



POLITECNICO
MILANO 1863

Release date 26-02-2016

Software Engineering 2: MyTaxiService

Requirements Analysis and Specifications Document V2.0

Dimitar Anastasovski, Marco Colombo

Contents

1	Introduction	2
1.1	Purpose	2
1.1.1	Intended audience	2
1.2	Scope	3
1.3	Goals	3
1.4	Abbreviations	3
1.5	Acronymous	4
1.6	Document overview	4
2	Overall description	5
2.1	Product perspective	5
2.2	The world and the machine	5
2.3	Product functions	6
2.4	User characteristic	7
2.5	Constraints	7
2.6	Assumptions and dependencies	8
3	Requirements	10
3.1	Functional requirements	11
3.2	External interface requirements	12
3.2.1	User interface	12
3.2.2	Software interface	22
3.3	Scenarios	22
3.4	Nonfunctional requirements	23
3.4.1	Reliability	23
3.4.2	Availability	23
3.4.3	Security	23
3.4.4	Usability	23
3.4.5	Maintainability	23
3.4.6	Portability	23
3.4.7	Future implementation	23

4	UML Models	24
4.1	Use Case Models	24
4.2	Class Diagram	31
4.3	Sequence Diagrams	32
4.4	Activity diagrams	35
5	Alloy	37
5.1	Signatures	37
5.2	Facts	38
5.3	Predicates	38
5.4	Results	38
5.5	Generated world	39
6	Appendix	40
6.1	Software and used Tools	40
6.2	Working hours	40
6.3	Changelog	40

Chapter 1

Introduction

1.1 Purpose

This document is intended to help understand and communication of the requirements of the system, explaining both the application domain and the system that you want to accomplish. It explains the functional features of the MyTaxiService, along with interface details, design constraints and related considerations such as performance characteristics. It can be considered as a contract between the developer and the customer. It is the basis for the planning of the project and to estimate its duration and its cost. It is the base for the activities of testing, verification and validation. The RASD should in fact contain sufficient information to verify if the final system actually satisfies the requirements contained in the document itself. This document follows the IEEE standard for software requirements specification documents.

1.1.1 Intended audience

This document is intended for all individuals participating in and/or supervising the project:

- Expected audience of this document is the developers and anyone who intends to develop on this program
- Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (it sets the guidelines for future development).
- Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.

- End users of this application who wish to read about what this project can do.

1.2 Scope

The main accent is to simplify and optimize the access of passengers to the system and to guarantee fair management of taxi queues. We will build flexible and user-friendly web application and a mobile application that will run on Android and IOS mobile phones. This application can be used by anyone who previously will be register on the registration page. After the registration is done the user will have a user name and password that should remember for furthermore usage of the system. The passenger can call a taxi after a successful logging on the application. After that he can call a taxi and he will be informed about the code of the incoming taxi, waiting time. On the other hand taxi drivers will have a mobile application where the major purpose will be to inform the system about their availability, confirmation of a certain call and global map navigation. City is divided into taxi zones that are uniquely associated with corresponding taxi queues for efficient usage of the system.

1.3 Goals

We think that the system has to provide this features:

- [G1] Only registered users can use the service - high priority
- [G2] Registered users can use service if and only if they are logged in - high priority
- [G3] Only eligible users approved by administration can become taxi drivers - high priority
- [G4] Users can request taxi - high priority
- [G5] Users can reserve taxi - high priority
- [G6] Users can cancel a ride - high priority
- [G7] Checking taxi availability - high priority
- [G9] Confirming/declining a ride (taxi driver) - high priority
- [G10] Taxi driver's availability can be changed - high priority

1.4 Abbreviations

- ETA: Estimated Time of Arrival, approximated time of arrival of taxi vehicle to destination
- Reservation: Passenger request for a vehicle at least 2 hours before ride

- Request: Passenger filled form for immediate ride
- Reservation Confirmation: Notification sent to the user about the confirmed ride with informations about it
- Ride Confirmation: Notification sent to the user about the confirmed ride with informations about it
- User: A person already registered and logged into the application
- Guest: A person accessing a system that has either never registered or has not logged in yet. Guest has only two available options: log in or register for the first time.
- Taxi Driver: A person already register and logged into the system as a driver
- GPS: Global Positioning System
- API: Application Programming Interface

1.5 Acronymous

- IEEE Standard 830-1998 Recommender Practice for Software Requirements Specifications
- Specification Document: myTaxiService Project

1.6 Document overview

This document is divided in four parts with clean and non-ambiguity description of the whole system.

- Chapter 1: General description and basic informartion of the system
- Chapter 2: Explanation of the main functionalities with constraints, assumptions and hardware dependencies.
- Chapter 3: Specifications of the system functionalities and non-system functionalities possible scenarios
- Chapter 4: Use cases and UML models
- Chapter 5: Alloy modelling and generated world

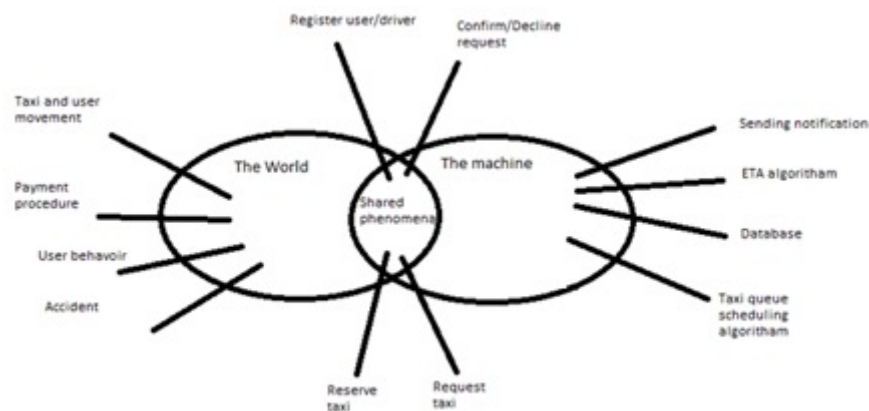
Chapter 2

Overall description

2.1 Product perspective

MyTaxiService is a mobile application (Android and iOS) and web application that require users to have access to a web browser on their smartphone or personal computer. The concept is based on the idea of establishing a direct connection between drivers and passengers to offer both sides a modern alternative to conventional booking processes. After research on the market we will provide unique system for booking a taxi that will benefit customers with cheap fast and satisfying usage of the system.

