

Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “myTaxiService”

Requirements Analysis and Specifications

Document

Version 1.0

Milica Jovanovic (mat. 835953), Pavle Vidanovic (mat. 854472)

23 October 2015

Contents

[1 Introduction 5](#_Toc434334340)

[1.1 Purpose 5](#_Toc434334341)

[1.2 Scope 5](#_Toc434334342)

[1.3 Glossary 5](#_Toc434334343)

[1.4 Reference Documents 6](#_Toc434334344)

[1.5 Document Overview 6](#_Toc434334345)

[2 Overall Description 7](#_Toc434334346)

[2.1 Product perspective 7](#_Toc434334347)

[2.2 Identifying stakeholders 7](#_Toc434334348)

[2.3 User characteristics 7](#_Toc434334349)

[2.4 Actors identifying 7](#_Toc434334350)

[2.5 Goals 7](#_Toc434334351)

[2.6 Domain properties 8](#_Toc434334352)

[2.7 Constraints 8](#_Toc434334353)

[2.7.1 Regulatory policies 8](#_Toc434334354)

[2.7.2 Hardware limitation 8](#_Toc434334355)

[2.7.3 Interfaces to other applications 8](#_Toc434334356)

[2.7.4 Parallel operation 8](#_Toc434334357)

[2.8 Assumptions 8](#_Toc434334358)

[2.9 Future possible implementation 9](#_Toc434334359)

[3 Specific Requirements 10](#_Toc434334360)

[3.1 External Interface Requirements 10](#_Toc434334361)

[3.1.1 User Interfaces 10](#_Toc434334362)

[3.1.2 API interfaces 19](#_Toc434334363)

[3.1.3 Hardware Interfaces 19](#_Toc434334364)

[3.1.4 Software Interfaces 19](#_Toc434334365)

[3.2 Functional Requirements 19](#_Toc434334366)

[3.3 The world and the machine 20](#_Toc434334367)

[3.4 Scenarios 21](#_Toc434334368)

[3.4.1 Scenario 1 21](#_Toc434334369)

[3.4.2 Scenario 2 21](#_Toc434334370)

[3.4.3 Scenario 3 21](#_Toc434334371)

[3.4.4 Scenario 4 21](#_Toc434334372)

[3.5 UML Models 22](#_Toc434334373)

[3.5.1 Use Case 22](#_Toc434334374)

[3.5.2 Class Diagram 31](#_Toc434334375)

[3.5.3 Sequence Diagram 32](#_Toc434334376)

[3.5.4 State Chart diagrams 38](#_Toc434334377)

[3.6 Non Functional Requirements 40](#_Toc434334378)

[3.6.1 Performance Requirements 40](#_Toc434334379)

[3.6.2 Software System Attributes 40](#_Toc434334380)

[4 Appendix 41](#_Toc434334381)

[4.1 Alloy 41](#_Toc434334382)

# Introduction

## Purpose

The purpose of this document is to provide a comprehensive description of the myTaxiService system. It’s aim is to communicate what the software should do and identify the capabilities and characteristics of the system being developed, as well as the constraints it should respect. This document is meant for everyone included in the production of the software.

## Scope

The aim of this project is to develop and implement myTaxiService, an application similar to Uber, which makes the process of assigning an available taxi vehicle to possible passengers.

The developed system should allow new users to register. Users, once logged in, should be able to:

* request a taxi
* reserve a taxi
* cancel a ride
* check taxi availability around him
* receive a confirmation with information about the assigned vehicle and ETA once taxi is requested
* create/maintain user profile
* report a taxi driver

The developed system should allow new taxi drivers to register. Drivers, once logged in, should be able to:

* inform the system about their availability
* confirm/decline that they are going to take care of a certain call
* create/maintain taxi driver profile
* report a passenger

The system should keep information about new arrived requests, as well as the confirmed rides. A ride should have and id number, information about the passenger that requested the ride, as well as the code of the assigned vehicle and ETA. System should also keep information about taxi queues connected to particular zone of the city and ensure fair management of the queues. Developed system should keep information about the list of reservations made by passengers, such as id number of the reservation, information about the passenger that made the reservation and the time of reservation and time of the ride.

## Glossary

The following are the definitions of some commonly used phrases throughout the document:

|  |  |
| --- | --- |
| *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 the ride |
| *Request* | Passenger filled form for immediate ride |
| *Reservation* *Conformation* | Notification sent to the user about the confirmed reservation |
| *Ride Conformation* | Notification sent to the user about the confirmed ride with information of the ride |
| *Report* | Short description of problem that user/driver stumped into |
| *User* | A person already registered and logged into the system |
| *Guest* | A person accessing a system that has either never registered or hasn't logged in yet. Guest has only two available options, to log in or to 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*.* |

Podsetnik

* OS:

## Reference Documents

* IEEE Std 830-1998 Recommender Practice for Software Requirements Specifications
* Specification Document: myTaxiService Project AA 2015-2016.pdf

## Document Overview

The document is essentially structured in four parts:

* Chapter 1: Introduction, gives description of document and some basic information about the software
* Chapter 2: Overall Description, gives an overview of the main functionalities of the software to be with constraints, as well as the hardware limitations.
* Chapter 3: Specific Requirements, body of the document that describes in more detail functionalities, possible scenarios and use cases.
* Chapter 4: Appendix, Alloy simulation of the previous UML diagrams.

# Overall Description

## Product perspective

*myTaxiService* is mobile web application that provides user with services described in section 1.2. System will consist of two applications and server between them. The software will be developed using a client-server model. The server side contains the application logic and is used to interact with permanent storage, serve pages to the client and process user input. The web client consists of dynamic web pages which provides user friendly graphical interface and the web browser through which they are accessed. Considering *myTaxyService* is a mobile web application, it is platform independent. The only requirement is users having a web browser installed on a device of their choice.

## Identifying stakeholders

There are four distinct interest groups of people regarding this project:

* *Company* that provided project specification and expect it to be delivered in a way that satisfies given specification while respecting the set deadlines and budget
* *Developer group,* in this case group of two people
* *Taxi driver,* person working for the company that ordered the software product
* *Passenger*, person who need a ride to specific location

## User characteristics

myTaxiService is expected to have users across a wide range of demographics, meaning users of any age, gender and educational background. Still, given of the ubiquitous nature of internet and social media, it is assumed that people using our software do have the basic web browsing skills.

## Actors identifying

Four possible actors interacting with our system are the following:

* *Guest* person accessing a system that has either never registered of hasn't logged in yet. Guest can only access the initial page from where he has only two available options, to log in or to sign up for the first time
* *User* a person already registered and logged into the system. User can use all of the features offered by the passenger application
* *Taxi**driver*a person already registered and logged into the system. User can use all of the features offered by the taxi driver application
* *Admin* a person responsible for handling reports on users of my taxi service. Admin can ban a user or a driver from the system

## Goals

Having possible users in mind, myTaxyService should have these features:

* [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)

## Domain properties

It is supposed that 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

## Constraints

### Regulatory policies

myTaxiService application will not take advantage of users personal information and will respect the privacy policy. User will be notified about it.

