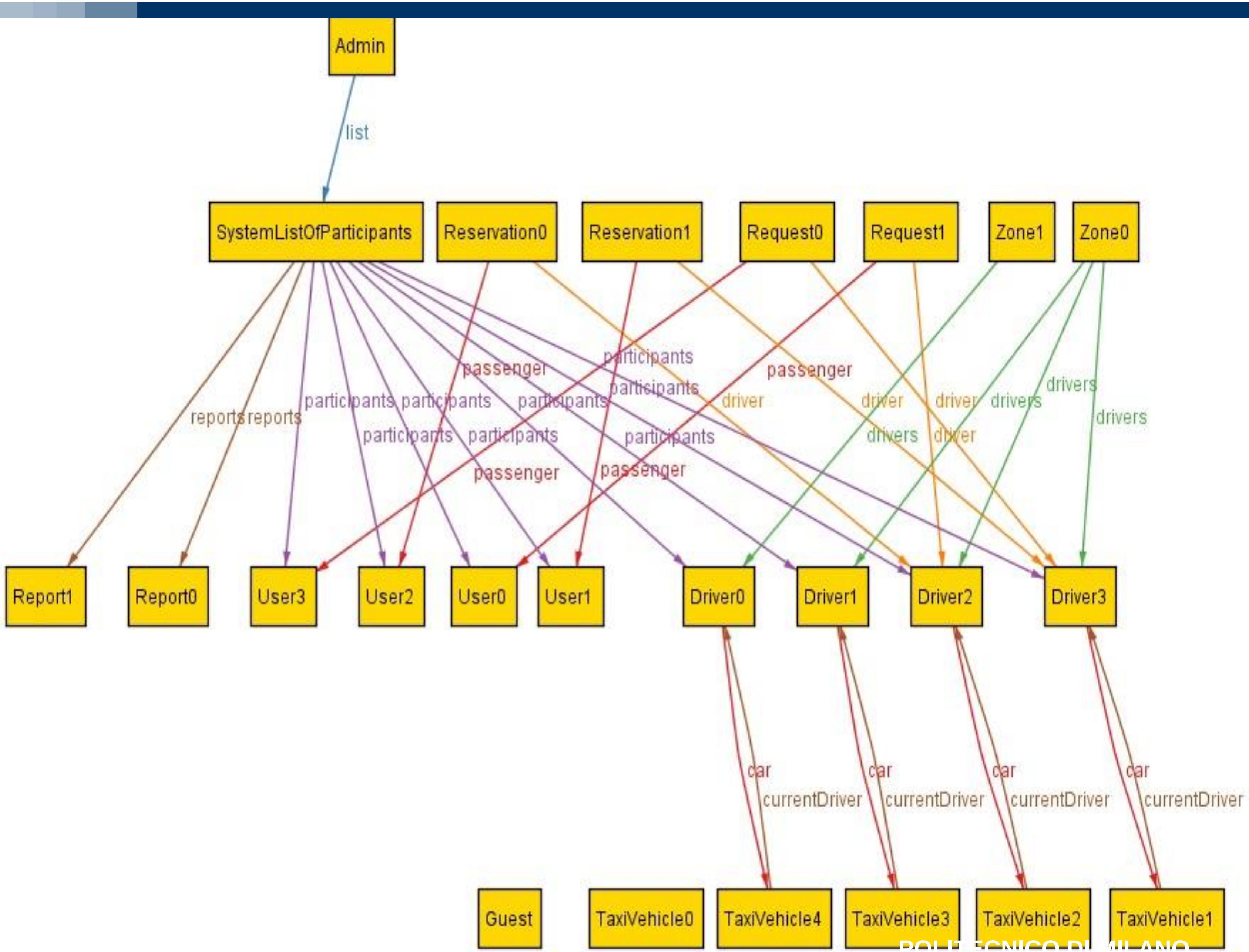
  - Application could be run on device with any OS that has access to Internet and has a web browser

