

Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “myTaxiService”

Design Document

Version 0.1

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

17 November 2015

Contents

[1 Introduction 4](#_Toc435522730)

[1.1 Purpose 4](#_Toc435522731)

[1.2 Scope 4](#_Toc435522732)

[1.3 Glossary 5](#_Toc435522733)

[1.4 Reference Documents 5](#_Toc435522734)

[1.5 Document Overview 5](#_Toc435522735)

[2 Architectural design 6](#_Toc435522736)

[2.1 Overview 6](#_Toc435522737)

[2.2 High level components and their interaction 6](#_Toc435522738)

[2.3 Component view 6](#_Toc435522739)

[2.4 Deployment view 6](#_Toc435522740)

[2.5 Runtime view 6](#_Toc435522741)

[2.6 Component interfaces 6](#_Toc435522742)

[2.7 Selected architectural styles and patterns 6](#_Toc435522743)

[2.8 Other design decisions 6](#_Toc435522744)

[3 Algorithm design 9](#_Toc435522745)

[Software Interfaces 9](#_Toc435522746)

[4 User interface design 28](#_Toc435522747)

[5 Requirement traceability 28](#_Toc435522748)

[6 References 28](#_Toc435522749)

# Introduction

## Purpose

The purpose of this document is to provide a comprehensive description of the structure of the myTaxiService system. It will state and analyze the design decisions made in order to satisfy all the requirements stated in the Requirements Analysis and Specification Document (RASD). This document is meant mainly as a guideline for developers of the software in question.

## 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:

|  |  |
| --- | --- |
| *CSS* | Cascading Style Sheets |
| *DD* | Design Document |
| *DB* | Database |
| *DBMS* | Database Management System |
| *ER* | Entity-Relationship |
| *HTML* | Hypertext Markup Language |
| *HTTP* | Hypertext Transfer Protocol |
| *MVC* | Model View Controller |
| *RASD* | Requirement Analysis and Specification Document |
| *UML* | Unified Modeling Language |

## Reference Documents

* IEEE Design Document template
* Specification Document: myTaxiService Project AA 2015-2016.pdf
* RASD myTaxiSevice

## 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: Architectural Design, gives an overview of how  and why the system was decomposed, and  how the  individual parts work together
* Chapter 3: Algorithm Design, description of the most relevant algorithms of the software system
* Chapter 4: User interface Design, overview on how the user interfaces of your system will look
* Chapter 5: Requirements traceability, gives an overview of how the requirements defined in RASD map into the design elements defined in DD.
* Chapter 6: References

# Architectural design

## Overview

Identify modules in the software system and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality.

## High level components and their interaction

## Component view

## Deployment view

## Runtime view

## Component interfaces

## Selected architectural styles and patterns

## Other design decisions

*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

# Algorithm design

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

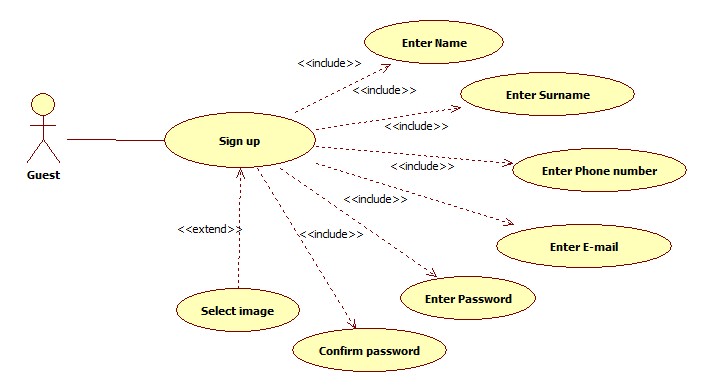
* [R1] Taxi driver must be already registered and signed in the application
* [R2] Taxi driver must set his availability option in order to receive requests

