MyTaxiService
Requirement Analysis and Specification Document

Mite Ristovski   Dushica Stojkoska

Software Engineering 2 Project

Politecnico di Milano

5th School of Engineering

Milan, November 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this Requirement Analysis and Specification Document is to introduce and describe the general functionalities of the software product myTaxiService, which was assigned as a project in the Software Engineering 2 course.

The intended audience of this document is the people participating in the software engineering course. Among active participants in this course are the students, professors and professor assistants. Not only shall the document serve as a reference for the developers to follow the requirements, but it will also serve to the testers to check whether the stated requirements and goals are met or not.

The main functionalities that myTaxiService software product will offer are:

- Simplified and easy access to taxi services in large city;
- Guaranteed fair management of taxi distribution.

1.2 Scope

The software product that is going to be developed is myTaxiService application. This software product is intended to ease access of passengers to the taxi service in the city.

MyTaxiService will allow users, or the passengers, to make requests for taxi services, as well as reservations of taxi for specific time periods. The system will provide fair distribution and management of taxis for the passengers, through automatical computation of distribution of taxis. This distribution of taxis will be based on GPS information each taxi sends to the system. Taxi drivers will be able to confirm or decline passenger's request. After confirmation of drivers for a request, the system will send a notification to the passenger, containing coresponding taxi code and the waiting time.

Moreover, the system will offer programmatic interfaces - API, to enable development of additional services to the basic ones available in the application.

The software product will have the following general objectives:

- Allow users (passengers and taxi drivers) to register on the system and have their own profiles.
- Allow passengers to request a taxi service.
- Allow passengers to reserve a taxi service for specific date and time period.
- Allow taxi drivers to confirm or decline particular requests.
- Allow passengers to be notified when their request is confirmed.

MyTaxiService will have the following general functionalities:

- **Users**

myTaxiService will manage registering, loging in and loging out of users. Only registered users can use services of the application.

- **Profiles**

myTaxiService will manage personal data of different types of users. Users can be passengers, taxi drivers and administrators.

- **Requests**

myTaxiService will manage requests for available taxi made by the passengers.

- **Reservations**

myTaxiService will manage reservations for specific date and time period made by passengers.

- **Confirmations of requests and reservations**

The system needs to manage fair distribution of taxi services, to respond to every user's request and reservation.

Limitations of this software product (which can be developed in the future versions) will be:
Payment - myTaxiService will not offer any form of online payment for taxi services.

Goals of the software product are:
[G1] Users can register on the system.
[G2] Users can consult and update their profiles.
[G3] Passengers can make requests for available taxi.
[G4] Passengers can make reservations.
[G5] Taxi drivers can accept request or decline it.
[G6] The system informs passengers for confirmation of available taxi by sending code of the corresponding taxi and waiting time.
[G7] The system should consistently track the distribution of taxis in the specific zones, based on the GPS information it receives from each taxi.
[G8] The system offers programmatic interfaces for enabling development of additional services.

1.3 Definitions, Acronyms and Abreviations

1.3.1 Definitions

Keyword	**Definition**
User	All registered users in the system, more specifically: administrators, passengers and taxi drivers.
Administrator	Administrator has a role to manage the system and register taxi drivers to the system.
Passenger	Users that request the service of the system.
Taxi driver	Users that enable the services, they accept or decline requests from the passengers.
Request	Asking for available taxi.
Reservation	Asking for available taxi for a specific date and time period.

Table 1: Definitions

1.3.2 Acronyms and abreviations

Acronym or abreviation	**Definitions**
HTML	HyperText Markup Language
RASD	Requirements Analysis and Specification Document
G	Goal
FR	Functional Requirement
NFR	Non-functional Requirement
DBMS	Database Management System
AS	Application Server
JEE	Java Enterprise Edition
JVM	Java Virtual Machine
RAM	Random Access Memory
TCP	Transmission Control Protocol

Table 2: Acronyms and abreviations

1.4 Reference Documents

1. IEEE Recommended Practice for Software Engineering Requirements Specification

(http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?reload=true&punumber=5841)

1. Alloy model file: model.als.

1.5 Overview

This document is organized in the following way:

1. **Introduction:** A synopsis of the software product to be developed is provided in this section.

2. **Overall description:** The general factors that affect the software product are described in this section. The requirements for the software product are also included. Different interfaces, operations, functions, user characteristics, constraints, assumptions and dependencies, future requirements are included as well.

3. **Specific requirements:** This section contains all the analysis done to the project requirements. It describes all of the software requirements to a level of detail sufficient to be externally perceivable. It contains the different use cases, interfaces, scenarios and sequence diagrams, among others.

4. **Appendices:** Additional supporting information is provided in this section. More specifically, the contribution of alloy model to the analysis model and requirement analysis is shown.

2 Overall Description

This section describes the general factors that affect the software product and its requirements, and it provides different kind of interfaces (system interfaces, user interfaces, etc.), memory requirements, operations, functionality, user characteristics and constraints.

2.1 Product Perspective

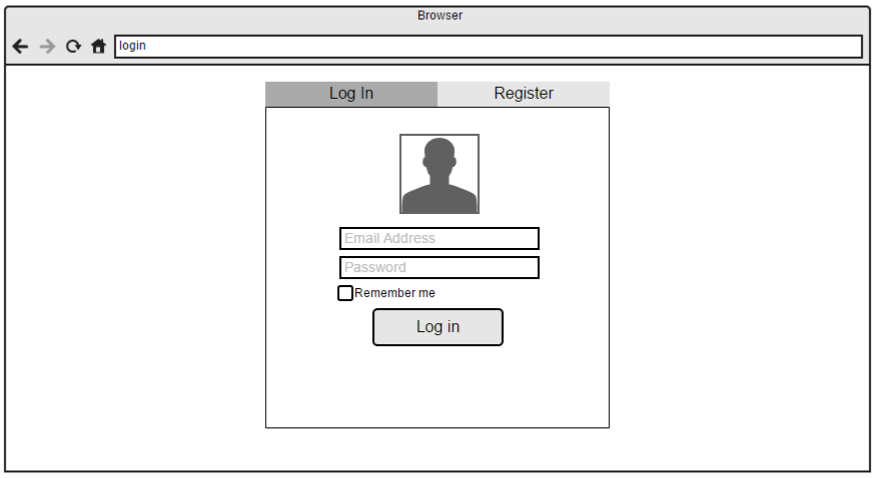
The current development will be independent from other systems. This software product is self-contained software that can evolve during time.

2.1.1 System Interfaces

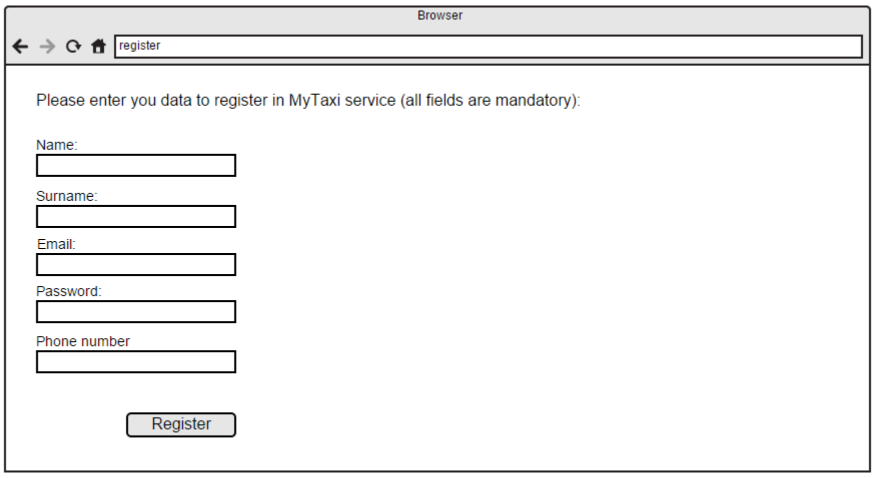
Beside user interfaces for passengers and taxi drivers, this software product additionally offers a programmatic interface for enabling development of additional services.