### Hardware limitation

User must have access to Internet and own a device with a web browser and GPS service.

### Interfaces to other applications

myTaxiService application is integrated with Google Maps API to access their maps, Google Places API to suggest street addresses to the user and email service in order to make authentication.

### Parallel operation

myTaxiService will support parallel access to the applications database in a transparent way.

## Assumptions

Considering that there were some ambiguities in the specification document, the following facts are assumed:

* 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
* there is a Terms & Conditions section to indicate clearly the usage of the application, which if not followed will result in account deactivation
* we assume that Google Maps service will calculate location used by myTaxiService accurately
* we assume that 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

## Future possible implementation

* + - another way of payment will be added as an options for users to pay online
    - an option for users to rate the drivers
    - Facebook authentication could be added as a way of registering
    - application could be updated so it works more efficiently
    - taxi sharing options could be added to the system

# Specific Requirements

## External Interface Requirements

### User Interfaces

In this section mockups of myTaxiService application will be enclosed which will mimic the general idea of the software system:

In figure 1. Initial page is shown, which offers two possibilities to the guest user. Guest user could create an account if he does not own one or he could sign in with his credentials.

In figure 2. Sign Up page is shown, which provides user with input fields to add his account information. After the fields are filled user can confirm by clicking the Ok button.

|  |  |
| --- | --- |
| C:\Users\milica13629\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1 Guest screen.png | D:\FAX\SW2\SW2015\Mockups\2 SignUp.png |
| Figure 1. Initial page | Figure 2. Sign Up |

In figure 3. Sign In page is shown, which provides user with input fields to add credentials. After the fields are filled user can sign in by clicking the button.

In figure 4. Passenger Home page is shown, which shows user’s location and a drop down menu with different options. From this page user can choose option to request a taxi, reserve a taxi, change his profile, check available taxis around him, see received notifications, go to Help section or simply log out from his account.

|  |  |
| --- | --- |
| C:\Users\milica13629\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3 Sign In.png | D:\FAX\SW2\SW2015\Mockups\4 Home screen.png |
| Figure 3. Sign In | Figure 4. Passenger Home page |

In figure 5. Request page is shown, which provides user with possibility to request a vehicle from his current location showed on the map.

In figure 6. Reservation page is shown, which provides user with possibility to reserve a vehicle from a certain source address to certain destination address at picked time.

|  |  |
| --- | --- |
| C:\Users\milica13629\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5 Request.png |  |
| Figure 5. Request page | Figure 6. Reservation page |

In figure 7. Driver Home page is shown, which provides driver with possibility to state his availability, manage his profile and check information about his current ride.

In figure 8. Pop up message is shown, it informs a driver that a certain passenger has requested his service. Driver has the possiblility to accept or decline the ride.

|  |  |
| --- | --- |
|  |  |
| Figure 7. Driver Home page | Figure 8. Pop up message |

In figure 9. Navigation page is shown, which provides the driver with the route from his location to the source address of the requested ride.

In figure 10. Taxis available screen is shown, which shows user taxi vehicles available in his zone.

|  |  |
| --- | --- |
|  |  |
| Figure 9. Navigation page | Figure 10. Taxis available screen |

In figure 11. Notification screen is shown, which shows list with all requests and reservations. It shows requests that are completed, canceled and the requests that are active.

In figure 12. Reservation details screen is presented, which shows all informations about the ride and enables quick access to user’s profile as well as an option of reporting a user/driver.

|  |  |
| --- | --- |
|  | D:\FAX\SW2\SW2015\Mockups\12 Reservation details.png |
| Figure 11. Notification page | Figure 12. Reservation details |

In figure 13. Admin Sing In page is shown, which provides admin with input fields to add credentials. After the fields are filled admin can sign in by clicking the button.

In figure 14. Admin list reports page is presented, which shows all reports submitted by users or taxi drivers, inclusive with timestamp of the reports.

In figure 15. Admin report details page is presented, which shows details about the user/taxi driver for particular selected report, as well as the full description of the enclosed report. Admin can resolve the report by banning the user or taxi driver connected to the report.