UML Models

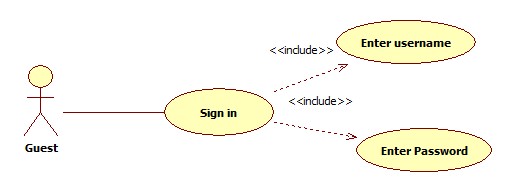
Use Case

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

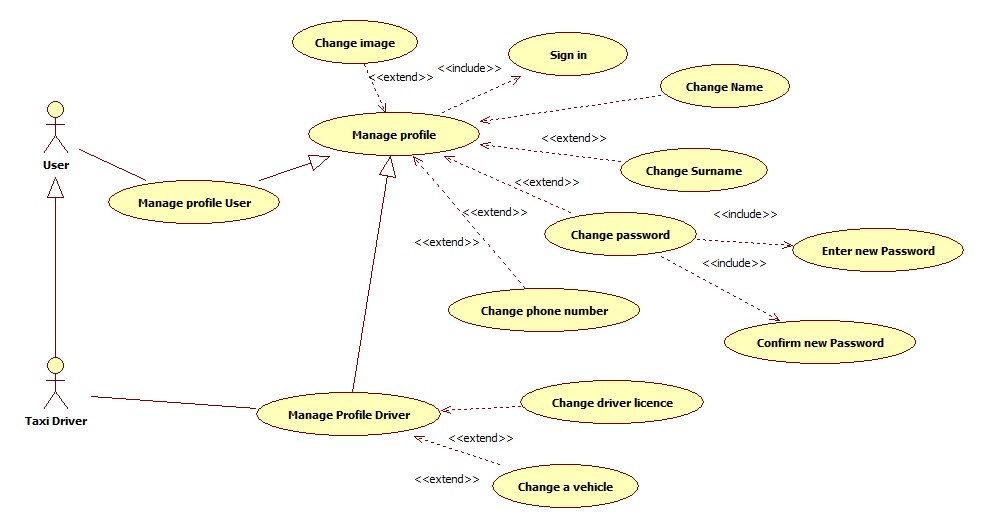


|  |  |  |
| --- | --- | --- |
| **Name** | **SignUp** | |
| **Actors** | | Guest | |
| **Entry conditions** | | The guest has never registered in the system | |
| **Event Flow** | | * Guest navigates to the myTaxiService homepage containing a form asking him to enter basic information, email, password, 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 | |



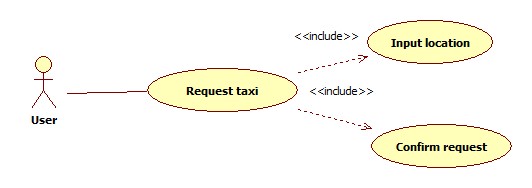
|  |  |  |
| --- | --- | --- |
| **Name** | **SignIn** | |
| **Actors** | | Guest |
| **Entry conditions** | | The guest has already registered in the system and knows his email and password |
| **Event Flow** | | * After navigating to myTaxiService homepage containing a form asking him to enter his 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



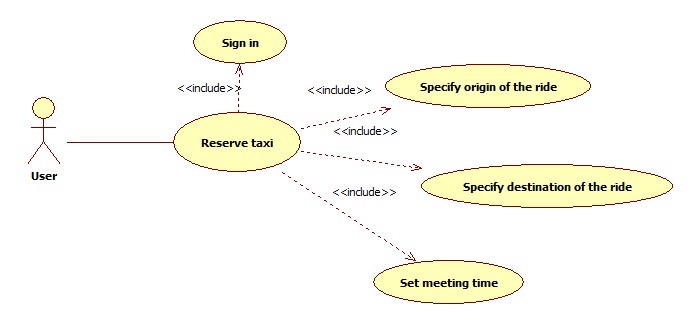
|  |  |  |
| --- | --- | --- |
| **Name** | **Manage Profile** | |
| **Actors** | | User |
| **Entry conditions** | | The user is logged in |
| **Event Flow** | | * User chooses the Manage Profile option from the drop down menu in home screen * User is redirected to a Manage profile screen where he/she 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



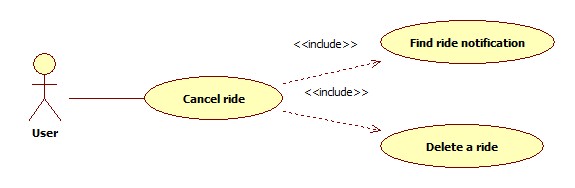
|  |  |  |
| --- | --- | --- |
| **Name** | **Request taxi** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is logged in | |
| **Event Flow** | | * User chooses the *Request taxi* option from drop down menu of the home screen * User is redirected to a Request screen where he/she can 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



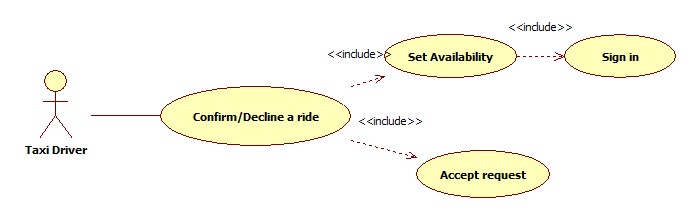
|  |  |  |
| --- | --- | --- |
| **Name** | **Reserve taxi** | |
| **Actors** | | User | |
| **Entry conditions** | | The user is signed in | |
| **Event Flow** | | * User chooses the *Reserve taxi* option from drop down menu of the home screen * User is redirected to a Reserve screen where he/she must:   + 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