2.1.2 User Interfaces

User interfaces will have the following characteristics. Interface formats are based on the general recommendations for designing user interfaces. Among this recommendations are the number of options displayed to the user, the depth of the navigation when interacting with the system, and fast access to all the functionalities allowed.



User interface 1: Log in



Please enter you data to register in MyTaxi service (all fields are mandatory):

Name:

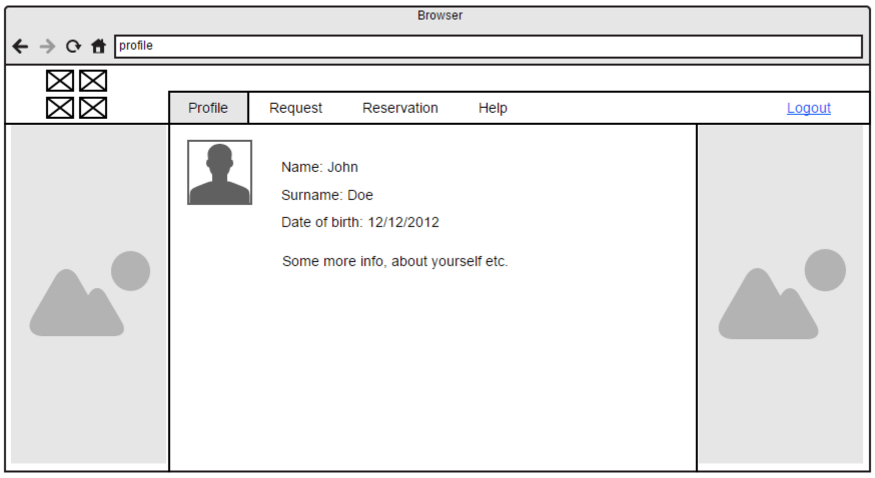
Surname:

Email:

Password:

Phone number

User interface 2: Register form



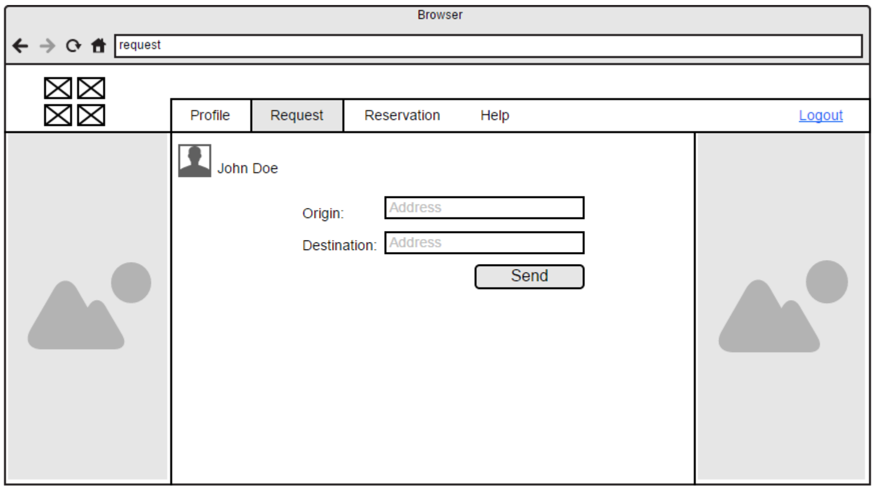
Name: John

Surname: Doe

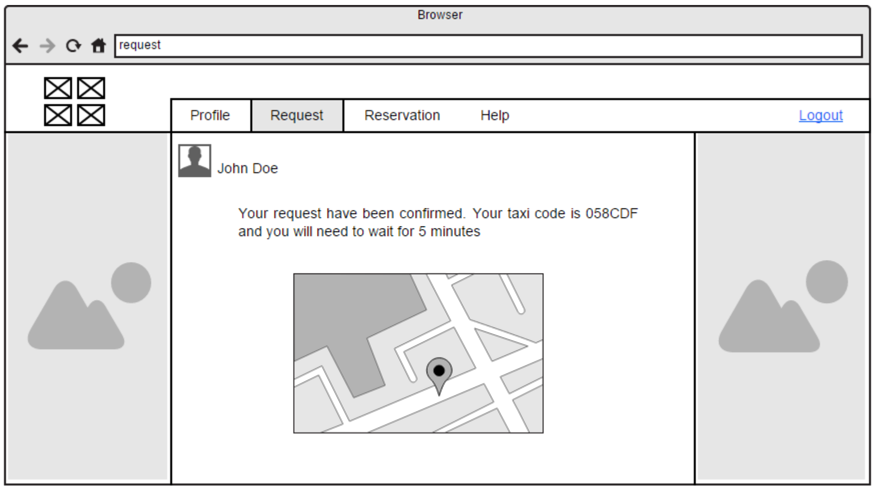
Date of birth: 12/12/2012

Some more info, about yourself etc.

User interface 3: User profile

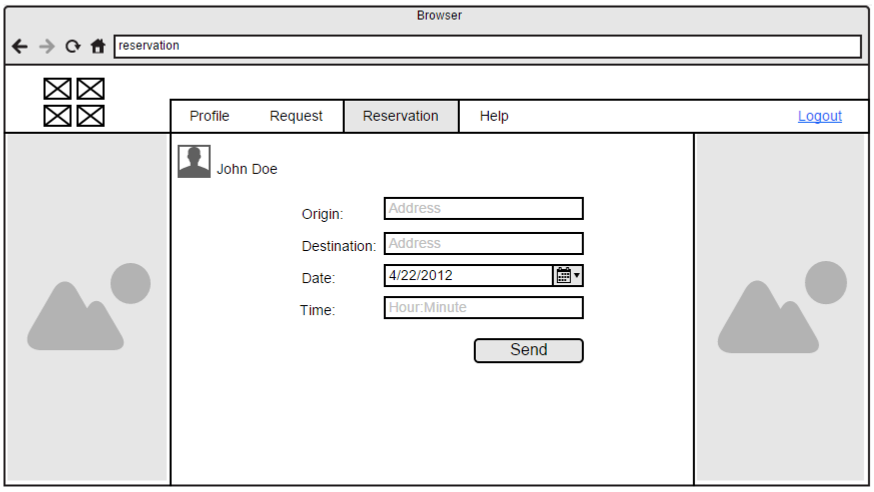


User interface 4: Request form

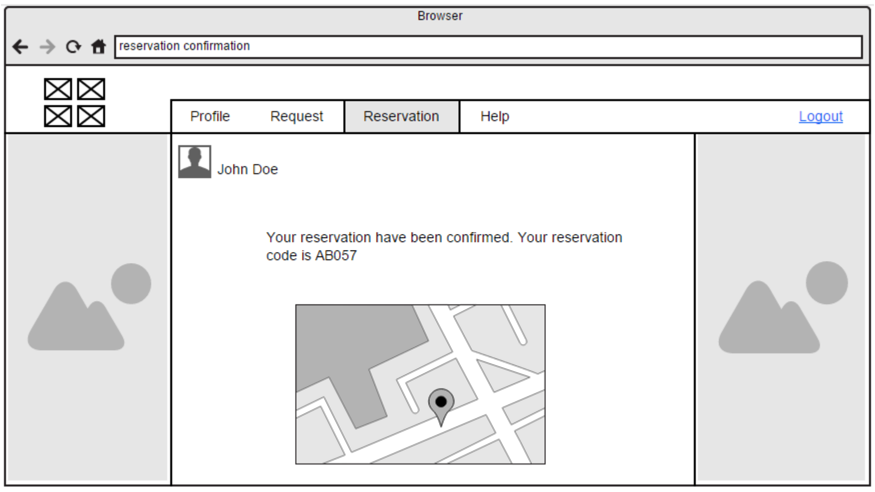


Your request have been confirmed. Your taxi code is 058CDF and you will need to wait for 5 minutes