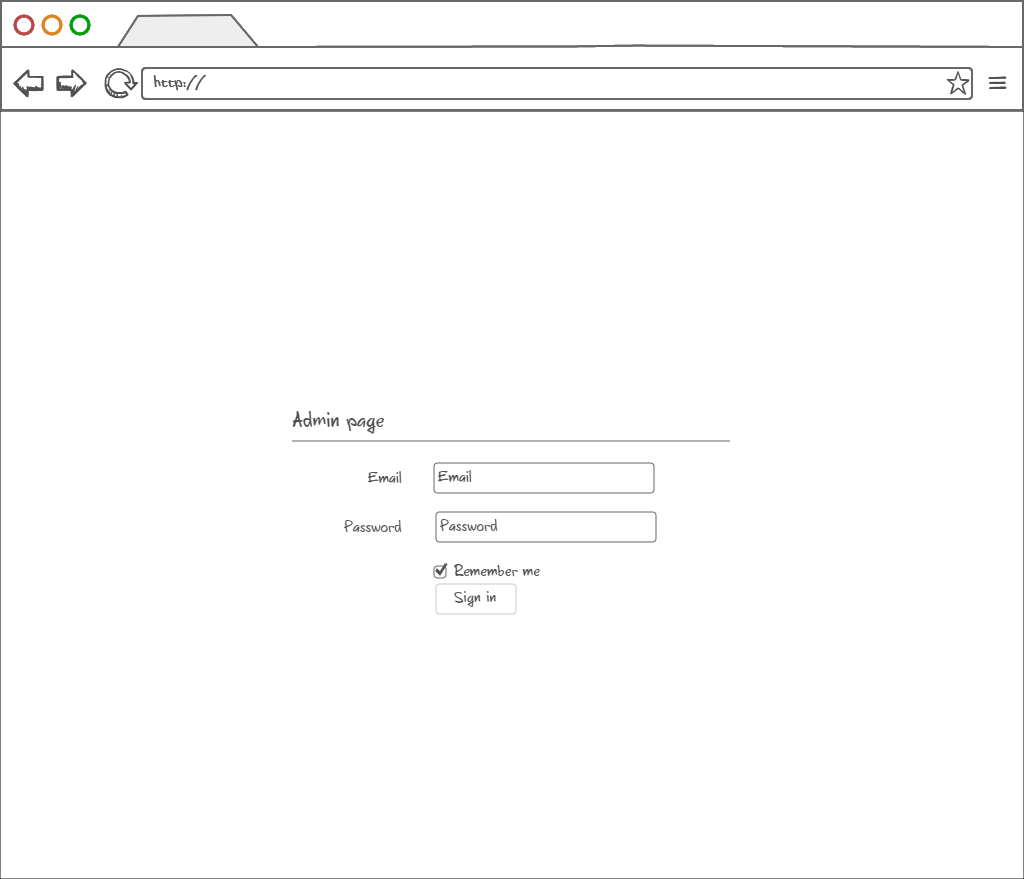


Figure 13. Admin Sign In page



Figure 14.Admin list reports page

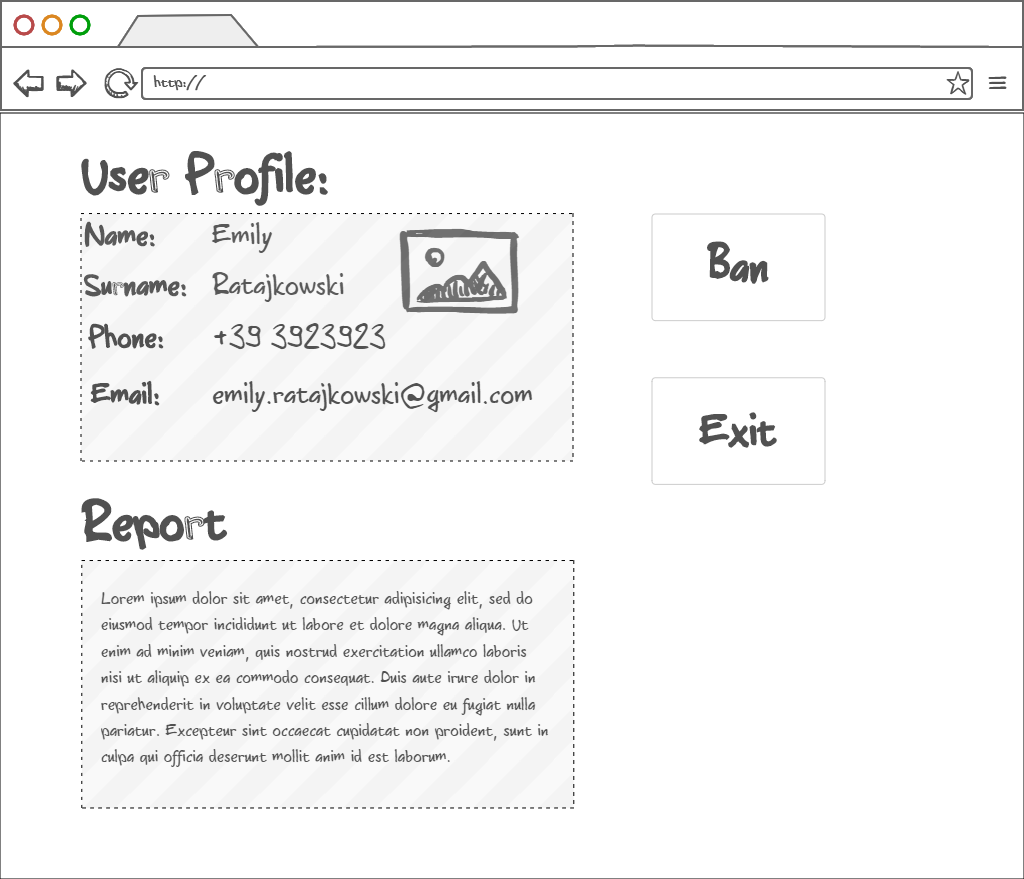


Figure 15. Admin report details page

### API interfaces

In myTaxiService application Google Maps API (https://developers.google.com/maps) is used to represent the map around user’s current location, as well as to provide routes between two given addresses. Google Places API (https://developers.google.com/places) is used to provide the user with autocompleting addresses and to suggest addresses. Email service is used to allow the user to authenticate himself with his email address, a link will be sent to given email address and by clicking the link user confirms his identity. This service is used as a security measure.

### Hardware Interfaces

This project does not support any hardware interfaces.

### Software Interfaces

* Database Management System (DBMS): MySQL
* Programming technologies: HTML5, CSS3, jQuery, PHP
* Application server: Apache
* Operating System (OS): Cross platform

## Functional Requirements

[G1] Allow a guest user to become registered user and choose his role in the myTaxiService system

* [R1] Guest must not be already registered to perform sign up process
* [R2] Guest must choose a username or in our case email address that is not already used by another user
* [R3] Visitor can just see Guest screen
* [R5] Guest can only access to registration form in signup screen
* [D1] Email address used for signing up must be formally correct

[G2] Allow user to sign in to application.

* [R1] User must be already registered to success to sign in process
* [R2] User must know his email address and password used during registration to success login
* [R3] Email address and password inserted during sign in process must be correct
* [R4] Wrong credentials will not grant access to user

[G3] Allow user to manage his profile

* [R1] User must be already registered and signed in the application
* [R2] User must confirm updating process

[G4] Allow user to request a taxi

* [R1] User must be already registered and logged in the application
* [R2] User must specify starting location
* [R3] User must confirm request

[G5] Allow user to make a reservation of a taxi

* [R1] User must be already registered and signed in the application
* [R2] User must specify origin of the ride
* [R3] User must specify destination of the ride
* [R4] User must specify meeting time at least 2 hours before the ride

[G6] Allow user to cancel a ride

* [R1] User must be already registered and signed in the application
* [R2] User can view list of all his rides in notification area
* [R3] User must select an active ride that he wants to cancel and confirm

[G7] Allow user to see available taxi vehicles around him

* [R1] User must be already registered and signed in the application
* [R2] User must have location settings enabled

[G8] Allow user/taxi driver to submit a report

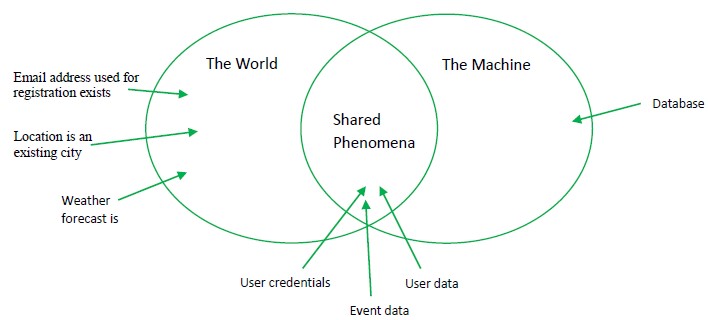
* [R1] User/driver must be registered and signed in
* [R2] User/driver must have an active request or reservation
* [R3] User/driver must select a particular request or reservation he wants to report
* [R4] User/driver must write a description for the report

[G9] Allow driver the option to confirm or decline request for a ride

* [R1] Taxi driver must be already registered and signed in the application
* [R2] Taxi driver must set his availability option in order to receive requests

## The world and the machine

For a first domain analysis of MeteoCal application we use “The World & Macchine” model by M. Jackson & P. Zave. This approach let us identify the entities inside the domain that interact with the application (“The World”), entities to be developed (“The Machine”) and the intersection (“Shared Phenomena”) between the world and the application, that are all world informations known or managed direclty by the application.