2.2 The world and the machine



The world - these activities are initiated by the user or driver or by their environment

- Taxi movement - happens in user environment
- Payment - happens between user and driver without accessing the application
- Accident - can be caused by another vehicles in traffic user environment
- User behavior - can change the actual state

The machine - these activities are used in the system, transparent to users

- Sending notifications - those actions are initiated by the system, does not include user actions
- ETA algorithm - is performed by the system
- Taxi queue and schedule algorithm - is used to efficiently find available taxi vehicle from queue of user's current zone
- Database - is located in the server side where all datas are stored

Shared - these activities require both user and system actions to be completed

- Request taxi - first, user has to specify origin of the ride and send to system, then system needs to find available taxi driver to complete request
- Reserve taxi - user has to specify meeting time, origin and destination of the ride, then the system needs to find a taxi vehicle. After that, reservation is completed.
- Register user/driver - requires user to fill the form that will be sent to the system, which creates user's profile and send back the confirmation message.
- Confirm decline request - involves user and system. System sends proposal of the ride and then user can choose whether to confirm or decline a ride

2.3 Product functions

- **Registration:** a guest user can register as passenger: the user will be able to insert his information such as: first name, last name, e-Mail, phone number, credit card and date of birth.

- **Request Taxi:** after the request, the system will report the amount of the trip to the passenger and he can proceed with the payment. After that according to the informations provided by the passenger, the system will have to alert the closest driver in the queue of the desired zone.
- **Reserve taxi:** passengers can reserve a taxi at least 2 hours before the desired time. After the reservation, a taxi will be allocated to the request 10 minutes before the meeting time.

2.4 User characteristic

We expect massive usage from people who use public transport on a daily basis regardless age, gender and nationality. Expected users must have basic knowledge of using web browser and have access to internet.

- **Guest:** a person that has not registered and can only exploit basic functionalities such as looking for help.
- **User:** a person that has registered and has provided his personal information and a set of abilities.
 1. Passenger
 2. Taxi driver
- **Administrator:** the responsible for the web application. The only one that can accept new abilities from users.

2.5 Constraints

Major constraints are related to security and reliability of the system and customers. The Service may admit you to link to other websites, services or resources on the Internet, and other websites, services or resources may contain links to the Site. When you access third party websites, you do so at your own risk. These other websites are not under our control.

- Regulatory policies harmonized with Italian regulatory agency.
- Hardware and software limitations

	Requirements
Operating system	Windows(XP or higher), Linux(Edubuntu,Ubuntu or higher), Mac OS (X 10.6 or higher), Android(Froyo or higher), iOS(7.0 or higher)
CPU	1.5Ghz or higher
Memory	512 MB RAM(for PC) 32 MB RAM (for Android and iOS)
Hard drive	2.2 GB free hard disk space
Browser requirements	Internet explorer 6 or higher, Google Chrome, Mozilla Firefox, Apple Safari 5+
GPS	YES
Internet Connection	YES

- Interfaces to other applications:
MyTaxiService doesn't meet with other application
- Parallel operation:
MyTaxiService must support parallel operations
- For testing purposes you will need a simulating tool.

2.6 Assumptions and dependencies

- If you close the application the reservation of the taxi continues, push notification will be send before the taxi arrives.
- Administrator (or the system) has authority to delete account
- Customer can only reserve taxi at least two hours before the ride
- Taxi driver can receive ride request while he is unavailable
- Taxi driver that is currently on ride can't receive requests
- Only available taxi drivers can receive ride requests
- Taxi driver can cancel the reservation
- User can only have one account
- If customer had more than three cancellation, administrator (or the system) can block his account

- Customer can reserves maximum three taxis
- Customers can cancel reservation, but the system must inform the taxi driver
- Special code must send with the reservation for easily recognition by the customers
- MyTaxiService can be only use in Milan (including Malpensa and Linate airports)
- If in the concrete zone there are no available taxis in the queue, the system can ask the customer if he wants to automatically give the reservation to the closest zone with approximate waiting time and additional cost
- If there is no available taxi in the zone, customer can see queue of reservation taxis near the customer with approximate waiting time
- There is a Terms & Conditions section to indicate clearly the usage of the application, which if not followed will result in account deactivation

Chapter 3

Requirements

Starting from the goals we explain the following requirements:

1. **Allow the registration in the application to taxi drivers and passengers:**
 - The system should be able to provide a sign up functionality in order to create a new taxi driver or passenger profiles.
2. **Simplify the access of passengers to the taxi service:**
 - The system has to provide the possibility to the passenger to reserve a taxi
 - The system has to provide both a web application and mobile application in order to allow the passenger to save his time.
3. **Allow passenger to view if there is any taxi driver free:**
 - The system has to provide a functionality that allow a passenger to view if there is any free taxi driver for the desired route.
4. **Divide the city into different areas covered by some taxi drivers, organized in queues:**
 - The system has to provide a functionality to divide the city in some areas
 - The system has to organize the taxi of these areas into queues
5. **Guarantee a fair management of taxi queues:**
 - The system has to provide a functionality in order to manage the taxi queues in the best way possible.

6. **Allow the client to reserve taxi for an exact time:**

- The system has to provide a functionality that allow the passenger to reserve a taxi for a desired time.

3.1 Functional requirements

We specify here functional requirements for each actor:

1. **Guest:**

- Sign up

2. **Passenger:**

- Log in
- Manage personal informations
- Request a taxi
- Reserve a taxi

3. **Taxi driver:**

- Log in
- Manage his personal informations
- Share his position on the map
- Change his status (free or busy for a ride)

3.2 External interface requirements

3.2.1 User interface

Homepage & registration page



The passenger can access to the registration page and register himself to the application.