User interface 5: Request confirmation

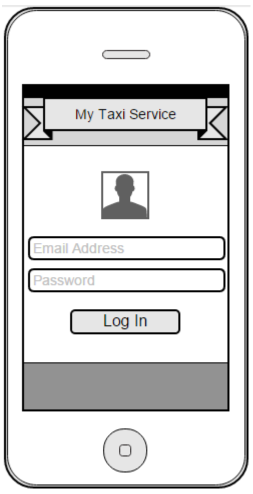


User interface 6: Reservation form

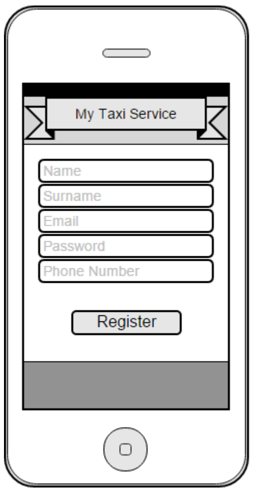


Your reservation have been confirmed. Your reservation code is AB057

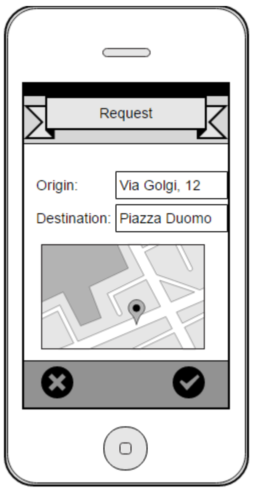
User interface 7: Reservation confirmation



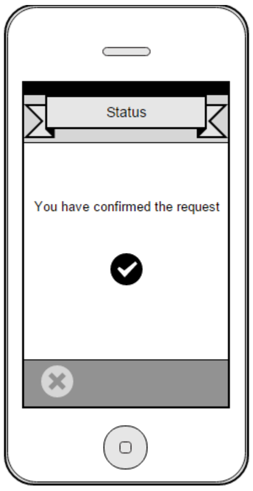
User interface 8: Log in form



User interface 9: Register form



User interface 10: Asking for request confirmation



User interface 11: Confirmation message

2.1.3 Hardware Interfaces

This software product does not provide any hardware interfaces.

2.1.4 Software Interfaces

Database Management System

Name: MySQL
Mnemonic: MySQL
Specification number: Community Server
Version number: 5.5.28
Source: http://www.mysql.com/downloads/mysql/

Application Server

Name: GlassFish Server
Mnemonic: GlassFish Server
Version number: 4.1
Source: https://glassfish.java.net/download.html

The mobile application will be able to run on multiple platforms, including iOS, Android, Windows OS.

Operating System

The software product will run in any operating that supports JVM, DBMS and application server described above.

2.1.5 Comunnication Interfaces

Protocol	Port	Service
TCP	80	TCP
TCP	3306	MySQL (only if is in a different physical server)

Table 3: Communication interfaces

2.1.6 Memory

	Memory requirements
RAM	2GB+
Storage	40GB+

Table 4: Memory requirements

Secondary memory is recommended to be physically in a different server from the software it is installed on, for reason that over the time it can exponentially grow, and can affect the system performance.

2.1.7 Operations

A user can interact with the system as functional user (passengers or taxi drivers), or as a maintenance user (administrators).

Functional operations for functional users are described in more details in the product functions section.

Operations for maintenance user, or administrators, will include: managing the profiles of taxi drivers, i.e. administrator will register taxi drivers. This will allow the system to recognize driver's profiles and manage other functions correctly.

Moreover, other capabilities can be implemented through the programmatic interfaces the system offers, for developing additional services, on top of the basic ones, like for example an option for sharing taxi, etc.

2.1.8 Site Adaptation Requirements

Conditions that need to be satisfied in order this software product to be installed and run successfully are:

- JVM
- AS
- DBMS
- Primary memory required space.
- Secondary memory required space.

Users are required to have only a compatible browser installed in his personal computer or mobile phone.

2.2 Product Functions

This subsection describes a summary of major functions, functional and non-functional requirements of the software product.

2.2.1 Functional requirements

Functional requirements are grouped in several principal topics:

- Manage users.
- Manage the requests from the users.
- Manage the reservations from the users.
- Manage confirmations of the requests and reservations.

These requirements are defined and explained in more details in the next subsections.

Manage users

Functional requirements:
[FR1] Create Passenger
[FR2] Create Taxi driver
[FR3] Consult user profile
[FR4] Update user profile
[FR5] Register to the system
[FR6] Log in
[FR7] Log out

Non-functional requirements:
[NFR1] The software product can be used only by registered users
[NFR2] User password must be saved securely
[NFR3] The system must manage high number of users

Manage requests from the users

Functional requirements:
[FR8] Create request
[FR9] Consult request

Non-functional requirements:
[NFR4] The system must manage high number of requests
[NFR5] Requests already made cannot be modified

Manage reservations from the users

Functional requirements:
[FR10] Create reservation
[FR11] Consult reservation

Non-functional requirements:
[NFR6] The system must manage high number of reservations
[NFR7] Reservations already made cannot be modified
[NFR8] If a reservation is not made at least two hours before the ride, the system will not accept the reservation

Manage confirmations of the requests and reservations

Functional requirements:
[FR12] Taxi driver can accept request
[FR13] Taxi driver can decline request
[FR14] Taxi driver can accept reservation
[FR15] Taxi driver can decline reservation
[FR16] System sends code and waiting time of available taxi to the passenger

Non-functional requirements:
[NFR9] The system must manage sending correct codes and waiting time to the passengers
[NFR10] The system must manage high number of confirmations of requests and reservations

2.3 User Characteristics

Main user characteristic is a knowledge in using a browser and mobile application.

2.4 Constraints

The following constraints apply to the software product:

2.4.1 Regulatory policies

The software product does not have any regulatory policies.

2.4.2 Hardware Limitations

The software product does not have any hardware limitations.

2.4.3 Interfaces to Other Applications

The software product does not have interface with other applications.

2.4.4 Parallel Operation

This software product must control different levels of concurrency, including: concurrent users, creation and management of taxi requests and reservations, management of confirmations and notifications, etc.

Therefore it is necessary management of parallel operations at application level, database level and file system level. These operations are important for the general functionality of the product and they will be resolved in the design planning phase of the project.

2.4.5 Audit Function

The software does not perform any audit.