## Scenarios

### Scenario 1

Emily is running late to a Saturday evening house party. The metro is about to close, so she needs to find a ride quickly. She has heard that there is new well-functioning application called myTaxiService and she wanted to give it a try. She picked up her smart phone and typed in the web address of the app. A login page has appeared and she had to register in order to use it. Emily quickly typed in her personal information and authenticated by email. After that she requested a ride from her current location, and instantaneously received a notification with ETA and code of the incoming taxi. Taxi arrived in few minutes and her dinner hasn’t got cold.

### Scenario 2

Dan and his friends are planning to go play football in an indoor pitch court. So Dan searched and found free court today from 16 to 17 o’clock so he booked that court. His friends live close to the court or near the metro station. Dan isn’t that lucky so he must find a ride that will be secure and that will respect his time schedule. His friends told him about an application myTaxiService that can solve his problems. He easily created an account and found an option of reserving a taxi for the time of the match. He made a reservation 3 hours before the meeting point and received a confirmation of the reservation. 10 minutes before the ride he received a message containing the information about the assigned vehicle. That night his team won the game.

### Scenario 3

Economist Jack has lost his job because of the world economic crisis and he found a part-time job at city taxi service. The taxi service uses well-functional cross platform web application. So jack registered as a driver and started to work hardly. He was assigned to work in zone 1, close to the central train station. Being assigned at that area he had a lot of calls from passengers. Around noon when is the rush hour he was really busy so he had to reject some rides when his vehicle was occupied. As soon as he finishes a ride he change his status to available so he could get a contacted by the taxi system about other rides.

### Scenario 4

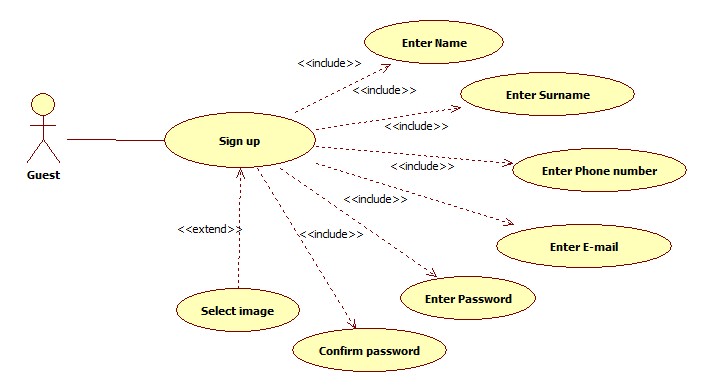
Catherine as a well experienced F1 driver got hurt in a car accident and after the recovery her doctor told her that she has to give up that extreme sport and find a nice, less invasive job. She thought about it intensively and she figured out that she couldn’t live her life without driving a car so with her resume she managed to get a job easily as a taxi driver. After a few days working as a taxi driver she still couldn’t remember all the street addresses, but luckily myTaxiService application when accepting a request for a ride offers an navigation option from her current location to the passenger location and after that as well as the route from starting to destination point. When Catherine found out about this convenience she felt so relieved that she didn’t have to remember all the routes around the city.

## UML Models

### Use Case

After stating all the desired features, goals and requirements, and describing possible scenarios we can identify some use cases. The diagrams are shown and described below.

#### Accessing the application

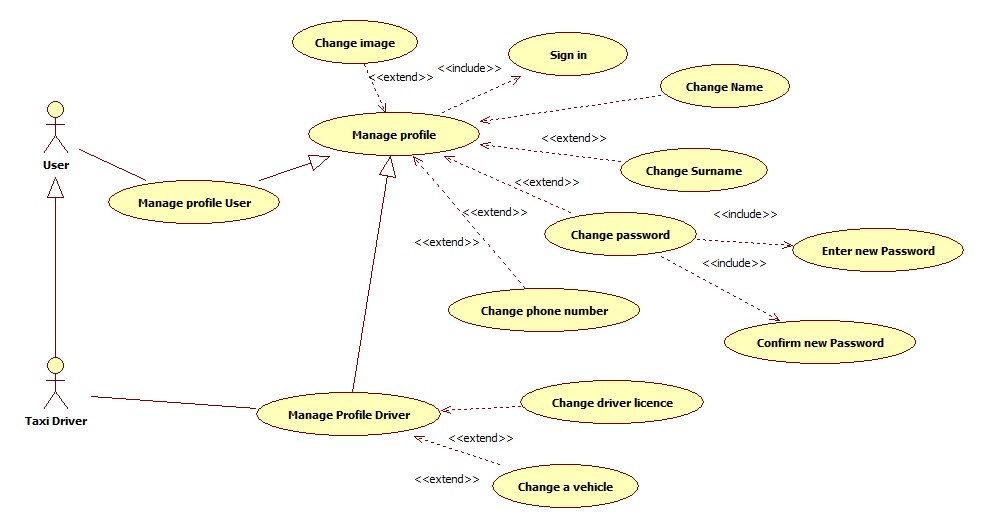


|  |  |  |
| --- | --- | --- |
| **Name** | **SignUp** | |
| **Actors** | | Guest | |
| **Entry conditions** | | The guest has never registered in the system | |
| **Event Flow** | | * Guest navigates to the myTaxiService homepage containing a form asking him to enter basic information, email, password, phone number and image which is optional * Upon the form submission system checks if the entered data is valid i.e. if all the fields are filled and email is not already in use. * If the entered data is valid, the user will get confirmation email with which he will be automatically logged in to the system and redirected to his personal home page | |
| **Exit Conditions** | | The information about the new user is correctly stored, a welcome email is sent and the user can use all functions of myTaxiService | |
| **Exceptions** | | If the data provided by the user is not valid, an appropriate message is displayed and the guest is asked to fill the missing fields or correct specific fields | |



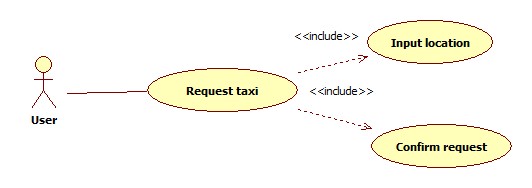
|  |  |  |
| --- | --- | --- |
| **Name** | **SignIn** | |
| **Actors** | | Guest |
| **Entry conditions** | | The guest has already registered in the system and knows his email and password |
| **Event Flow** | | * After navigating to myTaxiService homepage containing a form asking him to enter his email and password * Upon the form submission system checks if the username and password match an existing user * If the entered information is valid, the user is signed in to the system and redirected to his personal page |
| **Exit Conditions** | | The user is granted access to all of the myTaxiService's functionalities |
| **Exceptions** | | If he information the user entered is not valid, an appropriate message is displayed and the guest is asked to enter his username and password again |