Login



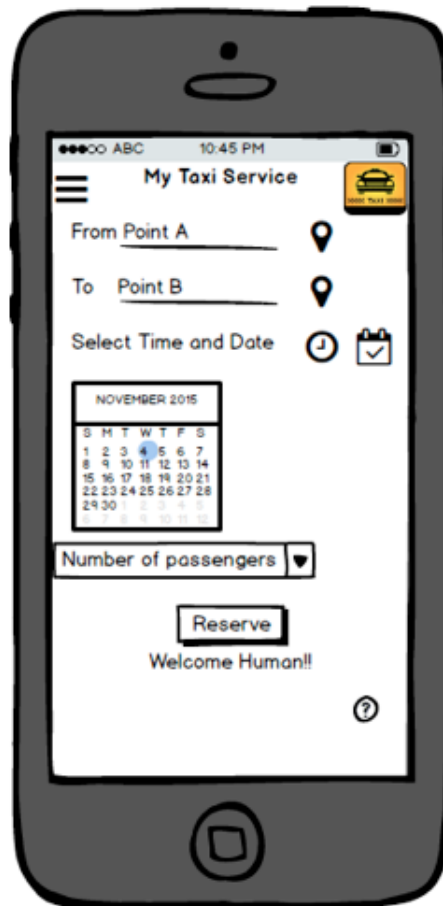
This is the login page. Here the user after entered his data can sign in the application.

Menu



This is the menu page, here the user can navigate and choose what to do. There are several options like: reserve taxi, check reservation, profile, help or log out.

Reservation



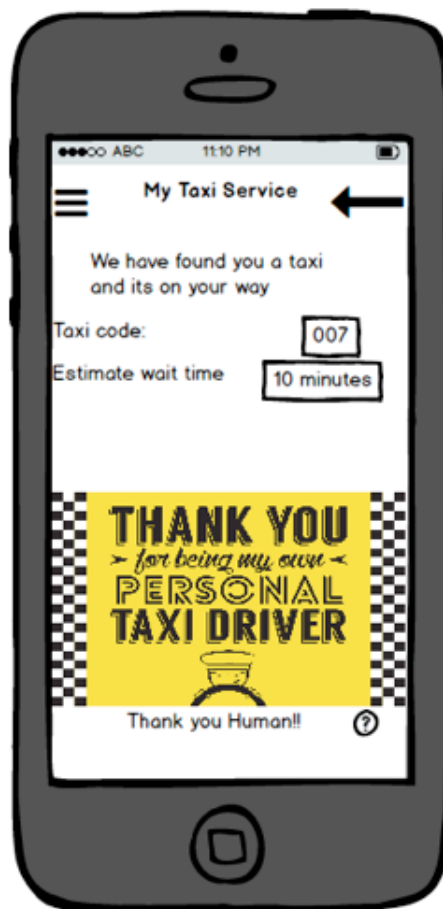
Here the passenger can reserve a taxi. He needs to insert the starting and destination point, the date and the number of passenger.

Confirm reservation



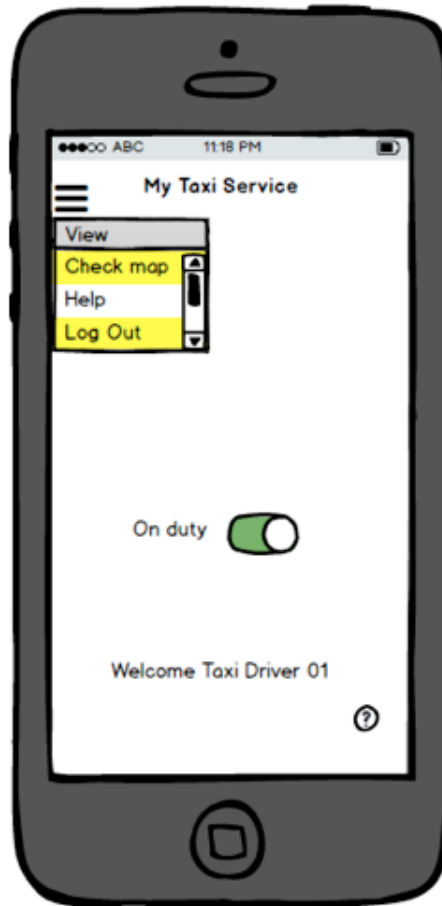
In this page the passenger can show a recap of his trip and confirm the reservation by clicking on the button

Recap reservation



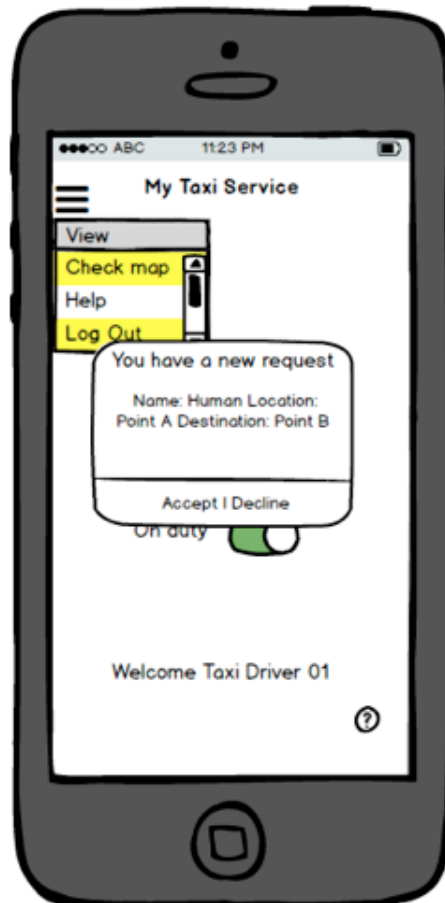
Here the passenger can see the estimate waiting time and the code of the taxi that will pick him up.