2.4.6 Control Functions

The software product does not control any device or any other system.

2.4.7 Higher-order Language Requirements

The developer must have knowledge in HTML, Java language and JEE technologies.

2.4.8 Signal Handshake Protocol

The software product does not manage any handshake protocol.

2.4.9 Reliability requirements

The software product does not require any specific requirements to perform and maintain its functions under normal operation.

2.4.10 Criticality of the application

The software product requires proper support for concurrent users.

2.4.11 Safety and security considerations

The software product does not require any safety and security considerations.

2.5 Assumptions and Dependencies

The requirements in this document are grounded on the following assumptions:

1. JVM is already installed on the OS.
2. Users have a decent and acceptable Internet connection.
3. The software product provides one administrator user by default.
4. The software product can support up to 3 administrators.

2.6 Apportioning of requirements

Future releases of this software product may provide support for adding additional services, including:

taxi sharing;

modifying and canceling requests and reservations;

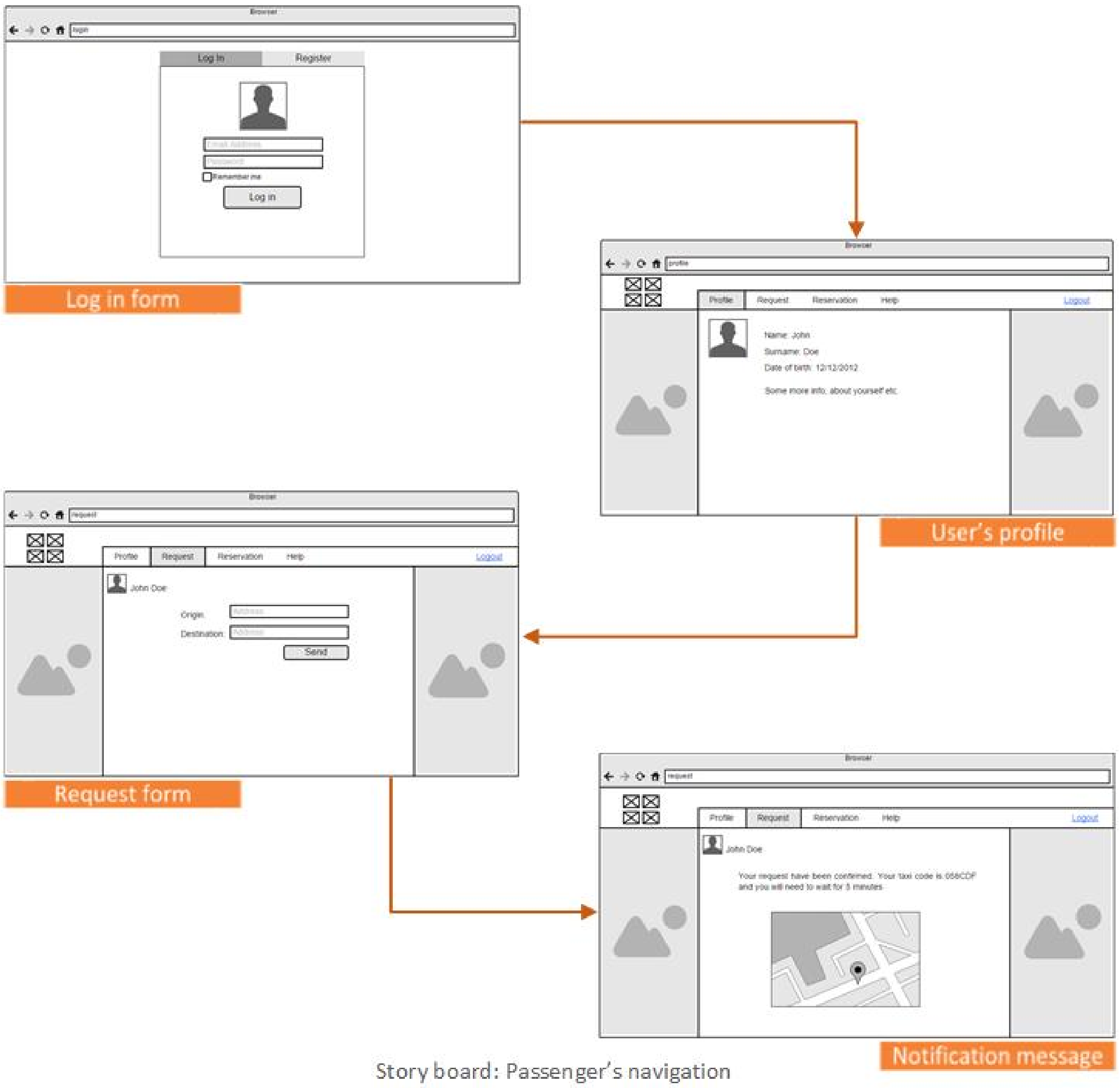
online payment, etc.

3 Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

This story board explains how previously mentioned user interfaces are connected with each other.



John Doe

Your request have been confirmed. Your taxi code is 058CDF and you will need to wait for 3 minutes.

Story board: Passenger's navigation in creating request

3.1.2 Hardware Interfaces

The system does not have hardware interfaces.

3.1.3 Software Interfaces

The software product does not provide any software interfaces.

3.1.4 Communication Interfaces

The system does not have communication interfaces.

3.2 Functional Requirements

3.2.1 Scenarios

User registers to the system	
Code	SC-001
Description	Describe how a user registers to the system
Goal	**[G1]** Users can register to the system and have their own profiles
Assumption	User is not registered on the system
A friend of Clara recommended to her this application. Clara decides to use myTaxiService. She connects to the internet and opens up a browser installed on her computer. She goes to the internet address of the application and when the first page loads, Lara notices that there is a register option. After clicking the "Register" button, the register form opens up where she writes her name, surname, email, password, phone number and other information. When she finishes filling out the form, she clicks on the button "Confirm", after which she receives a message stating that the registration has been successful. Clara is now registered on the system.	

Scenario 1: User registers to the system

Passenger sends request for available taxi	
Code	SC-002
Description	Describe how a user sends request for available taxi
Goal	**[G3]** Passengers can make requests for available taxi **[G6]** The system informs passengers for confirmation of available taxi by sending code of the correspondingtaxi and waiting time. **[G7]** The system should consistently track distribution of taxis in the specific zones, based on the GPS information it receives from each taxi.
Assumption	The user is a passenger, already registered to myTaxiService.
Viktor will have a coffee with his friend in half an hour. He needs ride to the place they should meet, because he is running errands in other area of the city. He unlocks his phone, connects to internet and log in to myTaxiService. He is redirected to his profile, where he can see "Request" button. Viktor clicks on that button and he is redirected to other page, where he needs to specify his current location. Viktor inputs his current address and clicks on "Confirm request" button. After that Viktor gets notification on his profile. The notification specifies the code of the taxi and the waiting time.	

Scenario 2: Passenger sends request for available taxi

Taxi driver accept request	
Code	SC-003
Description	Describe how taxi drivers accept requests
Goal	**[G3]** Passengers can make requests for available taxi. **[G5]** Taxi drivers can accept or decline request. **[G6]** The system informs passengers for confirmation of available taxi by sending code of the correspondingtaxi and waiting time. **[G7]** The system should consistently track distribution of taxis in the specific zones, based on the GPS information it receives from each taxi.
Assumptions	The user is taxi driver, already registered on the system.
Luka is taxi driver. He has a passenger in his car. They reach passenger's destination. Luka now has no passengers. He unlocks his phone, connects to internet and log in to myTaxiService. He is redirected to his profile, where he can navigate to the tab "Requests". Luka can see that a request for an address nearby has just been made. He clicks on the button "Accept this request". Luka starts driving his car to the address specified in the request.	