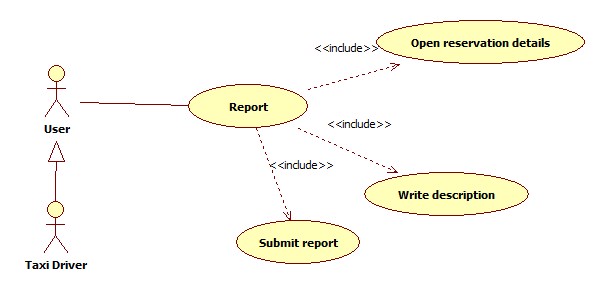
|  |  |  |
| --- | --- | --- |
| **Name** | **Confirm/Decline a ride** | |
| **Actors** | | Taxi Driver | |
| **Entry conditions** | | The driver is signed in, his availability status button is set to YES and he received a request | |
| **Event Flow** | | * Taxi driver receive pop-up message with a request 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 | |

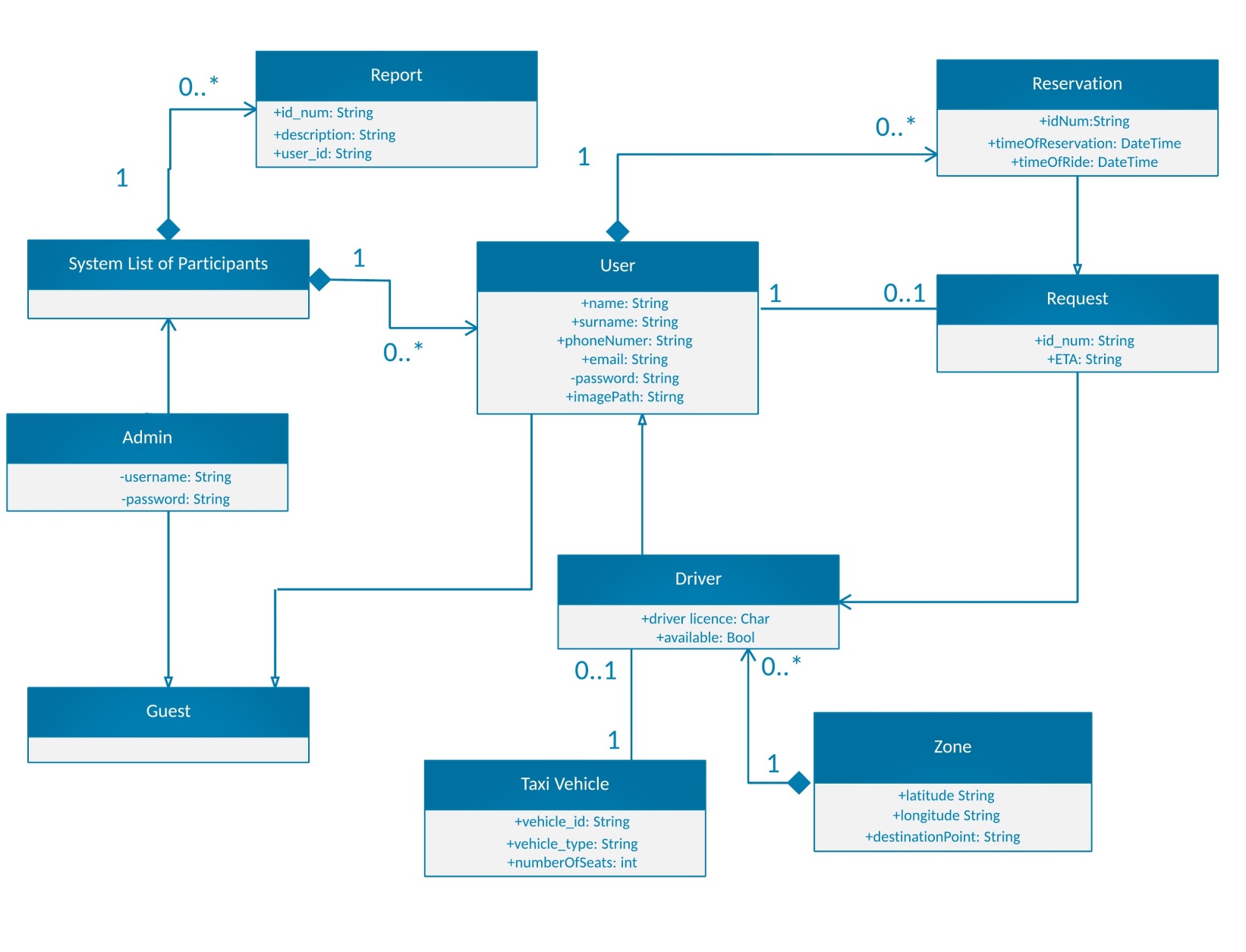
Report user/driver



|  |  |  |
| --- | --- | --- |
| **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 |