Taxi driver homepage



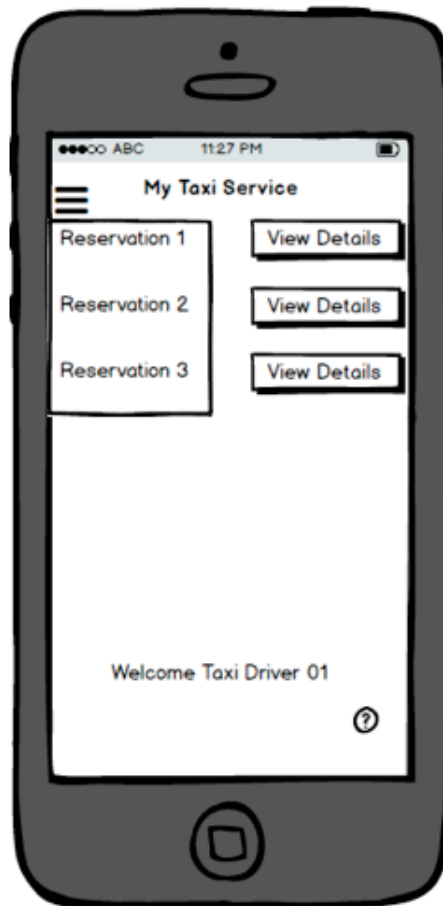
This is the home page for the taxi driver. He can choose from different option such as: Check map, Help or Log out. Also he can change his status in busy or free.

Taxi driver request



This is the notification that the taxi driver will receive when there is a request of a taxi. He can choose if accept or decline this request.

Reservation list for taxi driver



Here the taxi driver can give a look at the reservation he has in list and its details.

Taxi driver map



In this page the taxi driver can see the map of the itinerary for the reservation he chooses.

3.2.2 Software interface

Our software will have a database which saves members then serve a data management. In order to verify the data for the taxi drivers, the system must have access to the police database through and verify data on driving license and car. Enabling you to record your credit card the system will use bank pass that allows you to check the validity of information, the software have to use the maps google.

3.3 Scenarios

- **Scenario 1**

Alan has to leave for a trip in Milan and there is no one on arrival who can get him. Helping with his smartphone, he connects to myTaxiService app and does the log in using his data created in the registration phase. He requests a taxi in that area. In few minutes Alan is in the taxi and can get where he wants to lead.

- **Scenario 2**

Alan has to leave for a business trip in Milan and on arrival he must be at an appointment in less that an hour. Alan makes the access to the myTaxiService application, does the log in with his datas and makes a reservation for a taxi at the arrival time in Duomo. Now Alan can make the trip with tranquility and will not miss his appointment.

- **Scenario 3**

Mario is a taxi driver and he received a request for a ride, he accessed the app and change his status from free into busy. When he will finish the ride, he will change the status into free and waits for the next call.

- **Scenario 4**

Mario is a taxi driver and he must be at the Linate airport to pick up Alan at the arrival. During the travel the car has a problem and stops to work. Mario immediatly try to repair the car but there isn't anything to do. He sends a notification to Alan and cancel the reservation. Alan will be refunded on his credit card.

- **Scenario 5**

Luca hears a friend talking about MyTaxiDriver App so he decided to download it and use it because he need to make a reservation for the next day. After the download he registers himself into the application and in few minutes he is ready to go. He logs in and make the reservation. The next day he will find the taxi cab at the desired point.

3.4 Nonfunctional requirements

3.4.1 Reliability

- Application must provide real time driver availability information.
- Application must provide accurate estimate time arrival of the driver.
- Application must provide accurate estimate fare.
- Application must send the booking request to the licensed taxi drivers.

3.4.2 Availability

- Application must serve the customers 24 hours, 7 days in a week.

3.4.3 Security

- The system must use encryption to protect the customers and the drivers.

3.4.4 Usability

- User friendly interfaces (intuitive and easy to use)
- Fast access to content, information (in case of numbers request to the server)
- Presenting data and information?s in clear, logical and readable way.

3.4.5 Maintainability

- The program must be designed in a way that new addition can be done without changing the already developed structure.

3.4.6 Portability

- MyTaxiService is completely portable.

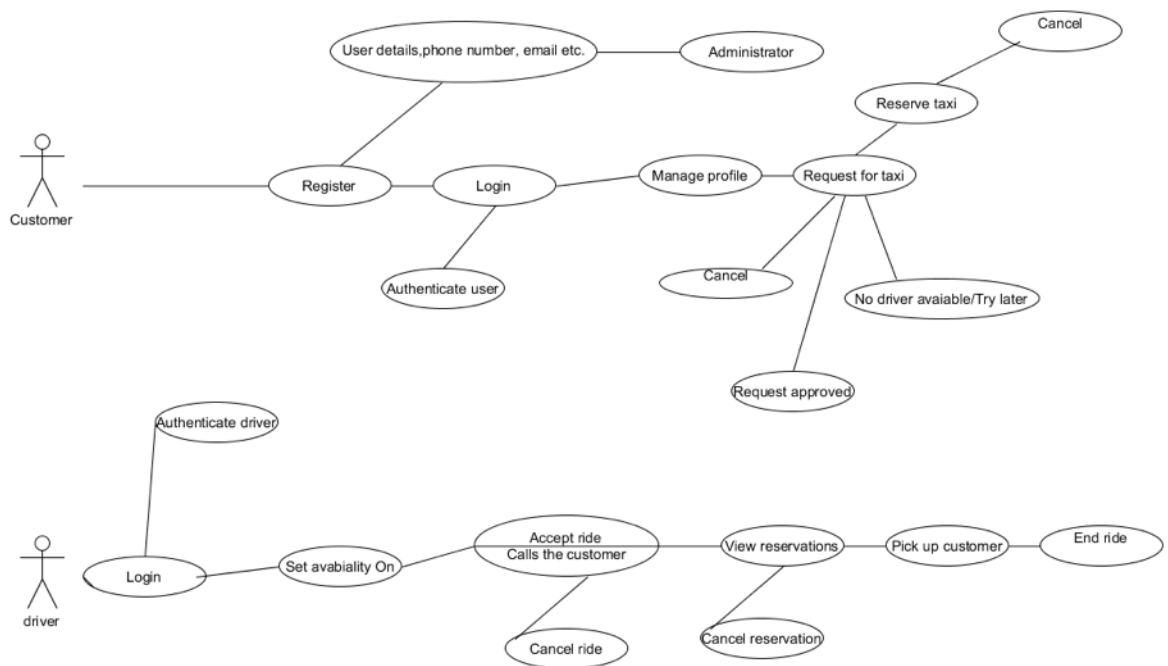
3.4.7 Future implementation

- Customers can use the option taxi sharing
- More available operating zones for the taxis
- You can add favorite drivers