Scenraio 3: Taxi driver accept request.

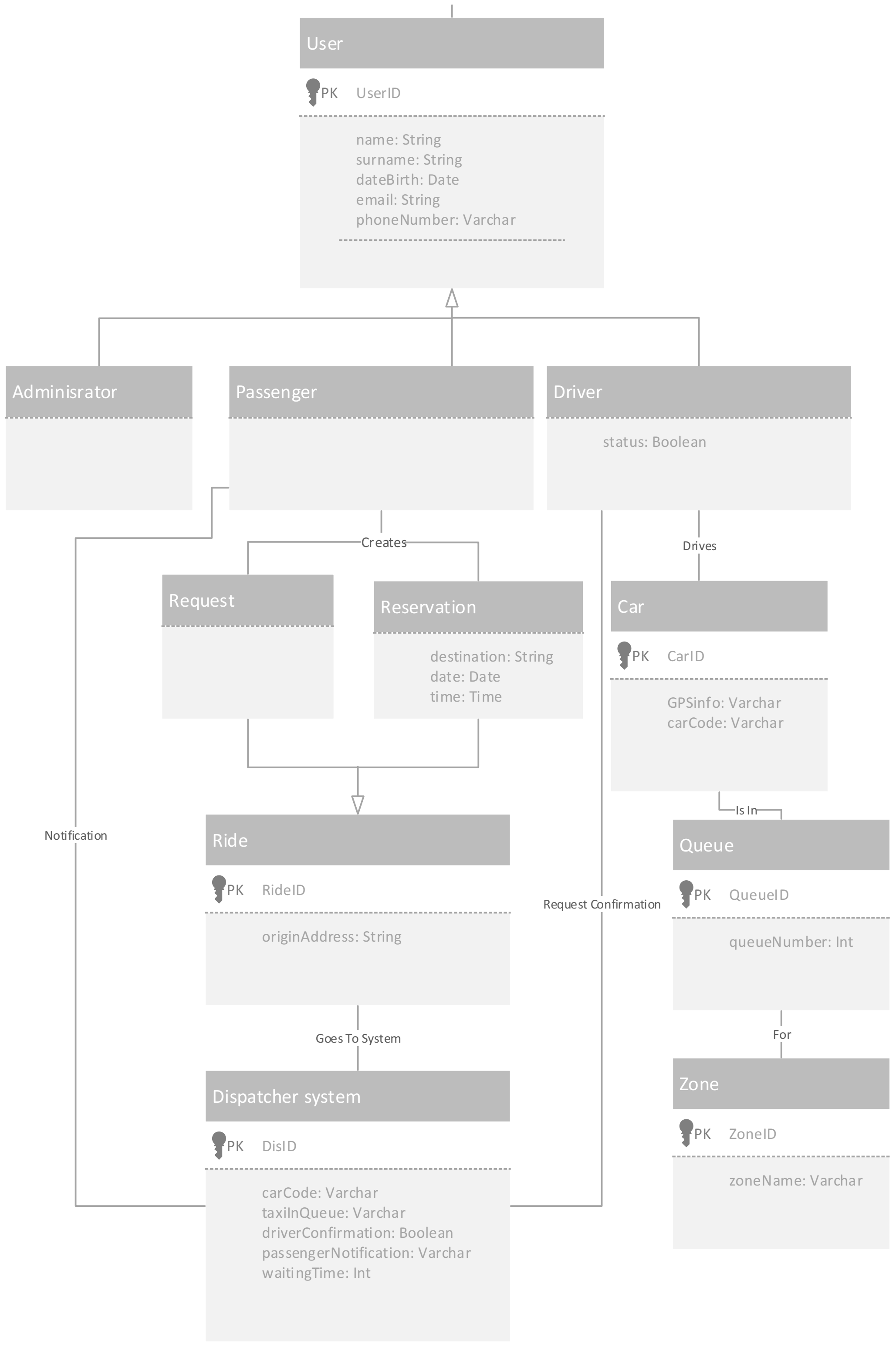
Taxi driver decline request	
Code	SC-004
Description	Describe how taxi drivers decline requests
Goal	**[G3]** Passengers can make requests for available taxi. **[G5]** Taxi drivers can accept or decline request. **[G7]** The system should consistently track distribution of taxis in the specific zones, based on the GPS information it receives from each taxi.
Assumptions	The user is taxi driver, already registered on the system.
Luka is taxi driver. He has a passenger in his car. They reach passenger's destination. Luka now has no passengers. He unlocks his phone, connects to internet and log in to myTaxiService. He is redirected to his profile, where he can navigate to the tab "Requests". Luka can see that there is a request, but the address is very far from his current location. Luka decides it will not be efficient to take this request, so he clicks on the button "Decline this request". Luka continues to search for other requests.	

Scenario 4: Taxi driver decline request.

Passenger makes reservation	
Code	SC-005
Description	Describe how passenger makes reservations
Goal	**[G4]** Passengers can make reservations.
Assumptions	The user is passenger, already registered on the system.
Clara is invited to her friend's graduation party on Friday. Clara wants to get there on time, so she decides to make a reservation beforehand. She opens a browser on her laptop, type the address of myTaxiService web application. She inserts her username and password. After successful log in, she is redirected to her profile. She navigates to the tab "Make a reservation". In the new window, Clara is asked to input date, time, address of the origin of the ride and address of the destination. She fills the necessary information and clicks on "Confirm reservation". A message that successful reservation has been made is shown.	

