

POLYTECHNIC UNIVERSITY OF MILAN

School of Industrial and Information Engineering

Computer Science and Engineering



**Project of Software Engineering 2:
MyTaxi Service
Requirements Analysis and Specification
Document**

Course Professor: Prof. Elisabetta DI NITTO

Authors:

Mattia CRIPPA 000000

Francesca GALLUZZI 000000

Marco LATTARULO 841399

Academic Year 2015–2016

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Domain Properties	5
1.4	Proposed System	5
1.5	Goals	5
1.6	Assumptions	6
1.7	Stakeholders Identification	7
1.8	Definitions, acronyms and abbreviations	7
1.9	References	7
1.10	Overview	8
2	Actors Identifying	9
3	Specific Requirements	10
3.1	Functional Requirements	10
3.2	Non Functional Requirements	12
3.2.1	External Interfaces	12
3.2.2	User Interfaces	12
4	Scenario Identifying	20
4.1	Scenario 1	20
4.2	Scenario 2	20
4.3	Scenario 3	21
4.4	Scenario 4	21
4.5	Scenario 5	21
4.6	Scenario 6	22

5 UML Models	23
5.1 Use Case Diagram	23
5.2 Use Case Description	24
5.3 Class Diagram	24
5.4 Sequence Diagram	26
5.5 State Chart Diagram	26
6 Alloy Modeling	28
6.1 Alloy Code	28
6.2 World generated	28
7 Used Tools	29

List of Figures

3.1	Login Page mockup	14
3.2	Registration Form mockup	15
3.3	Passenger Home Page mockup	16
3.4	Passenger Request Page & Reservation Page mockups	17
3.5	Taxi Driver Home Page mockup	18
3.6	Taxi Driver Accepted Call Page mockup	19
5.1	Use Case Diagram	23
5.2	Class Diagram	25
5.3	State Chart Diagram for Passengers	26
5.4	State Chart Diagram for Taxi Drivers	27

Chapter 1

Introduction

1.1 Purpose

This document represent the *Requirement Analysis and Specification Document* (RASD). The main goal of this document is to completely describe the system in terms of functional and non functional requirements, analyze the real need of the customer to modeling the system, show the constraints and the limits of the software and simulate the typical use cases that will occur after the development. This document is intended to all developers and programmers who have to implement the requirements, to system analysts who want to integrate other systems with this one, and could be used as a contractual basis between the customer and the developer.

1.2 Scope

The aim of the project is to optimize the taxi service of a city using modern techniques and technologies to better manage the distribution and the availability of the taxicabs, to shorten the waiting time for passengers and to reduce costs for passengers and traffic in the entire city using taxi sharing. The system is implemented as a mobile application associated to an external, high precision GPS receiver for the taxi drivers and as a mobile application or a web interface for the passengers.

1.3 Domain Properties

We suppose that the following properties hold in the analyzed domain:

D01 : Accurate taxicabs positions are known by the GPS

D02 : Taxi drivers correctly signal their availability

D03 : If a passenger requests or reserves a taxi, he will then take the ride from the specified origin to the specified destination

D04 : A passenger doesn't change the origin or the destination of a ride after the reservation

D05 : If a taxi driver confirms a ride, then he will reach the passenger location on time according to the reservation hour or the waiting time calculated by the system

D06 : If a taxi driver confirms a ride, he will complete it

1.4 Proposed System

The system will be composed by a server running all the business logic, generating dynamic web pages and managing all the accesses to the data sources, and by a number of clients implemented as mobile application deployed on Android or a web application using the JEE platform. GPS raw data will be provided by a specific GPS receiver, installed in every taxicab and provided by a specialized company.

1.5 Goals

Visitors should be able to:

G01 : Sign up

G02 : Log in

Passengers should be able to:

G03 : Request a taxi

G04 : Reserve a taxi

G05 : Receive information about the incoming taxi (confirmation, taxi ID, waiting time)

G06 : Share the ride

G07 : Receive receipts after each completed ride

Taxi drivers should be able to:

G08 : Signal whether they are available or not

G09 : Receive incoming requests

G10 : Accept or decline incoming requests

G11 : Visualize information about the optimal route

Developers should be able to:

G12 : Add new features

1.6 Assumptions

A01 : Given the collected data about the distribution and the total number of the taxicabs, we assume that the coverage of the taxicabs among the zones is almost uniform; therefore the probability of having an empty queue is reasonably low.