Chapter 4

UML Models

4.1 Use Case Models

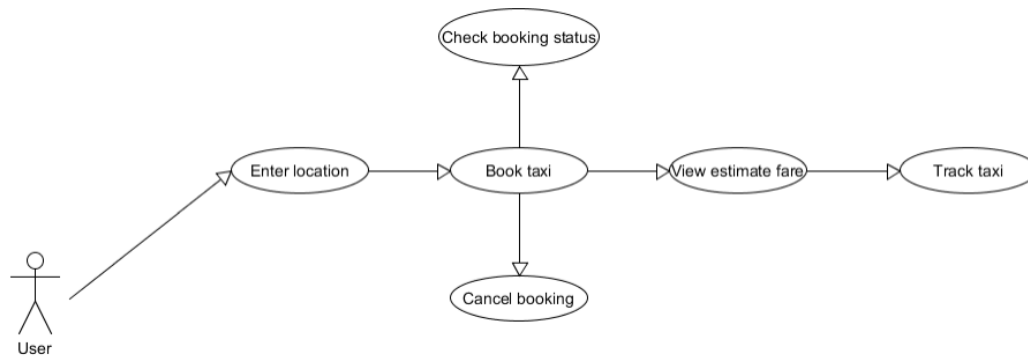




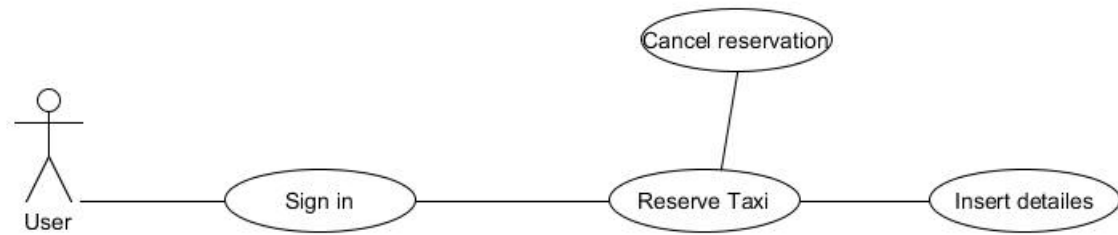
Name	Sign up
Actors	Guest
Entry conditions	Guest never made profile to our system
Events flow	<p>Guest is clicking on the registration form.</p> <p>The system shows to the guest the registration form where he needs to enter basic require informations and in the end he clicks sign in.</p> <p>The system checks if the informations are valid, confirms and redirects the guests to the homepage.</p>
Exit conditions	<p>The informations are stored in the database, confirmation mail is send to the guest and the user is redirected to the homepage</p>
Exceptions	<p>If the guest inserts non valid informations the system shows an error message</p>



Name	User log in
Actors	User
Entry conditions	The user wants to log in into the system
Events flow	<p>The user navigates to the home screen of the MyTaxiService.</p> <p>The system shows login page and the user inserts username and password and clicks on the sign in button</p> <p>The system validate the correctness of the inserted information, confirms and redirects to his personal page</p>
Exit conditions	The user has access to MyTaxiService functionality
Exceptions	If guest inserts non valid informations the system shows an error message



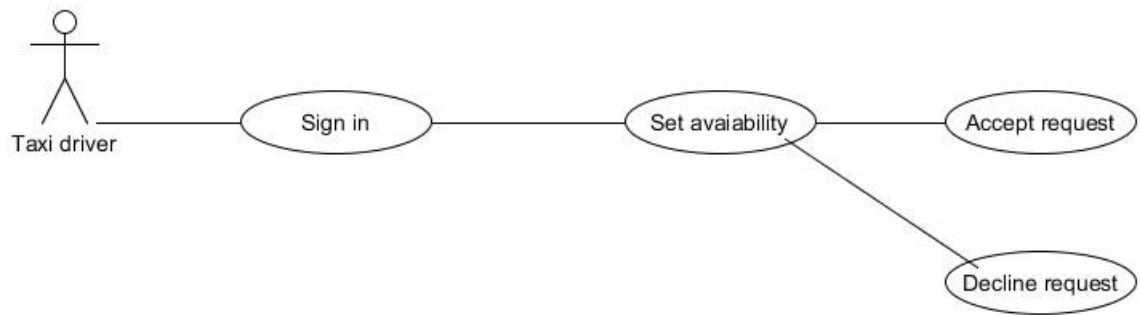
Name	Request for a taxi
Actors	User
Entry conditions	The user is logged in
Events flow	<p>The user choose "Request a taxi" in the homepage</p> <p>The user is redirected to the page where he enters his location and final destination. And he clicks request.</p> <p>After that the GPS is correctly acquired, the user can sent the request tapping the relative button.</p> <p>A confirmation page appears with the code of the incoming taxi and the waiting time</p>
Exit conditions	Code is send to the user and estimate waiting time
Exceptions	If the users didn't enter his location the system shows an error message and if there is no available taxi the system shows an error message



Name	Taxi reservation
Actors	User
Entry conditions	The user is logged in
Events flow	<p>The user choose "Reserve Taxi" in the home page User is redirected to "Reserve Taxi" section where he needs to enter the details for reserving taxi:</p> <ul style="list-style-type: none"> - Meeting location - Favorite destination - Date and time for pick up - Number of passengers <p>After entered required fields the user confirms by clicking the button "Reserve"</p>
Exit conditions	The user receives confirmation from the system and a push notification 10 minutes before the meeting time with the code of the incoming taxi
Exceptions	The user can't reserve taxi at least two hours before the ride