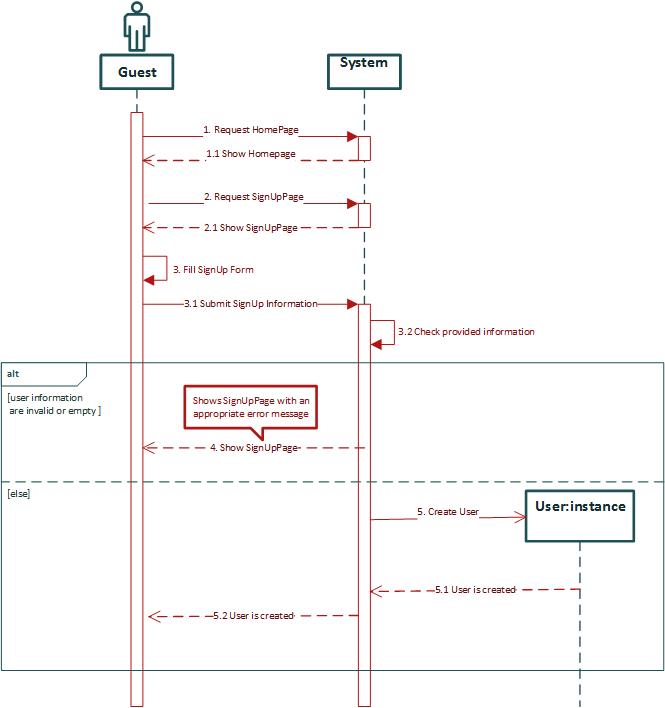
Class Diagram

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

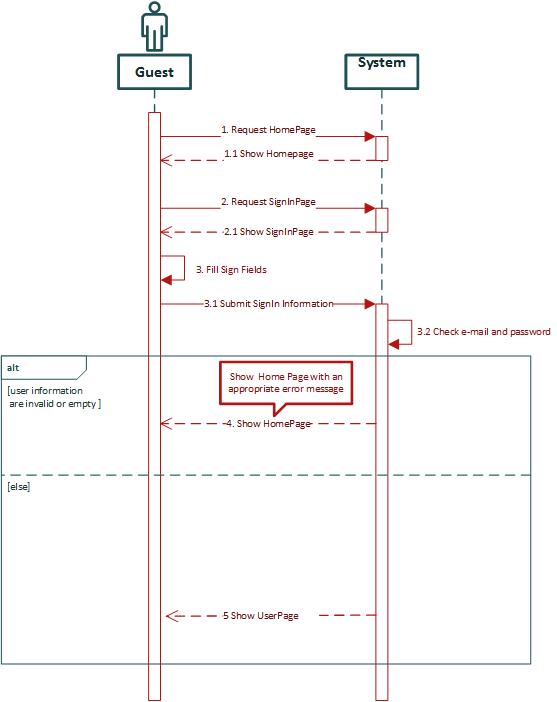


Sequence Diagram

SignUp

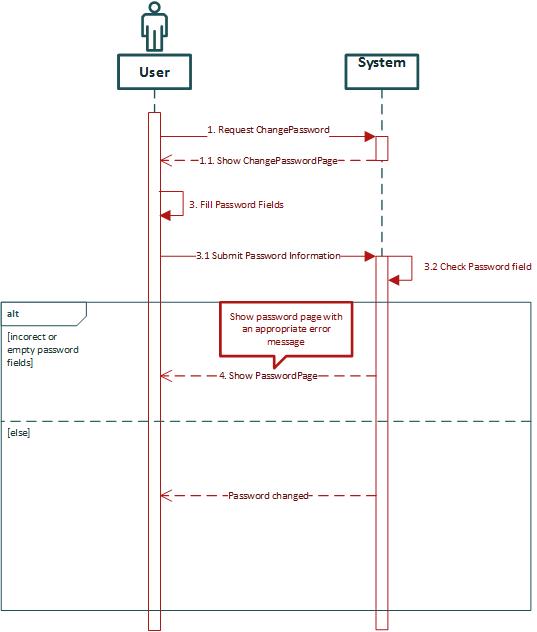


SignIn

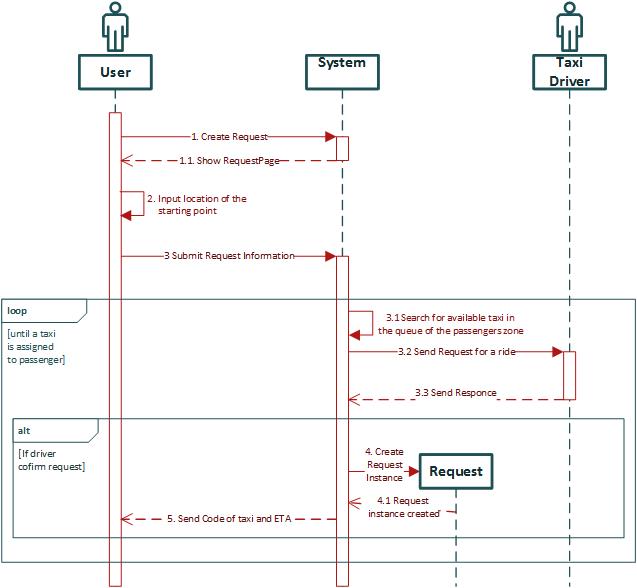


Manage profile

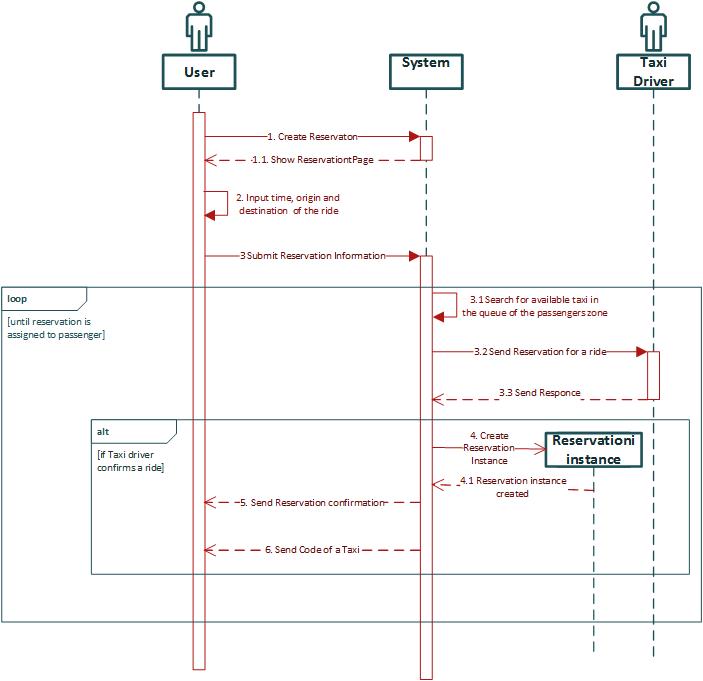
In this sequence diagram as example of managing user profile, Change password sequence of actions is shown.