A02 : Given the collected data about the acceptance rate of the taxi drivers, we assume that the probability of reaching the end of the queue without any acceptance is reasonably low.

A03 : We assume that the maximum number of people sharing the same ride is 3.

A04 : We assume that the taxi driver will correctly follow the optimal route suggested by the system.

1.7 Stakeholders Identification

- The government of the city in which the system will be used
- The citizens of the city in which the system will be used
- The taxi drivers of the city in which the system will be used
- A specific IT company which provides all the technology (GPS receivers, smartphones for taxi drivers, mainframes and computational power) in exchange of raw GPS data provided by the receiver used for data mining purposes

1.8 Definitions, acronyms and abbreviations

PASSENGER: is a citizen that benefits of the taxi service

TAXI DRIVER: is the owner of a taxicab, and provides a taxi service to citizens

RIDE: is the act of make use of the taxi service (provided by a taxi driver) by a passenger

REQUEST: is the act performed by a passenger when he immediately needs a taxicab in the location in which he is

RESERVATION: is the act performed by a passenger when he needs a taxicab in a certain future time (at maximum in 2 hours from now) at a certain specific location

ROUTE: is the path through the city from the origin of the specific ride to the destination of the specific ride

1.9 References

- Specification Document: MyTaxiService Project AA 2015-2016.pdf.
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

- IEEE Std 1016tm-2009 Standard for Information Technology - System Design - Software Design Descriptions.

1.10 Overview

This document is essentially structured in eight parts:

- **Section 1** → Introduction: it gives a description of document and gives general information about the software product with more focus about constraints and assumptions.
- **Section 2** → Actors Identification: it gives a description of all actors of the system.
- **Section 3** → Specific Requirements: this part lists all the functional and non functional requirements that are part of the system.
- **Section 4** → Scenarios Identification: it gives a description of typical scenarios.
- **Section 5** → UML Models: to give an easy way to understand all functionality of this software, this section is filled with UML diagrams and it contains also some use cases.
- **Section 6** → Alloy Modeling: this part contains the code used for the analysis of consistency of the Class Diagram and words generated by Alloy Analyzer in order to understand that the model is consistent.
- **Section 7** → Used Tools: this part contains some information about the software used to realize this document.

Chapter 2

Actors Identifying

The actors of our system are:

- ***Visitor***: unregistered user that access the application interface in order to sign up or Log In as passenger, taxi driver or developer and start interacting with the system.
- ***Passenger***: this user, after successful Log In, depending on his/her needs, access the request interface or the reservation interface. He/she has also access to an information page about all his/her accepted reservations or requests and has access to a page with the chronology of his/her receipts.
- ***Taxi driver***: this user, after successful Log In, is enabled to receive reservations or requests from passengers and can decide whether to accept them or not. He/she can access at any time after the acceptance a summary page about the specific ride. This user is also able to signal at any moment his/her availability to accept new reservations or requests.
- ***Developer***: this user, after successful Log In, can access a specific interface through which he/she can introduce modifications to the system or access privileged information about the system for maintenance scopes.

Chapter 3

Specific Requirements

3.1 Functional Requirements

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs.

G01 → To allow a visitor to sign up, the system shall provide functionalities to:

R01 Check the validity and correctness of the information provided by the visitor (personal information, password, payment information)

R02 Check if the user is already registered into the system

G02 → To allow a visitor to login, the system shall provide functionalities to:

R03 Check if username and password provided by the visitor correspond to an existing user, authorized to use the system

R04 Prevent unauthorized or banned users from accessing the system

G03 & **[G04]** → To allow a passenger to request and to reserve a taxi, the system shall provide functionalities to:

R05 Obtain the passenger location

R06 Access the queue associated to the right taxi zone

R07 Check the availability of the taxi drivers

R08 Iteratively contact all the taxi drivers of the queue starting from the first one until one of them accepts the call

R09 Iteratively search for an available taxi driver inside adjacent zones in the case that the right zone has an empty queue or all the contacted taxi drivers had declined the request

G05 → To allow a passenger to receive information about the incoming taxi the system shall provide functionalities to:

R10 Obtain the taxi driver position and the passenger position and estimate the time needed by the taxi driver to reach the passenger

R11 Obtain the taxicab unique identifier from the taxi drivers database

G06 → To allow a passenger to share a ride, the system shall provide functionalities to:

R12 Check for each request or reservation if the passenger had selected the sharing function

R13 Compare routes that start from the same taxi zone and determine whether or not they can be merged into one, according to specific rules of comparison

R14 Calculate the correct distribution of the fee according to specific rules based on the percentage of the kilometers shared with others or traveled alone

R15 Elaborate an optimal route for taking every passenger to the right destination and show it to the taxi driver

G07 → To allow a passenger to receive receipts after each completed ride, the system shall provide functionalities to:

R16 Keep track of the actual route followed by the taxi driver and keep track of the actual duration of the ride

G08 & **[G10]** → To allow a taxi driver to signal whether he is available or not, and to accept/decline an incoming request, the system shall provide functionalities to:

R17 Monitor and collect inputs from taxi drivers

G09 → To allow a taxi driver to receive incoming requests, the system shall provide functionalities to:

R18 Contact taxi drivers and forward them all the information about the proposed request (position of the passenger, destination of the passenger, sharing option enabled or not)

G11 → To allow a taxi driver to visualize information about the optimal route for the ongoing ride, the system shall provide functionalities to:

R19 Retrieve the taxi location and the locations of all the passengers of the ride

R20 Access and query the map provider service to obtain an updated map with information about traffic, smashes, road construction sites

G12 → To allow a developer to add new features, the system shall provide functionalities to:

R21

3.2 Non Functional Requirements

3.2.1 External Interfaces

Hardware Interface:

Device should be enabled with Internet and GPS receiver.

Software interface:

The user's browser should be HTML5 compatible and the resolution should be at least 1280x720 for a satisfactory user experience.

3.2.2 User Interfaces

Here are presented some mockups that represent an idea of the structure of the application pages:

Login

The mockup in Figure 3.1 shows the Login Page of MyTaxiService. Here users can log in to the application.

Registration Form

The mockup in Figure 3.2 shows the Registration Form Page that visitors can access.

Passenger Home Page

The mockup in Figure 3.3 shows the Home Page for a Passenger user.

Passenger Request Page and Reservation Page

The mockups in Figure 3.4 shows the Request and the Reservation Pages for a Passenger.

Taxi Driver Home Page

The mockup in Figure 3.5 shows the Home Page for a Taxi Driver user.

Taxi Driver Accepted Call Page

The mockup in Figure 3.6 shows the Page for a Taxi Driver user that accepts a call.

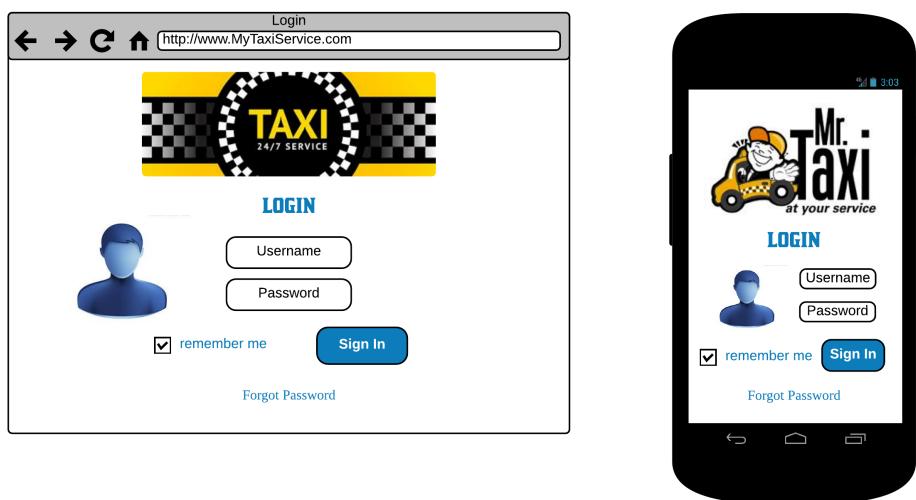


Figure 3.1: Login Page mockup

The image displays two versions of a registration form for a taxi service. On the left is a desktop browser window titled 'Login' with the URL 'http://www.MyTaxiService.com'. The page features a 'TAXI 24/7 SERVICE' logo with a checkered background. Below it is a 'REGISTRATION FORM' section. It includes fields for 'Username', 'Password', 'Confirm Password', and 'Email Address'. There are three radio buttons for 'Passenger', 'Driver', and 'Developer', with 'Passenger' selected. A checkbox labeled 'I have read the MyTaxiService Terms of Service' is checked. At the bottom are 'Register' and 'Cancel' buttons. On the right is a smartphone screen showing the same registration form. The phone's status bar indicates '3.03'. The app interface includes a cartoon character and the text 'Mr. Taxi at your service'. The registration fields and user type selection are identical to the desktop version. At the bottom of the phone screen are standard Android navigation icons.