#### Manage profile



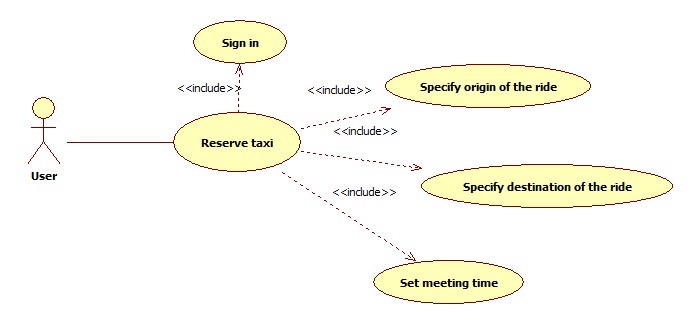
|  |  |  |
| --- | --- | --- |
| **Name** | **Manage Profile** | |
| **Actors** | | User |
| **Entry conditions** | | The user is logged in |
| **Event Flow** | | * User chooses the Manage Profile option from the drop down menu in home screen * User is redirected to a Manage profile screen where he/she can change the following information:   + name   + surname   + password   + phone number   + image (optional) * User submits the filled form |
| **Exit Conditions** | | The new user’s data are correctly stored and the account is successfully edited |
| **Exceptions** | | If the user submits the form leaving any field empty, an appropriate message will be displayed and the user will be asked to fill out the missing information |

#### Requesting a taxi



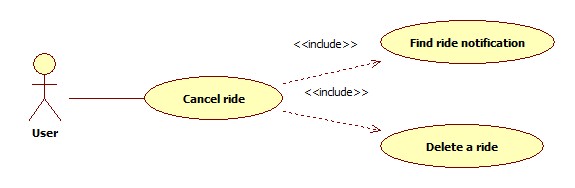
|  |  |  |
| --- | --- | --- |
| **Name** | **Request taxi** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is logged in | |
| **Event Flow** | | * User chooses the *Request taxi* option from drop down menu of the home screen * User is redirected to a Request screen where he/she can select desired location on the map where to catch a taxi * After selecting a location user submits request simply by clicking request button | |
| **Exit Conditions** | | The request is sent to myTaxiService system, confirmation with code of incoming taxi and ETA is sent back to user | |
| **Exceptions** | | If the user doesn’t select location on the map, an appropriate message will be displayed and the user will be asked to choose starting point location on the map | |

#### Reserving a taxi



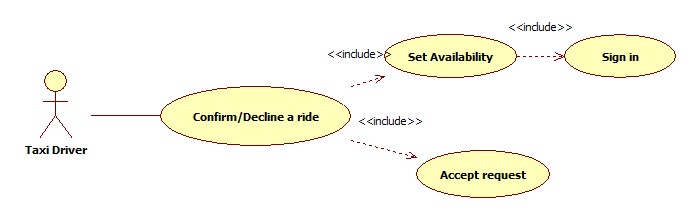
|  |  |  |
| --- | --- | --- |
| **Name** | **Reserve taxi** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is signed in | |
| **Event Flow** | | * User chooses the *Reserve taxi* option from drop down menu of the home screen * User is redirected to a Reserve screen where he/she must:   + Specify Origin of the ride   + Specify destination of the ride   + Set meeting time * After selecting a location user submits request simply by clicking Ok button | |
| **Exit Conditions** | | The request of reservation is send to myTaxiService system. User receives confirmation of the reservation immediately. And 10 minutes before the ride he/she receives message with the information of the assigned vehicle | |
| **Exceptions** | | * If user doesn’t fill all the fields the warning message will be displayed and the user will be asked to fill the missing information * If the doesn’t reserve a taxi at least 2 hours before the ride. The appropriate message will be displayed and the user will be asked to set new time | |

#### Canceling a taxi



|  |  |  |
| --- | --- | --- |
| **Name** | **Cancel ride** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is signed in and requested a ride | |
| **Event Flow** | | * User opens the *Notifications* from drop down menu of the home screen * User is redirected to a Notifications screen where he/she can delete the active requests or reservations for a ride | |
| **Exit Conditions** | | The cancel request is sent to myTaxiService system, after that the system responds with confirmation pop-up | |
| **Exceptions** | | * User cancels request while riding in a taxi. The appropriate error message will be displayed on users screen | |

#### Confirm/Decline a ride



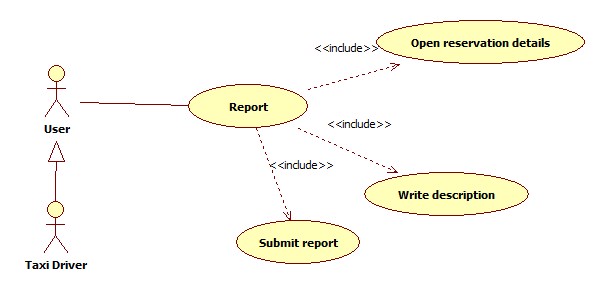
|  |  |  |
| --- | --- | --- |
| **Name** | **Confirm/Decline a ride** | |
| **Actors** | | Taxi Driver | |
| **Entry conditions** | | The driver is signed in, his availability status button is set to YES and he received a request | |
| **Event Flow** | | * Taxi driver receive pop-up message with a request for a ride * Taxi driver accept or declines the ride by choosing appropriate option on a pop-up window | |
| **Exit Conditions** | | If taxi driver confirmed a ride, navigation screen is shown to him. Otherwise, his state is unchanged and he awaits for new requests | |
| **Exceptions** | | If driver has some problem with the car and can’t reach destination point in the agreed time he/she can cancel the ride. The system sends a notification to the passenger about the issue and new assigned vehicle | |

#### Check taxis available



|  |  |  |
| --- | --- | --- |
| **Name** | **Check Taxis Available** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is signed in | |
| **Event Flow** | | * User chooses the Check available vehicle option from the drop down menu * User is navigated to Taxis available screen | |
| **Exit Conditions** | | - | |
| **Exceptions** | | * If user’s GPS service is not functioning, he will be notified by a pop-up message stating that his GPS is not working | |