Scenario 5: Passenger makes reservation

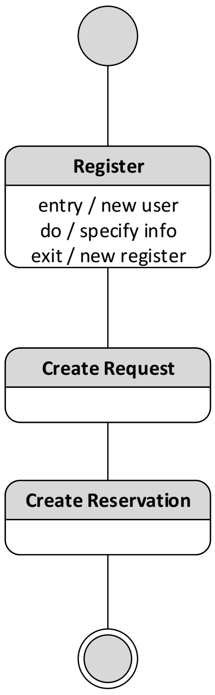
3.2.2 Analysis Model



Class diagram 1: Analysis model

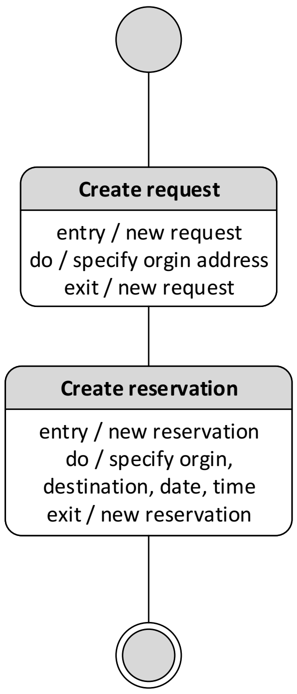
3.2.3 State Chart Model

State diagram of a user



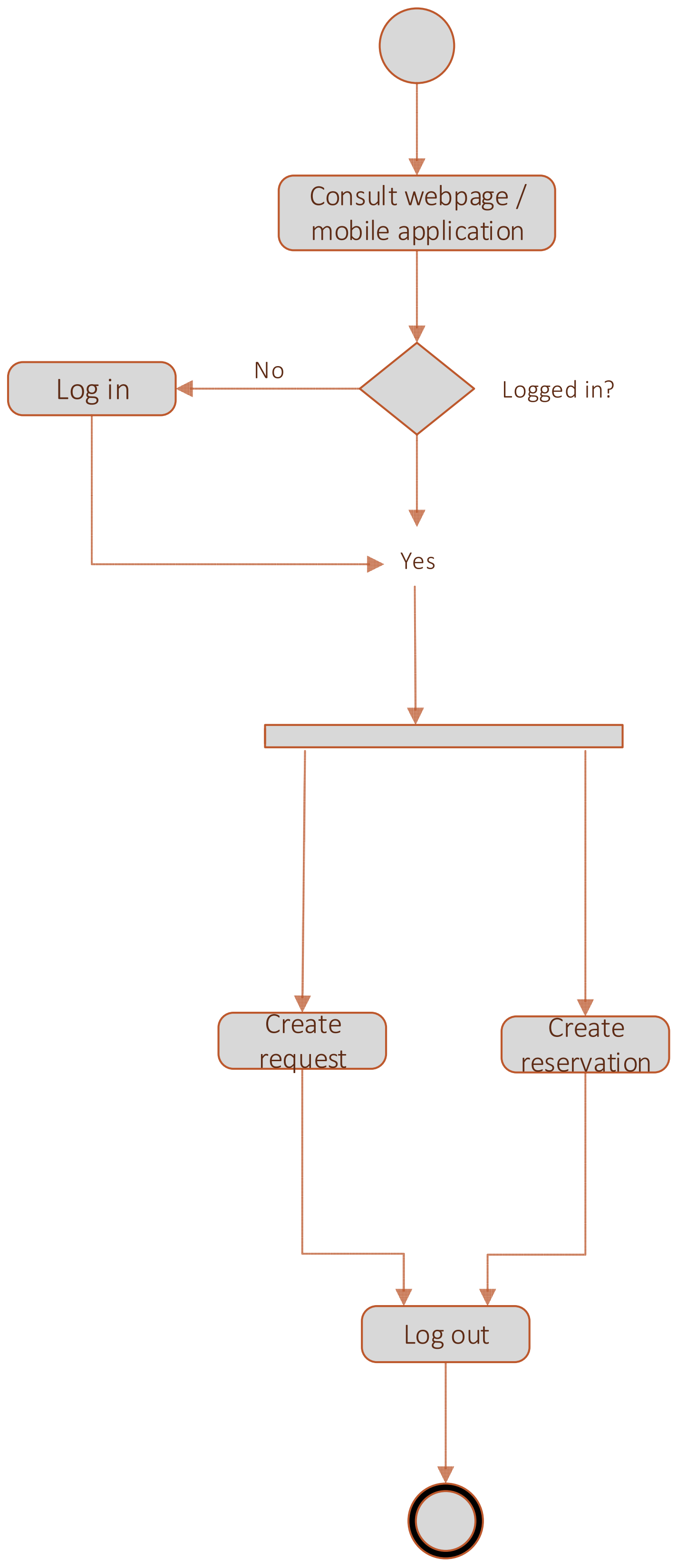
State diagram 1: User

State diagram of request and reservation

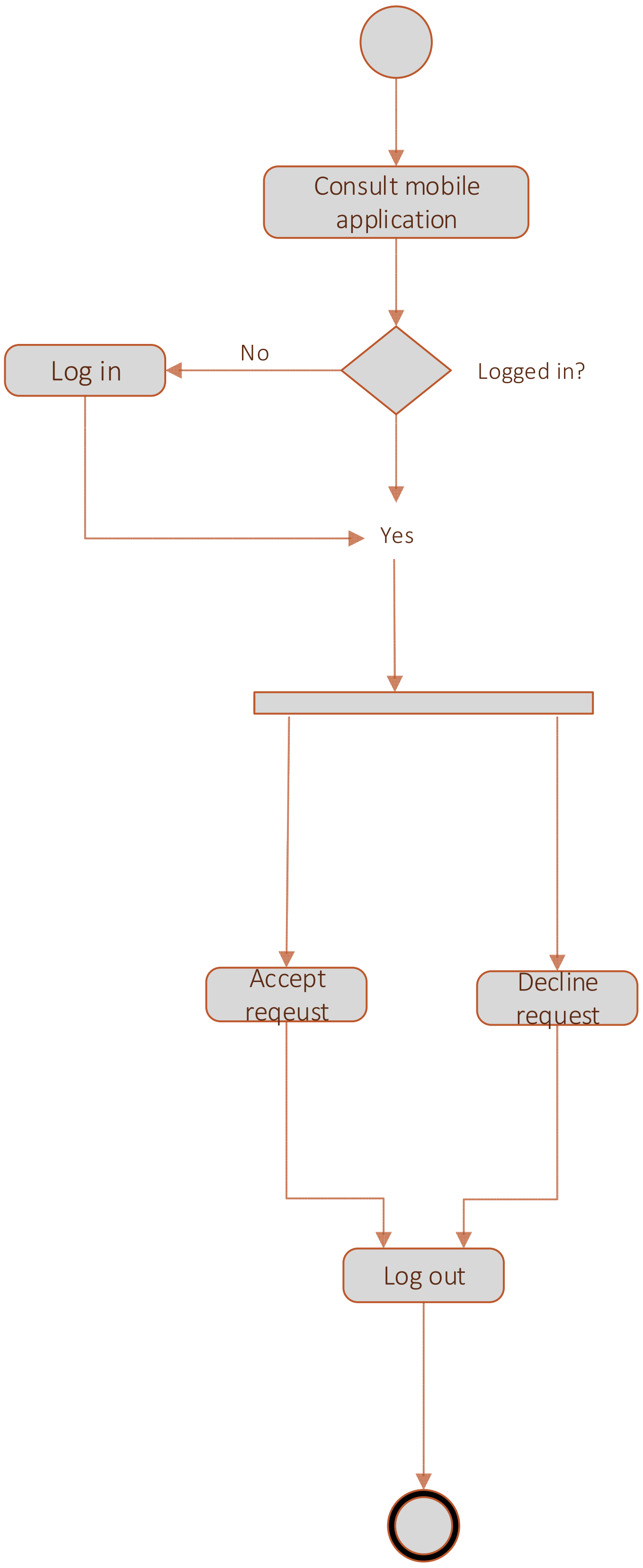


State diagram 2: Request/Reservation

3.2.4 Activity Model



Activity diagram 1: Actions of passenger



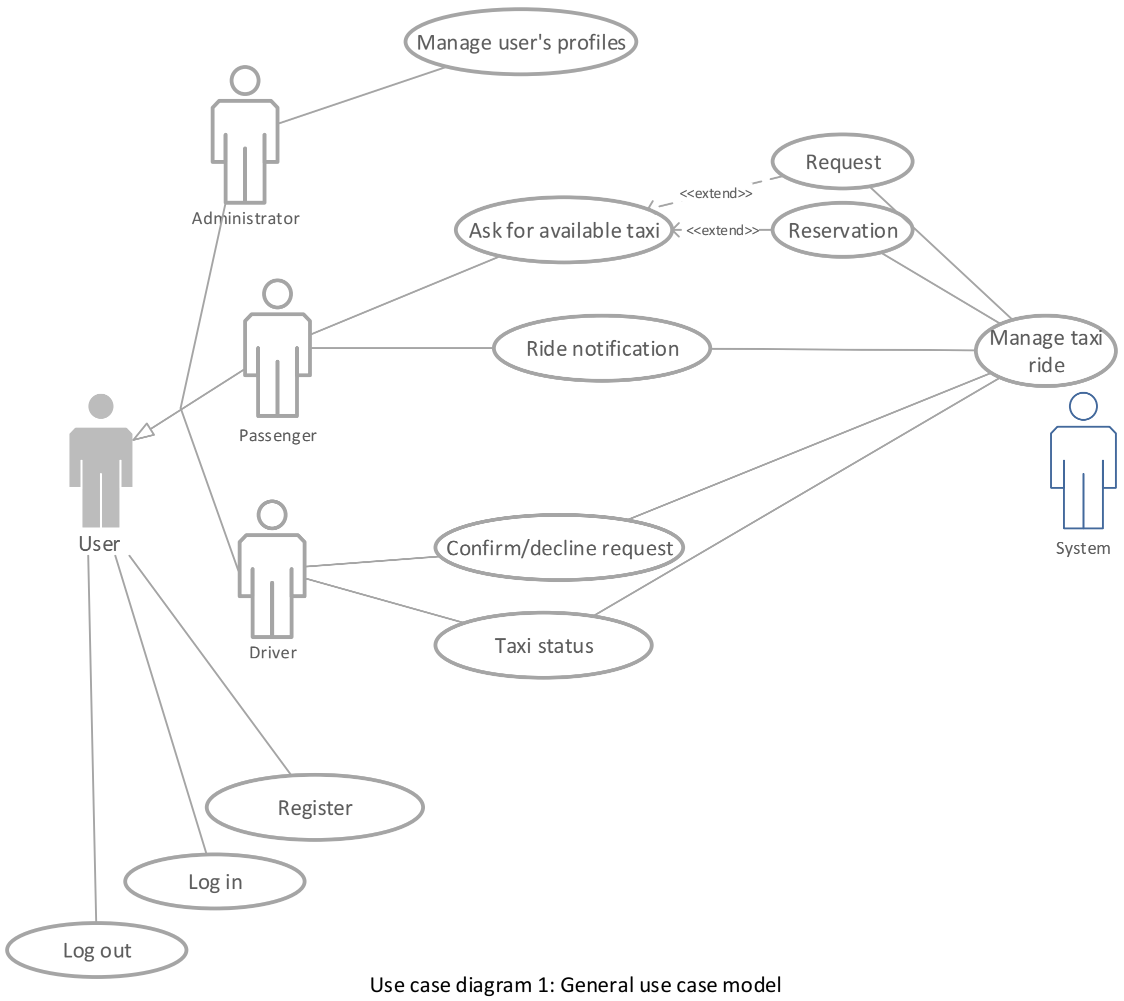
Activity diagram 1: Actions of taxi driver

3.2.5 Use Cases

The software product constains the following main actors:

- Administrators
- Passengers
- Taxi drivers
- System