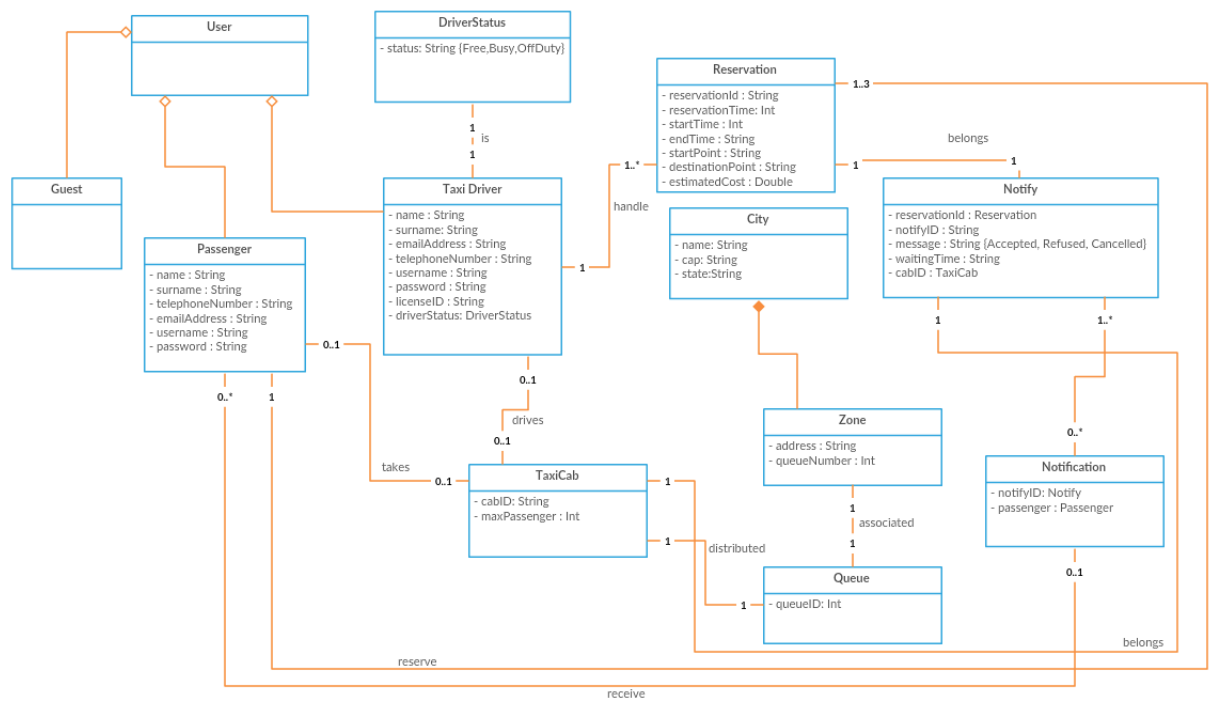
Name	Cancel ride
Actors	User
Entry conditions	The user is logged in
Events flow	The user goes to "Reserve taxi" option and clicks on "Cancel reservation"
Exit conditions	The reservation is properly deleted
Exceptions	The user can't cancel reservation less than 10 minutes before the taxi arrives



Name	Confirmation or declined request
Actors	Taxi driver
Entry conditions	The taxi driver is available, the taxi is the first in availability queue of its zone, so an available ride is coming soon
Events flow	The taxi driver is notified for a request and we can accept or declined the request
Exit conditions	The taxi driver accepts the ride The taxi driver rejects, the system will move the taxi in the last position of the relative available queue
Exceptions	If the taxi driver has some issues, he need to inform the administrator.