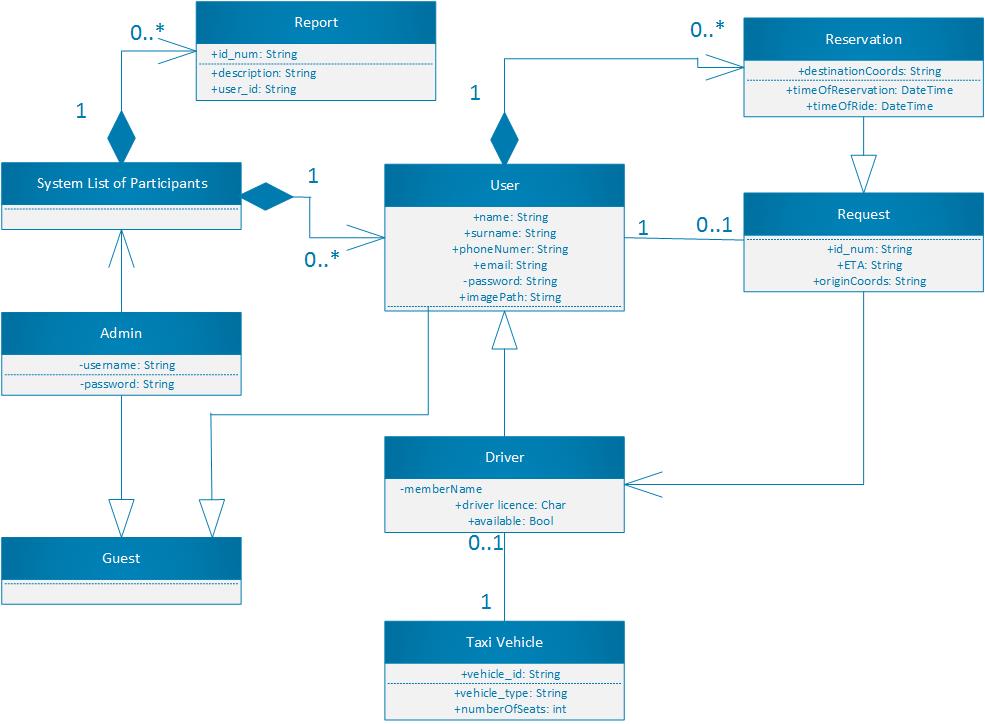
#### Report user/driver



|  |  |  |
| --- | --- | --- |
| **Name** | **Report** | |
| **Actors** | | User |
| **Entry conditions** | | The user is signed in |
| **Event Flow** | | * User opens reservation details page * User chooses report option and is redirected to report screen * User writes a report and then submits it to the system |
| **Exit Conditions** | | The user receives confirmation of sent report |
| **Exceptions** | | If user did not write the report description, he will receive an error message pointing out about the missing description |

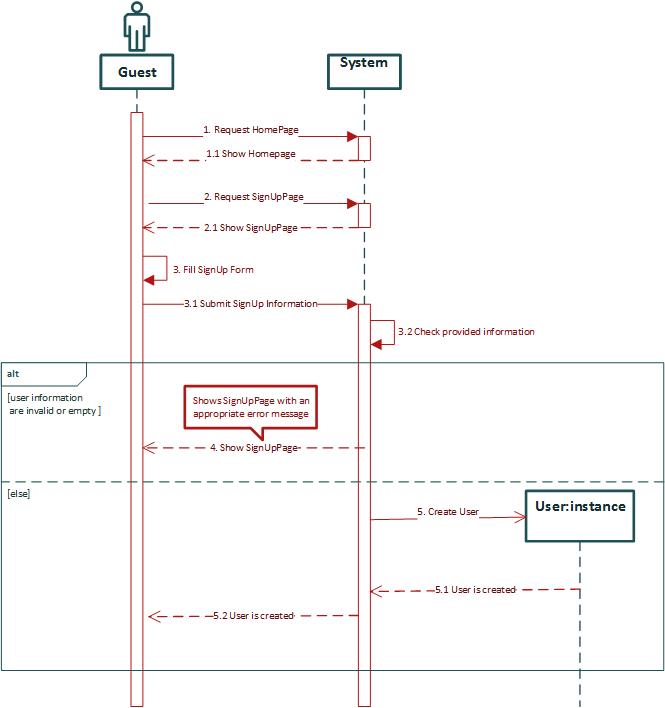
### Class Diagram

Here is presented the class diagram. This diagram will be updated during the developing process.