- User Interface

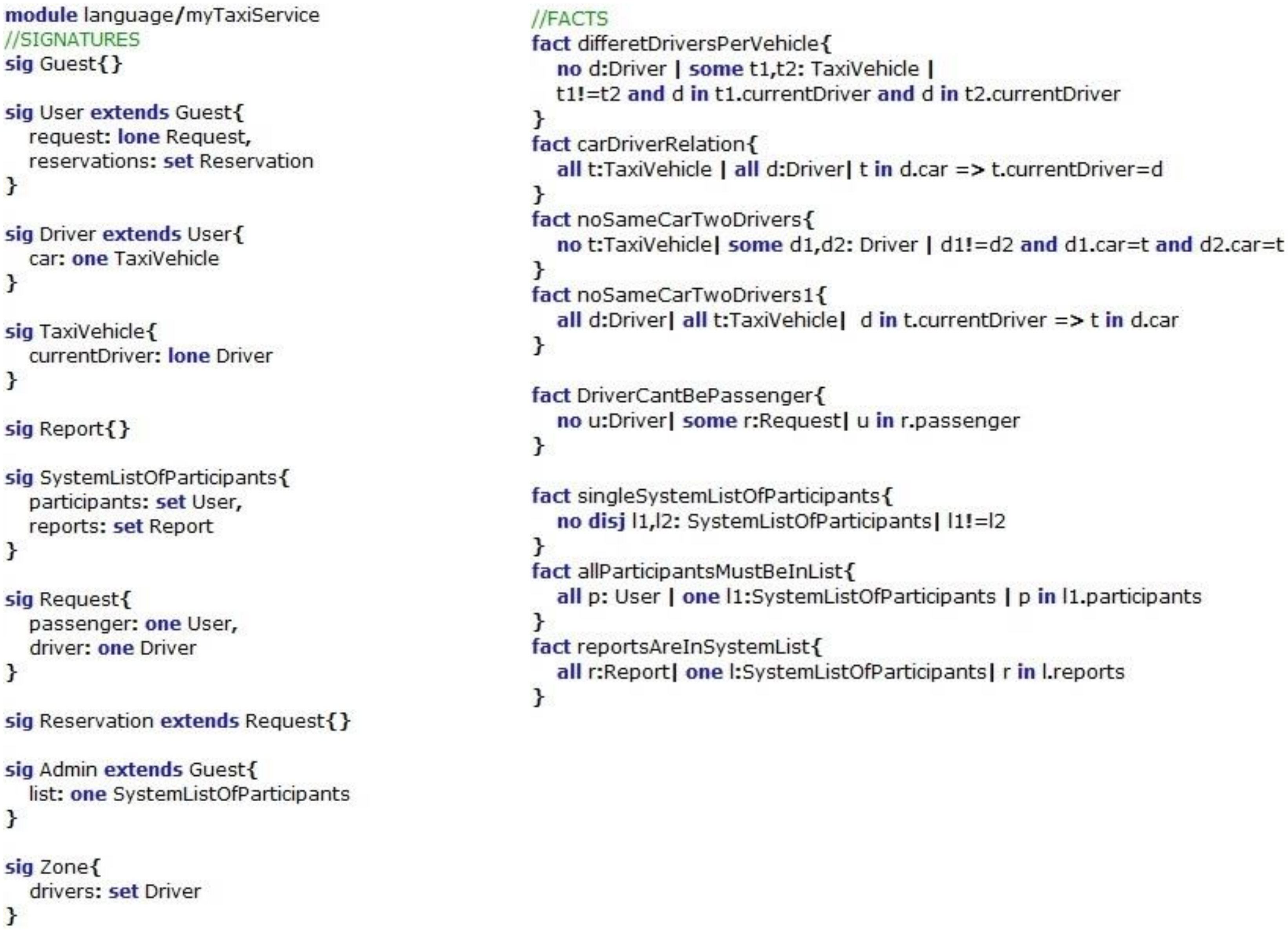
  - Web application should be intuitive so even the nontechnical users can use the system as simply and efficiently as possible

Alloy- World generated by Alloy analyzer

car: 4
currentDriver: 4
driver: 4
drivers: 4
list: 1
participants: 8
passenger: 4
reports: 2



Alloy- Prove the model's consistency



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

sig User extends Guest{
   request: lone Request,
   reservations: set Reservation
}

sig Driver extends User{
   car: one TaxiVehicle
}

sig TaxiVehicle{
   currentDriver: lone Driver
}

sig Report{}

sig SystemListOfParticipants{
   participants: set User,
   reports: set Report
}

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

sig Reservation extends Request{}

sig Admin extends Guest{
   list: one SystemListOfParticipants
}

sig Zone{
   drivers: set Driver
}
```

```
//FACTS
fact differetDriversPerVehicle{
   no d:Driver | some t1,t2: TaxiVehicle |
   t1!=t2 and d in t1.currentDriver and d in t2.currentDriver
}
fact carDriverRelation{
   all t:TaxiVehicle | all d:Driver| t in d.car => t.currentDriver=d
}
fact noSameCarTwoDrivers{
   no t:TaxiVehicle| some d1,d2: Driver | d1!=d2 and d1.car=t and d2.car=t
}
fact noSameCarTwoDrivers1{
   all d:Driver| all t:TaxiVehicle|  d in t.currentDriver => t in d.car
}

fact DriverCantBePassenger{
   no u:Driver| some r:Request| u in r.passenger
}

fact singleSystemListOfParticipants{
   no disj l1,l2: SystemListOfParticipants| l1!=l2
}
fact allParticipantsMustBeInList{
   all p: User | one l1:SystemListOfParticipants | p in l1.participants
}
fact reportsAreInSystemList{
   all r:Report| one l:SystemListOfParticipants| r in l.reports
}
```

Future possible implementation

- Online payment
- Option of rating the drivers
- Facebook authentication
- Taxi sharing options

Questions?

Thank you for your attention