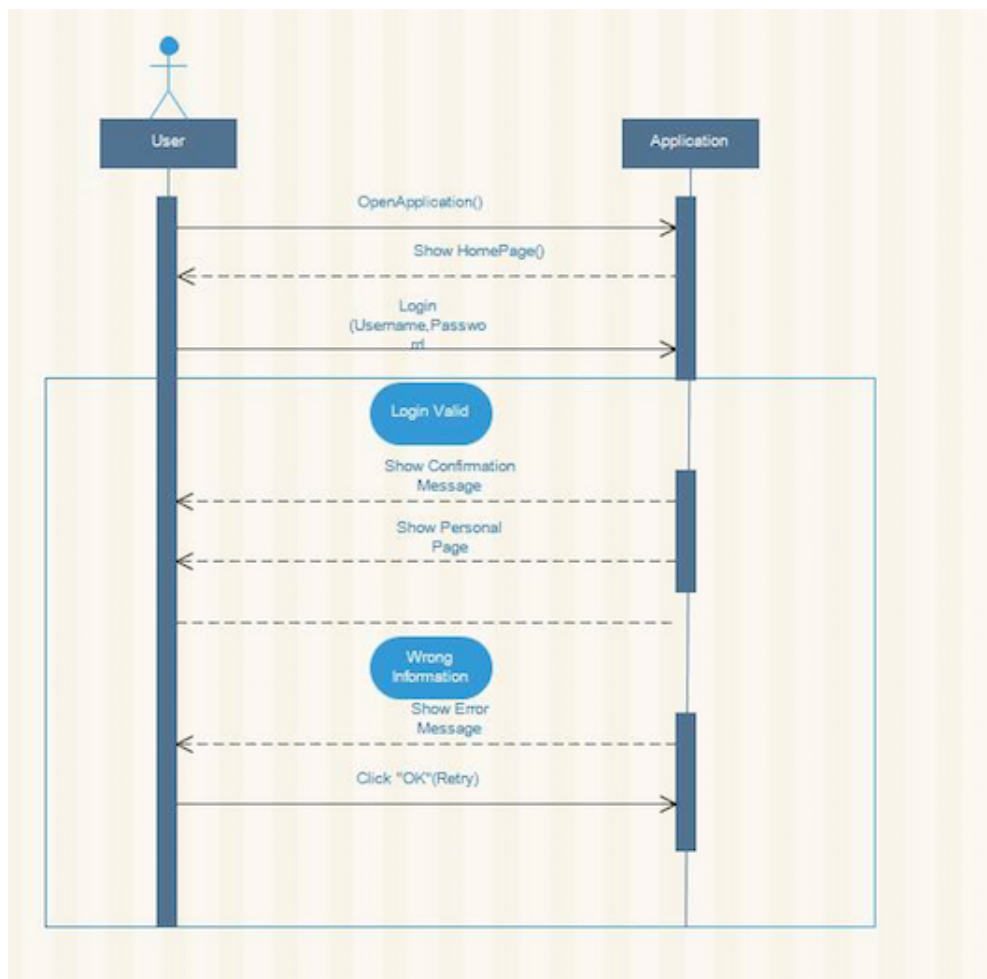
4.2 Class Diagram

Here we represent the class diagram of the entire system

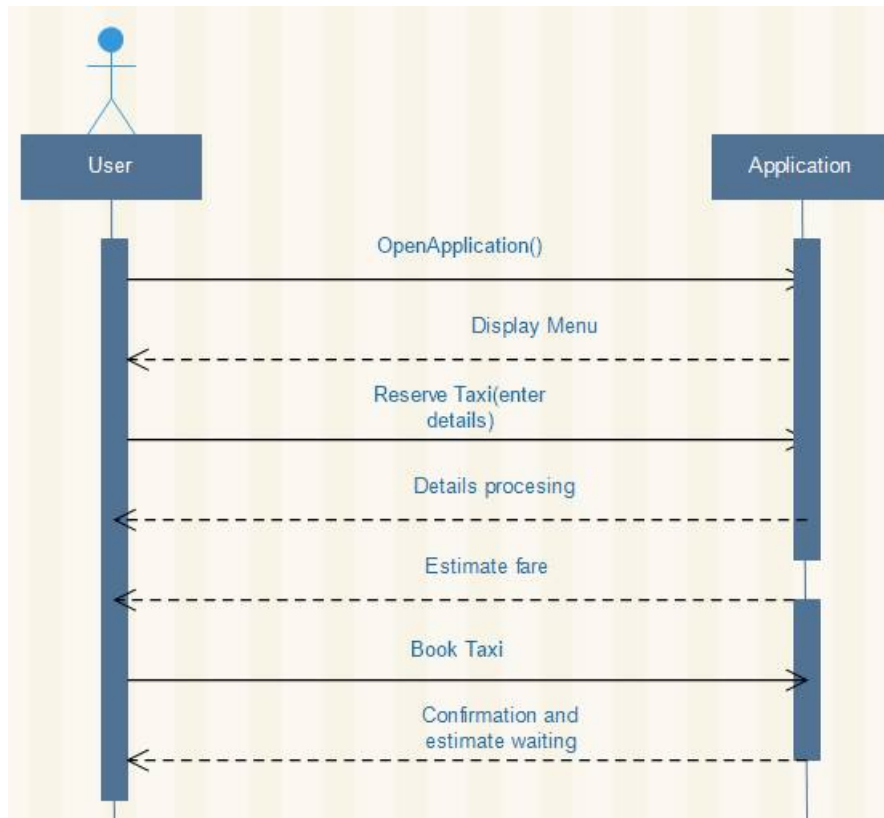


4.3 Sequence Diagrams

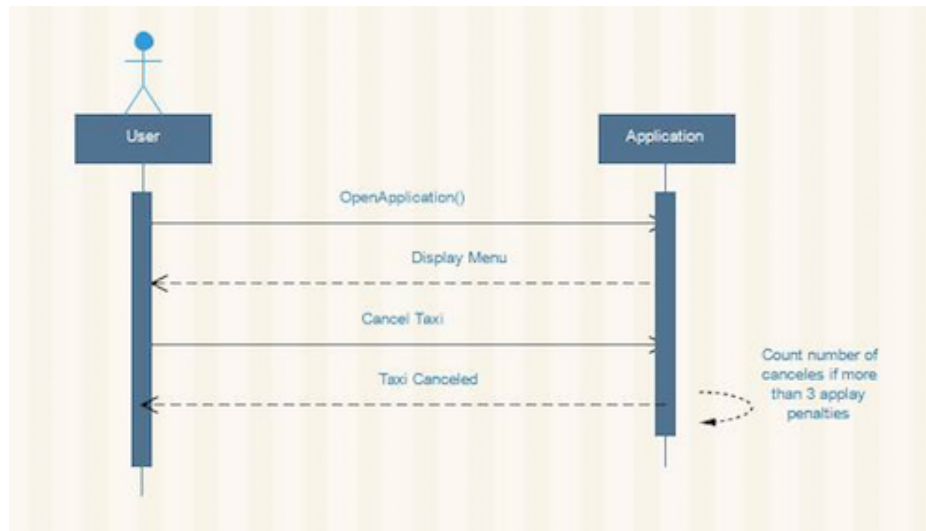
This is the sequence diagram for the login procedure