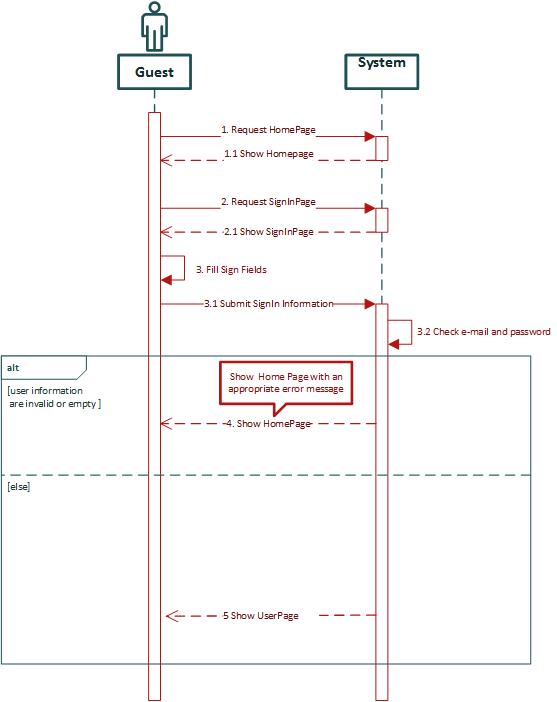


### Sequence Diagram

#### SignUp

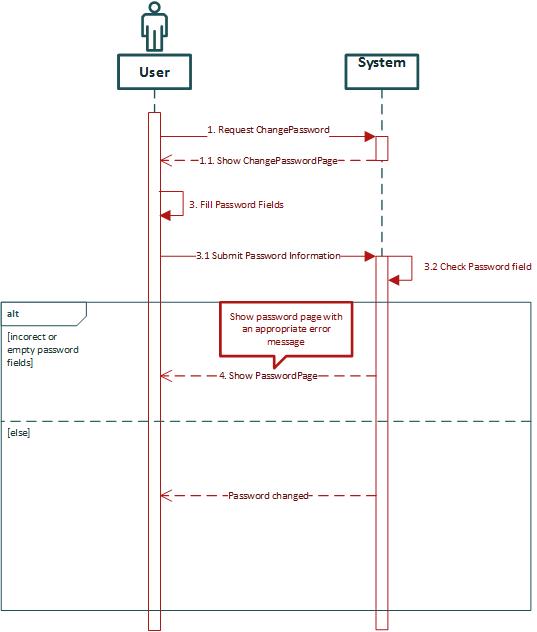


#### SignIn

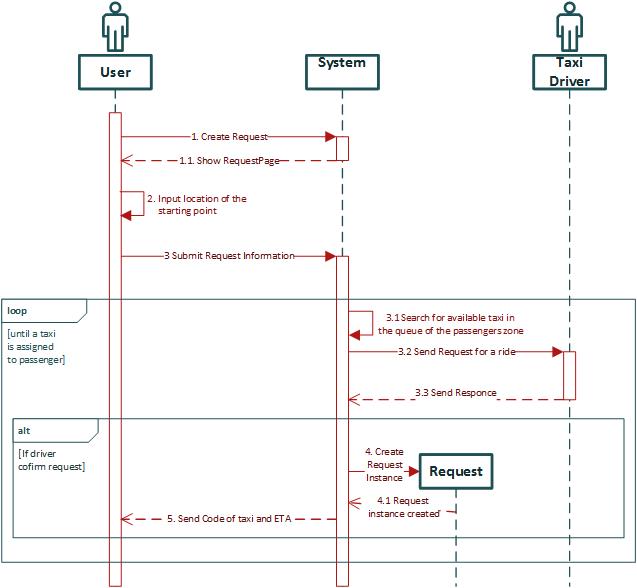


#### Manage profile

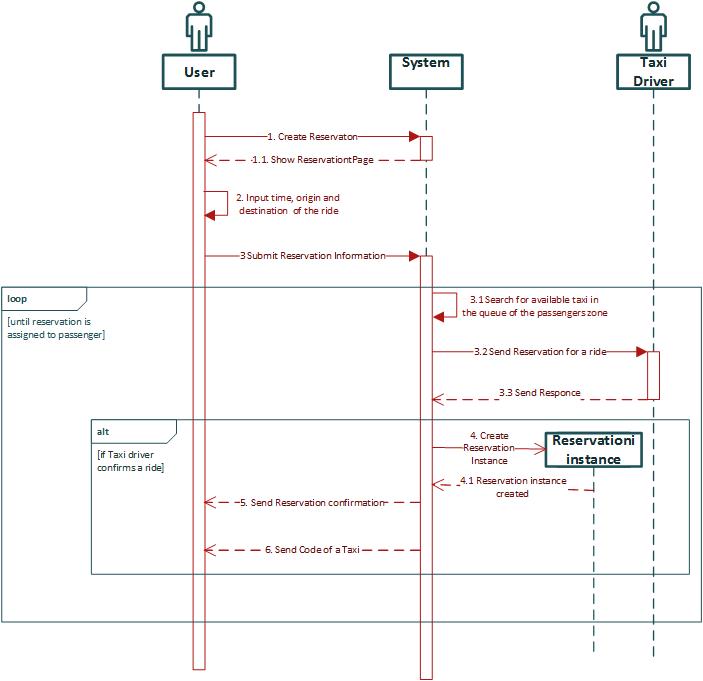
In this sequence diagram as example of managing user profile, Change password sequence of actions is shown.



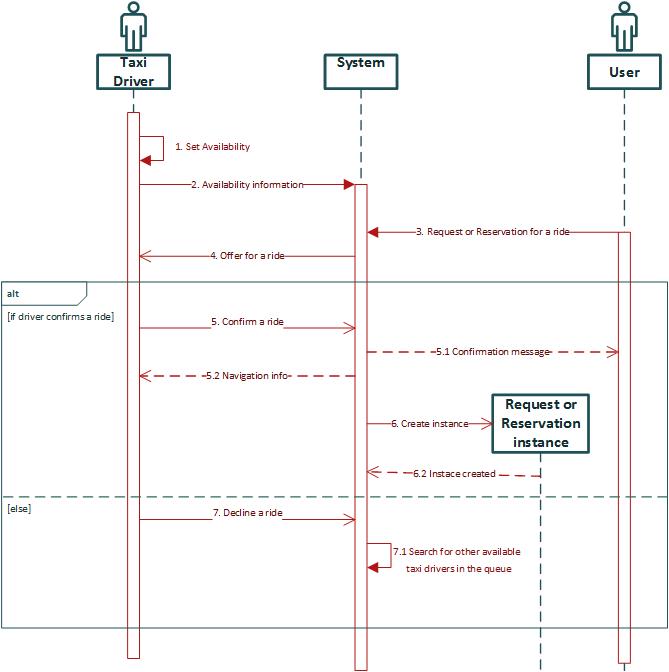
#### Request a ride



#### Reserve a ride

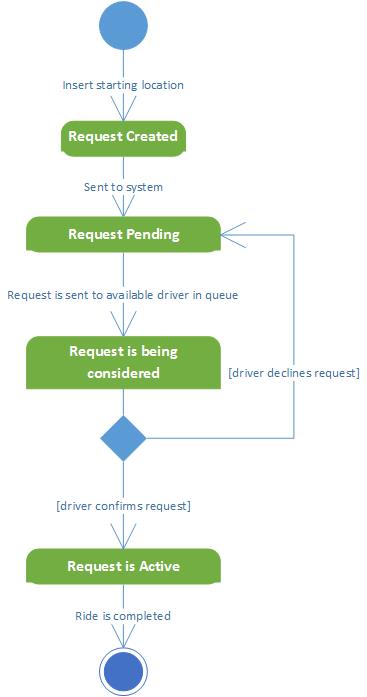


#### Confirm/Decline a ride

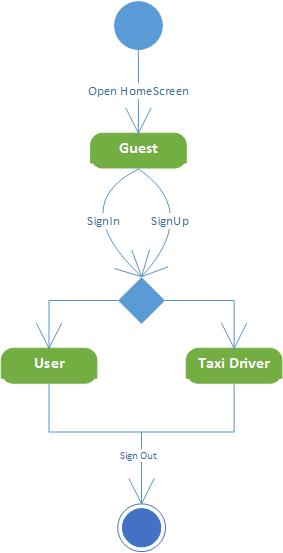


### State Chart diagrams

#### Request lifecycle



#### Visitor lifecycle



## Non Functional Requirements

### Performance Requirements

Performance of the system should be good enough to provide the user with fast responding software system. Response time should be small enough to enable good user experience.

### Software System Attributes

* *Availability,* the 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,* the application could be run on device with any OS that has access to Internet and has a web browser.
* *User Interface,* the web application should be intuitive so even the nontechnical users can use the system as simply and efficiently as possible.

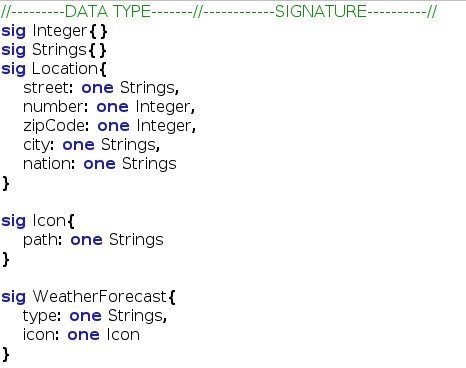
# Appendix

## Alloy

The complete alloy file (.als) could be find on our google code repository. The following alloy model presented is created using the class diagram. We try to divide the code in part dividing signature from fact, assert and predicates. In the last part there are the generated word.

4.1.1 Data Type

This is the definition of data type.