The following diagram shows the general use cases for each of these actors. In the next subsections they are explained in details.



Use case diagram 1: General use case model

The following use cases that are shared between passengers and drivers with the system are identified:

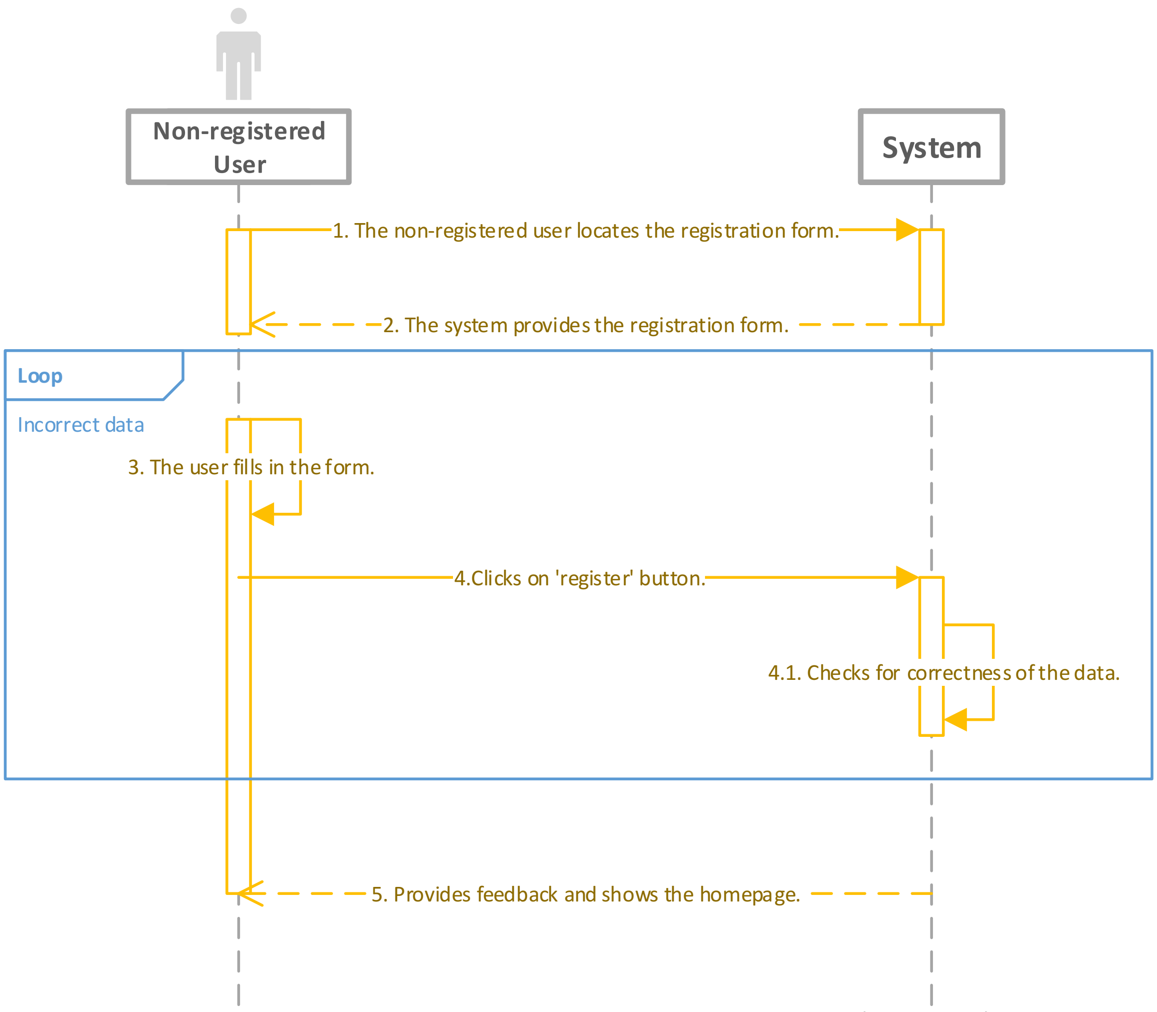
1. User registers on the system
2. User logs in to the system
3. User logs out of the system
4. Passenger creates request
5. Passenger creates reservation
6. Driver accepts request

7. Driver declines request

User registers on the system
Code: USC-001
Description: Registering in the system
Goal: [G1]
Actors: Non-registered user
Entry condition: Non-registered user navigates to the homepage of myTaxiService
Exit condition: Non-registered user is registered.