Figure 3.2: Registration Form mockup

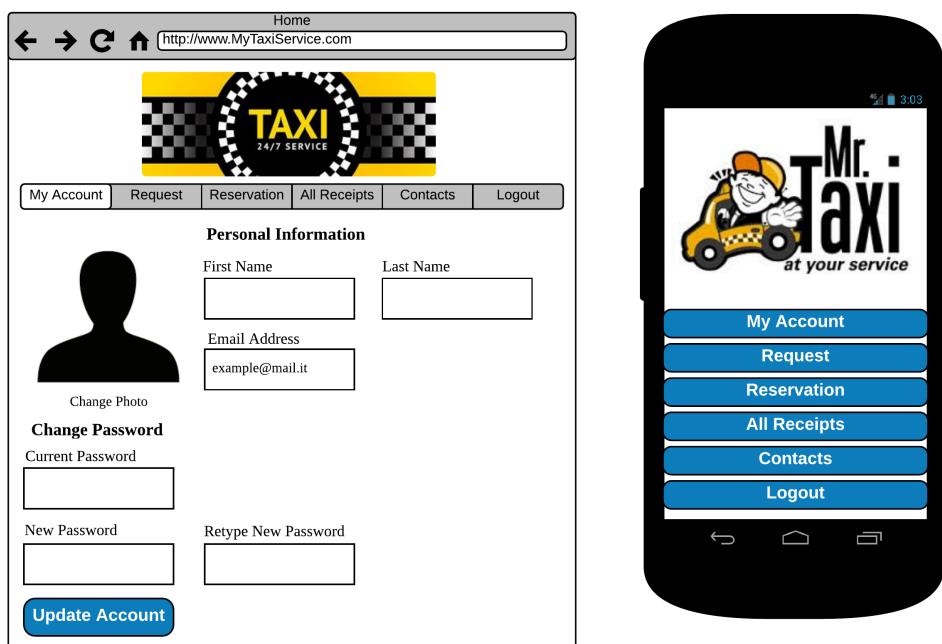


Figure 3.3: Passenger Home Page mockup

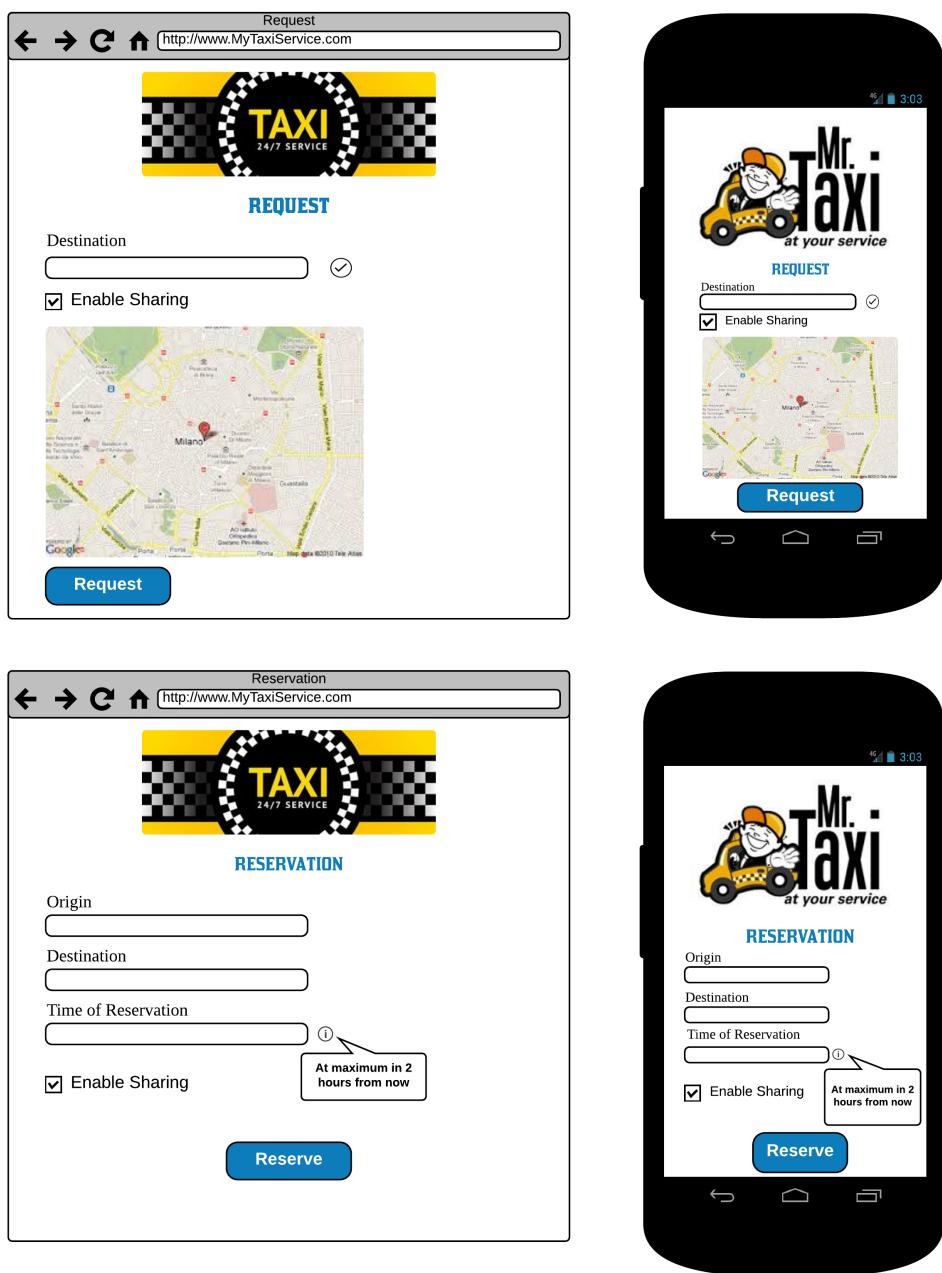


Figure 3.4: Passenger Request Page & Reservation Page mockups



Figure 3.5: Taxi Driver Home Page mockup

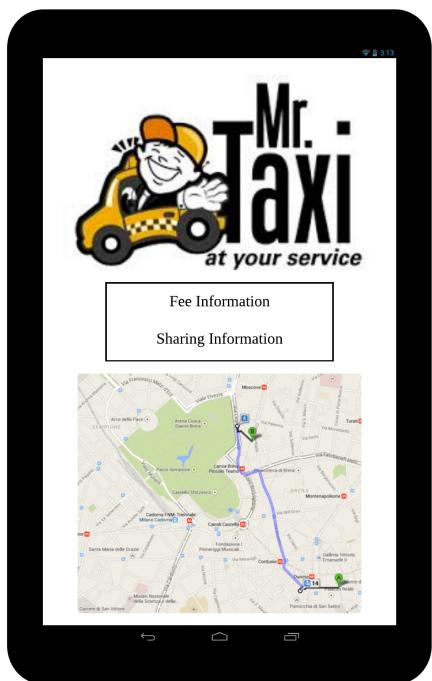


Figure 3.6: Taxi Driver Accepted Call Page mockup

Chapter 4

Scenario Identifying

4.1 Scenario 1

Visitor V wants to sign up into the system as a Passenger V opens the mobile or the web app and he/she is immediately asked to Log In or Sign Up. V clicks on the button "Sign Up" and is brought to the registration interface. He/She inserts the desired Username, Password and Email address and selects "Passenger" as type of user. Then V clicks on the button "Register" and, if the credentials are correct, he/she receives a confirmation message. The Passenger Home Page now appears on the app.

4.2 Scenario 2

Passenger A requires a standard ride R and driver D accepts A clicks on the "Request" tab from the Passenger Home Page and the request interface appears on the app. A inserts the destination for his/her desired ride and doesn't check the enable sharing option, then clicks on the button "Request". D receives the incoming request sent by A and decides to accept the call. D receives a message with a summary of R, including indication for an optimal route, and also A receives a confirmation message with information about the taxicab ID of D and the estimated waiting time.