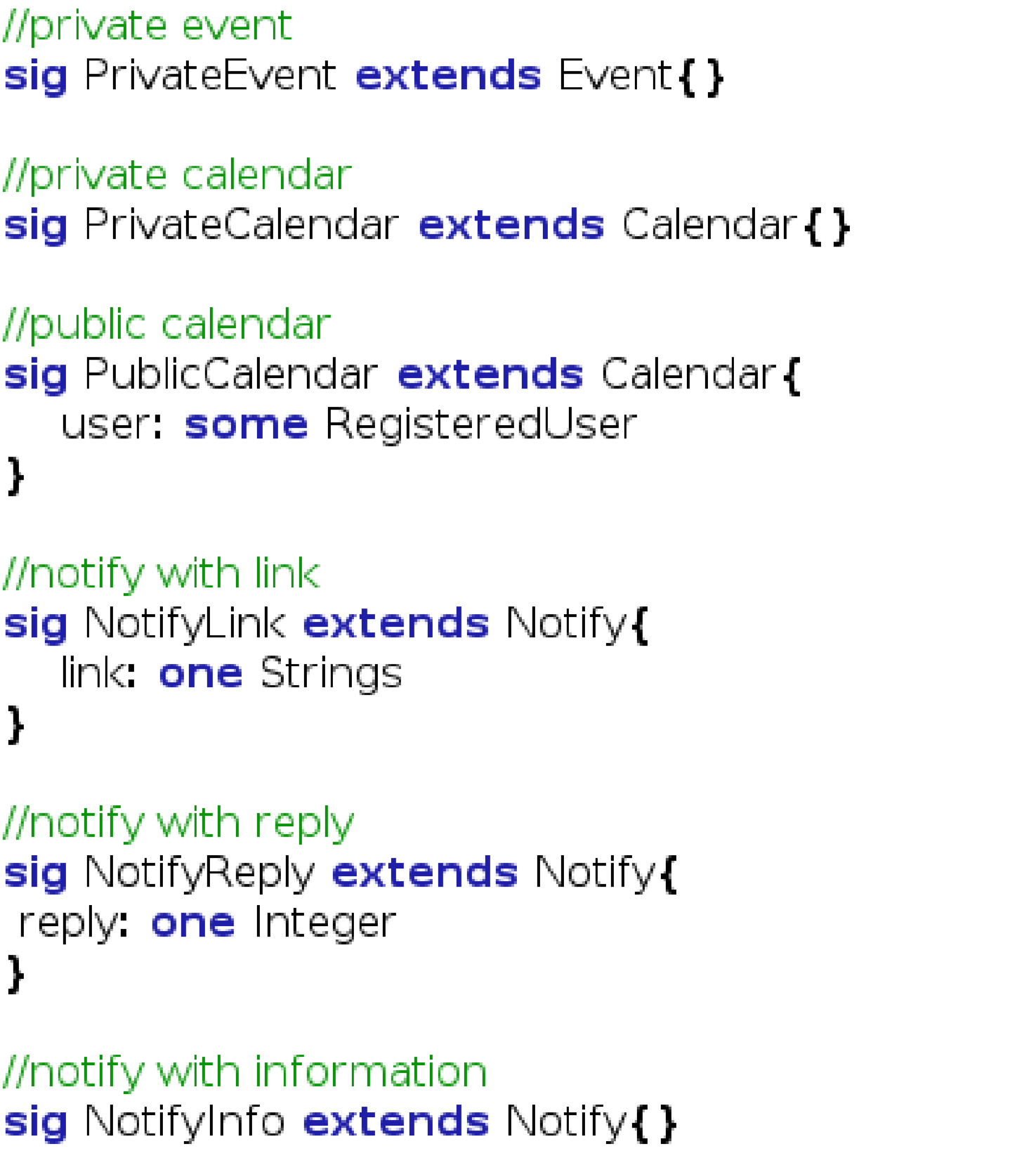
4.1.2 Abstract Entity

This is the definition of abstract entity.



4.1.3 Abstrac Entity Implementation and Signature This is the implementation of some abstract entity and other signature.





4.1.4 Fact

This

is

the

fact

part

that

defines

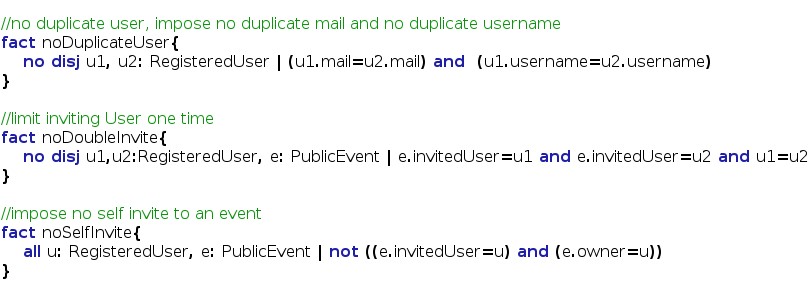
the

costraint

of

the

class.



4.1.5 Assert

In

this

last

code

part

is

presented

the

assert

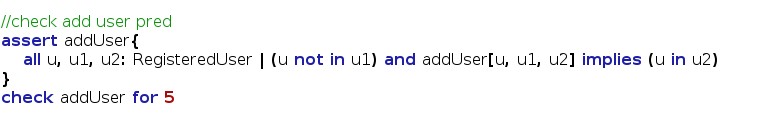
used

to

verify

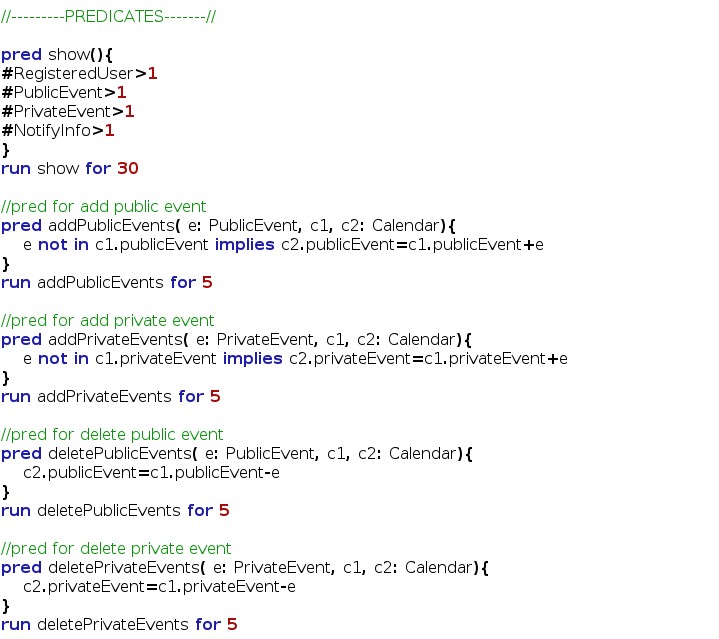
the

model.



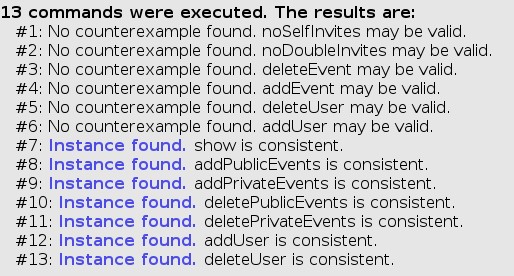
4.1.6 Predicates

This is the predicates used with the previous assert to verify the model.



4.1.7 Result

This screenshot of the Alloy Analizer software that shows the consistence of the model in all part.



4.1.8 Generated world

Here is presented the generated world using the alloy verification software. First diagram is the predicate show() for 4 case. The second is the predicate show() for 8 case. For more case the model will be more and more complex and difficult to read. The third graph is generated from the deleteUser predicate. We decide to not attach all diagram because they are very complex and would not give other information or help anyone want to read this document.