Here we represent the sequence diagram for the reservation of a taxi

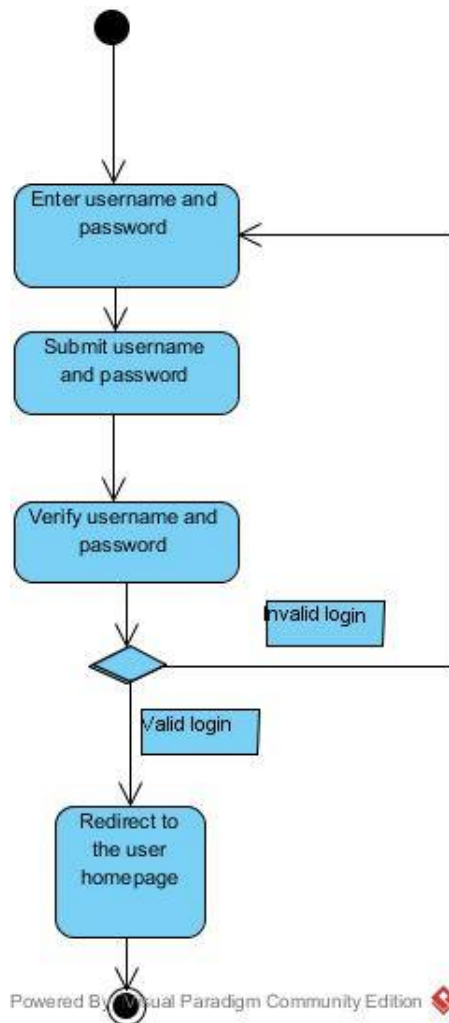


Here its described the sequence diagram for cancel a reservation

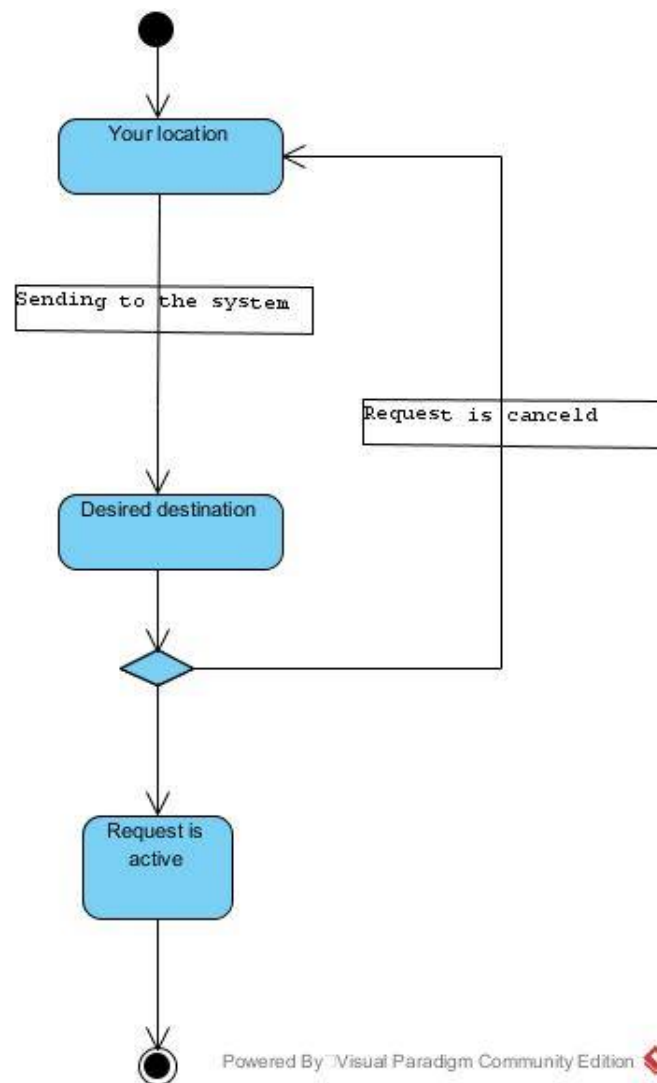


4.4 Activity diagrams

Login



Request



Powered By Visual Paradigm Community Edition

Chapter 5

Alloy

5.1 Signatures

```
abstract sig User{}
sig TaxiDriver extends User{}
sig Passenger extends User{}

sig TaxiCab{
  pass: some Passenger,
  driver: one TaxiDriver
}

abstract sig TaxiStatus{}
sig Available extends TaxiStatus{}
sig NotAvailable extends TaxiStatus{}

sig Queue{}

sig Zone{
  queues: one Queue
}

sig City{
  zones: set Zone
}{#zones >= 1}

sig Request{
  pass: some Passenger,
  driver: one TaxiDriver,
  taxi: one TaxiCab
}

sig Reservation{
  pass: some Passenger,
  driver: one TaxiDriver,
  taxi: one TaxiCab,
  resTime: Int,
  appointmentTime: Int
}
```


5.2 Facts

```
fact noSameQueueInTwoZones {  
  no q:Queue | some disj z1,z2: Zone | q in z1.queues and q in z2.queues  
}  
  
fact noSamePassangerInTwoTaxis{  
  no p: Passenger | some disj tc1,tc2: TaxiCab | p in tc1.pass and p in tc2.pass  
}  
  
fact noReservationAfter2Hours{  
  no r:Reservation | r.appointmentTime >= 2+r.resTime  
}
```

5.3 Predicates

```
pred RequestATaxi[r:Request, p:Passenger,d:TaxiDriver, t:TaxiCab]{  
  r.pass=p and  
  r.driver = d and  
  r.taxi= t  
}  
  
pred show(){#Zone>1 and #Queue>1}  
run show  
  
run RequestATaxi for 5
```

5.4 Results

Executing "Run show"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
2310 vars. 216 primary vars. 5047 clauses. 269ms.
Instance found. Predicate is consistent. 119ms.

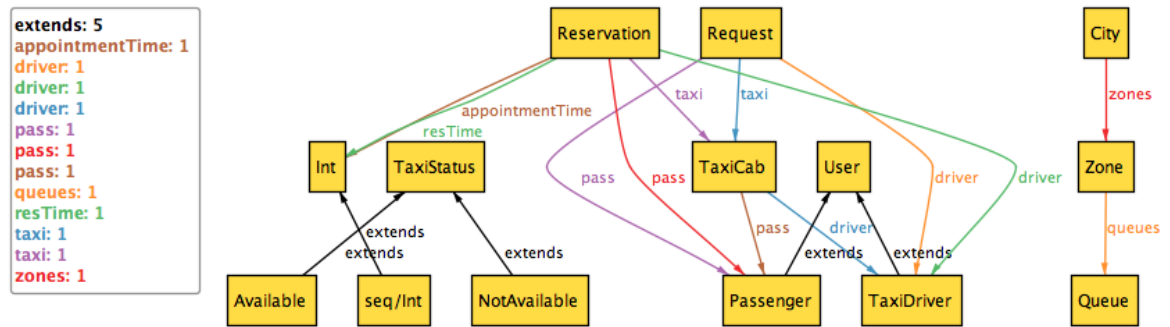
Executing "Run RequestATaxi for 5"

Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20
5247 vars. 470 primary vars. 11196 clauses. 256ms.
Instance found. Predicate is consistent. 72ms.

2 commands were executed. The results are:

- #1: Instance found. show is consistent.
- #2: Instance found. RequestATaxi is consistent.

5.5 Generated world



Chapter 6

Appendix

6.1 Software and used Tools

- TexShop (<http://pages.uoregon.edu/koch/texshop/>): to redact this document
- Alloy analyzer 4.2 (<http://alloy.mit.edu/alloy/>): to prove the consistency of the model
- CreatelyApp (<https://creately.com/app/>): to create Class diagram
- Balsamiq Mockups (<https://balsamiq.com/products/mockups/>): to create the user interface mockup
- Visual paradigm (<http://www.visual-paradigm.com/>): to create the activity diagrams
- UMLet (<http://www.umlet.com/>): to create sequence and use case diagrams

6.2 Working hours

Dimitar Anastasovski: ~35 hours

Marco Colombo: ~35 hours

6.3 Changelog

We had changed the alloy parts, the goals and some minor things from the first RAS document.