4.3 Scenario 3

Passenger A reserves a standard ride R. Taxi Driver D1 decline the call but second Taxi Driver D2 accepts A clicks on the "Reserve" tab from the Passenger Home Page and the reservation interface appears on the app. A inserts the origin, the destination, the desired time for R and doesn't check the enable sharing option, then clicks on the button "Reserve". D1 receives the incoming reservation sent by A and decides to decline the call, so the reservation is sent to D2, that decides to accept the call. D2 receives a message with a summary of R, including indication for an optimal route, and also A receives a confirmation message with information about the taxicab ID of D2.

4.4 Scenario 4

Passenger A1 requires shared ride and another Passenger A2 is added to the same ride. The Taxi Driver D accepts A1 clicks on the "Request" tab from the Passenger Home Page and the request interface appears on the app. A1 inserts the destination for his/her desired ride and checks the enable sharing option, then clicks on the button "Request". A2 does the same thing and, because they are in the same zone and they are going in the same direction, they are put together in the same ride. D receives the incoming request sent by A1 and A2 and decides to accept the call. D receives a message with a summary of R, including indication for an optimal route and the number of passengers, and also A1 and A2 receive a confirmation message with information about the taxicab ID of D the estimated waiting time and the total number of passengers.

4.5 Scenario 5

After a ride, Passenger A wants to see the receipt A has just completed the ride that he/she requested. A opens the app and, after logging in, clicks on the "All Receipts" tab. A list of all his/her receipts appears in the app ordered from the most recent to the oldest. A clicks on the first receipt, which is the one that corresponds to the just completed ride, and sees all the details of the receipt.

4.6 Scenario 6

A Developer D access the app to introduce modifications to the system D opens the mobile or the web app and he/she is immediately asked to Log In or Sign Up. D clicks on the button "Log In" and inserts his/her credentials. If the credentials are correct, D has access to a specific interface dedicated to developers. D introduces modifications to the system and, after checking that everything is correct, submits the modifications. The app restarts and the new modifications are applied. D logs in again and verifies that everything works correctly.

Chapter 5

UML Models

5.1 Use Case Diagram

The Figure 5.1 gives an overview on the main actors involved in the system and the functionalities that they are able to execute.

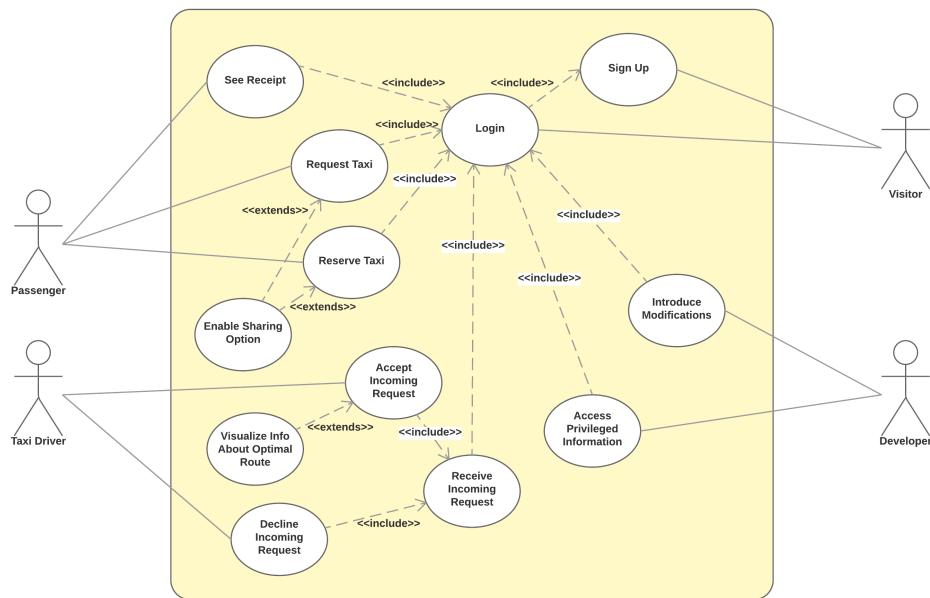


Figure 5.1: Use Case Diagram

5.2 Use Case Description

Sign Up

Login

Request Taxi

Reserve Taxi

Enable Sharing Option

See Receipt

Accept Incoming Requests

Decline Incoming Requests

Receive Incoming Requests

Visualize Information About Optimal Route

Introduce Modifications

Access Privileged Information

5.3 Class Diagram

Here is presented the UML class diagram in Figure 5.2. This diagram will be updated during the developing process especially by adding all methods:

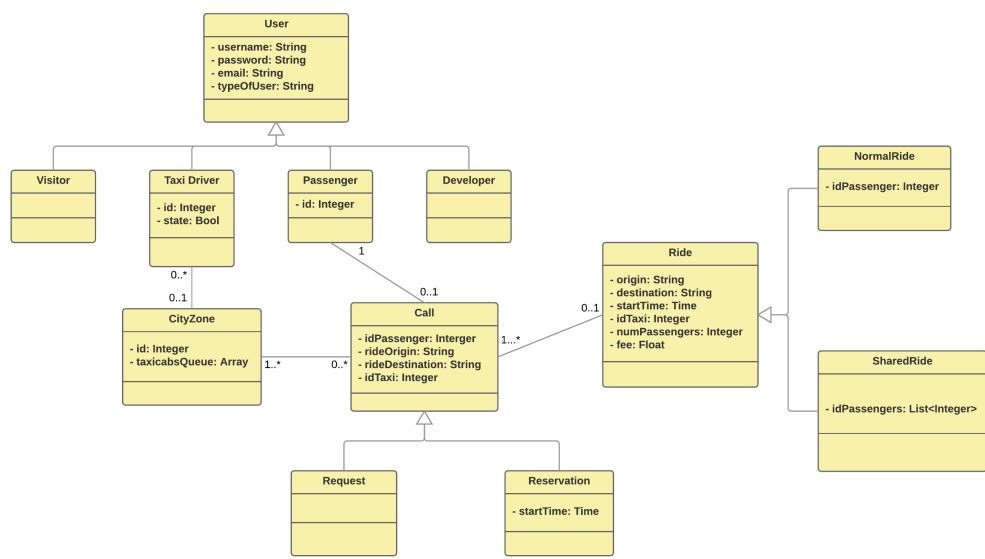


Figure 5.2: Class Diagram

5.4 Sequence Diagram

Log In

Request a Taxi Ride

Reserve a Taxi Ride

Introduce Modifications in the System

5.5 State Chart Diagram

In this section the behavior of some entities presented in Figure 5.2 is exposed using UML state chart diagrams. The following state chart diagrams will give a simplified vision of entire application:

State Chart Diagram for Passengers

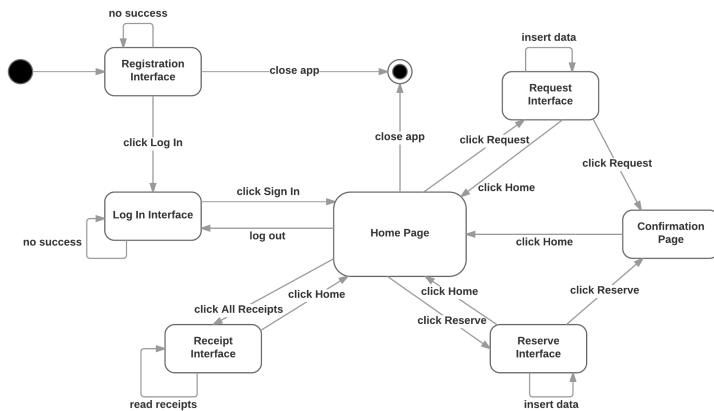


Figure 5.3: State Chart Diagram for Passengers

State Chart Diagram for Taxi Drivers

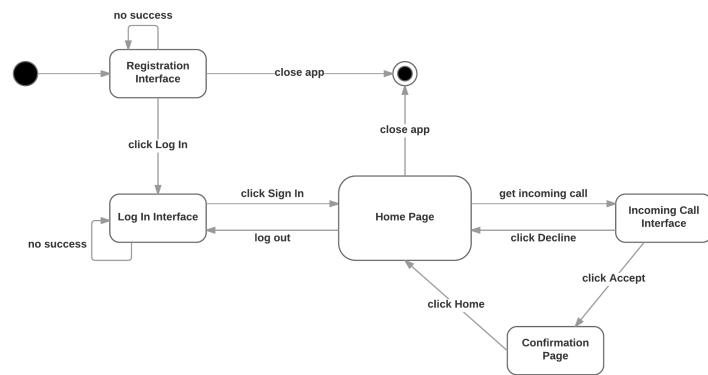


Figure 5.4: State Chart Diagram for Taxi Drivers

Chapter 6

Alloy Modeling

6.1 Alloy Code

6.2 World generated

Chapter 7

Used Tools