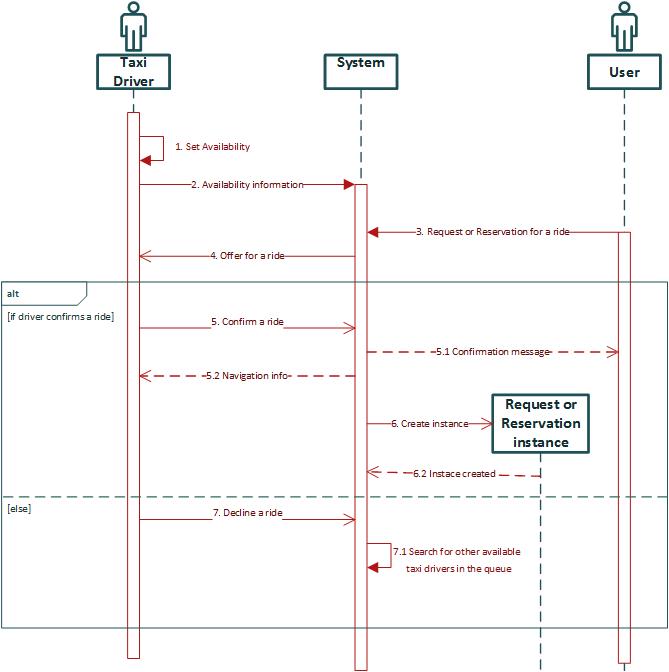
Request a ride



Reserve a ride



Confirm/Decline a ride



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.

# User interface design

Refer to the chapter 3.1 of RASD documents. In chapter 3.1 are shown user interfaces of myTaxiService system.

# Requirement traceability

# References