Flow of events:
1. The non-registered user locates the registration form.
2. The system provides the registration form.
3. User fills in the form.
4. User clicks on "register" button.
5. The system provides feedback and shows homepage.

Exceptions: the user adds incorrect data.


Non-registered User
System
1. The non-registered user locates the registration form.
2. The system provides the registration form.
Loop
Incorrect data
3. The user fills in the form.
4.Clicks on 'register' button.
4.1. Checks for correctness of the data.
5. Provides feedback and shows the homepage.

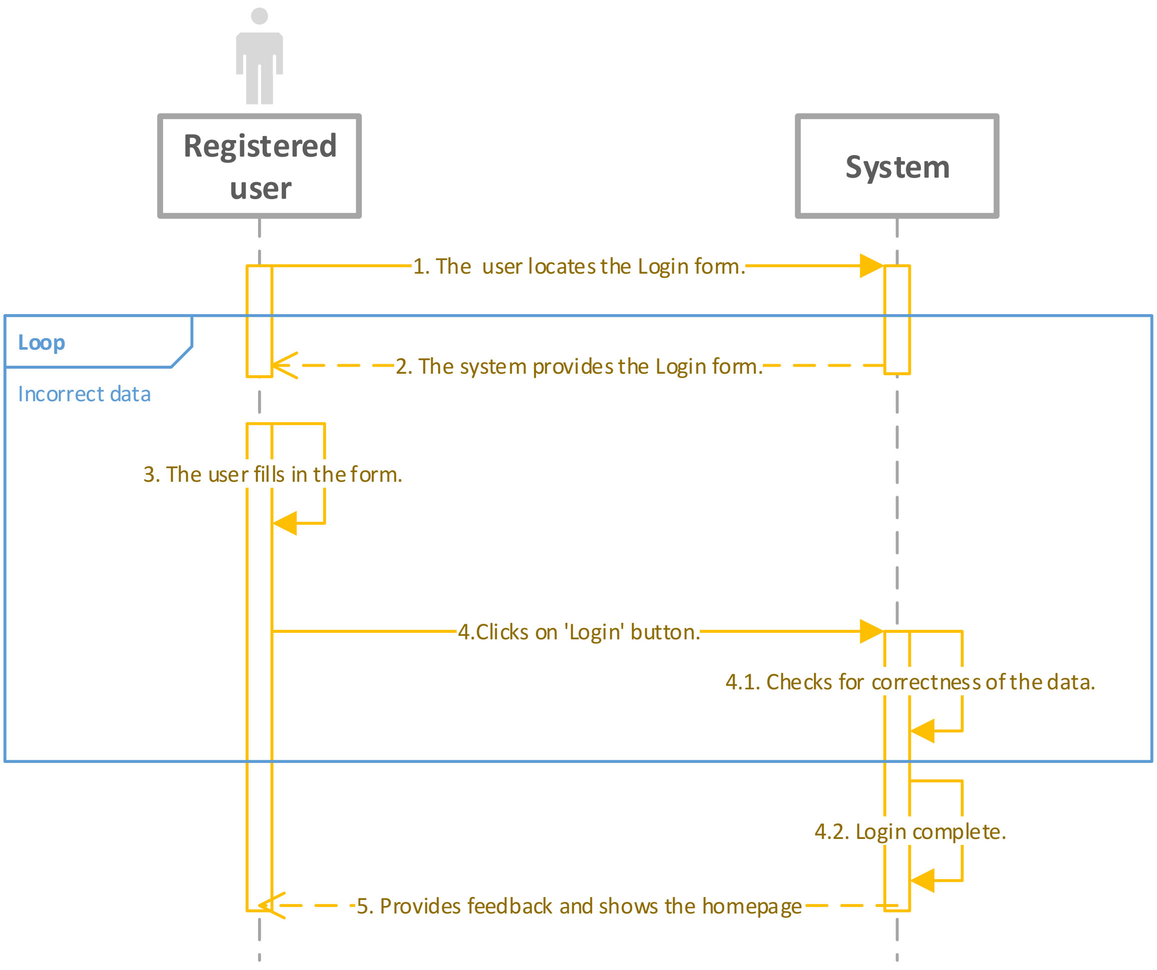
Sequence diagram 1: User registers to the system (USC-001)

User logs in the system
Code: USC-002

Description: Registered user login.
Goal: [G1], [G2]
Actor: Registered user
Entry condition: Registered user navigates to the homepage of myTaxiService.
Exit condition: User is logged out.

Flow of events:
1. User locates the Login form.
2. System provides the Login form.
3. User fills in the form.
4. User clicks on “Login” button.
5. System provides feedback and shows homepage.

Exceptions: the user adds incorrect data.

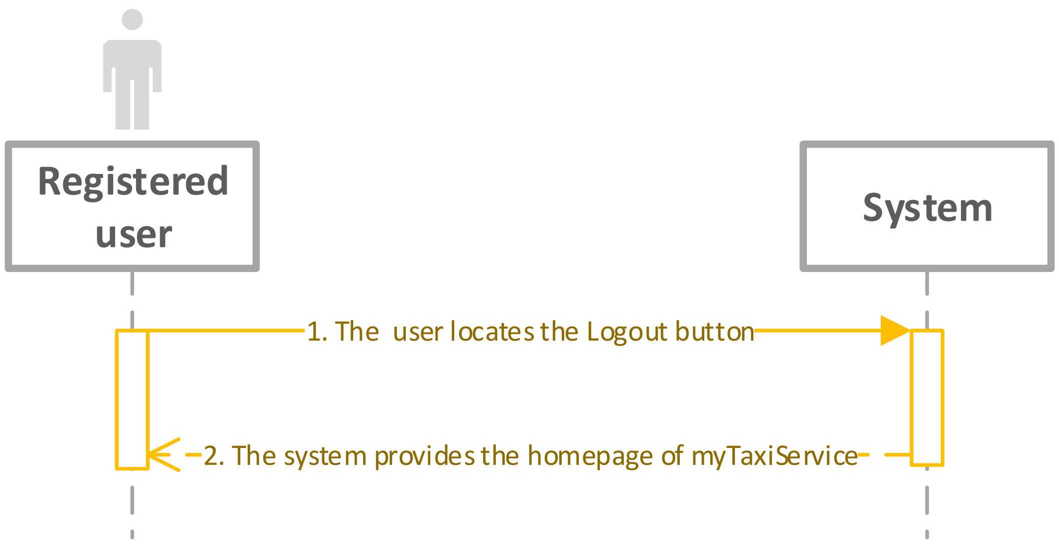

Registered user
System
1. The user locates the Login form.
Loop
Incorrect data
2. The system provides the Login form.
3. The user fills in the form.
4.Clicks on 'Login' button.
4.1. Checks for correctness of the data.
4.2. Login complete.
5. Provides feedback and shows the homepage

Sequence diagram 2: User logs in to the system (USC-002)

User logs out of the system
Code: USC-003
Description: User logs out.
Goal: [G1], [G2]
Entry condition: The user is logged in.
Exit condition: The user is logged out.

Flow of events:

1. User locates the “Logout” button.
2. System provides the homepage of myTaxiService.


Registered user
System
1. The user locates the Logout button
2. The system provides the homepage of myTaxiService

Sequence diagram 3: User logs out of the system (USC-003)

Passenger creates request

Code: USC-004

Description: Passenger creates request.

Goal: [G3], [G6]

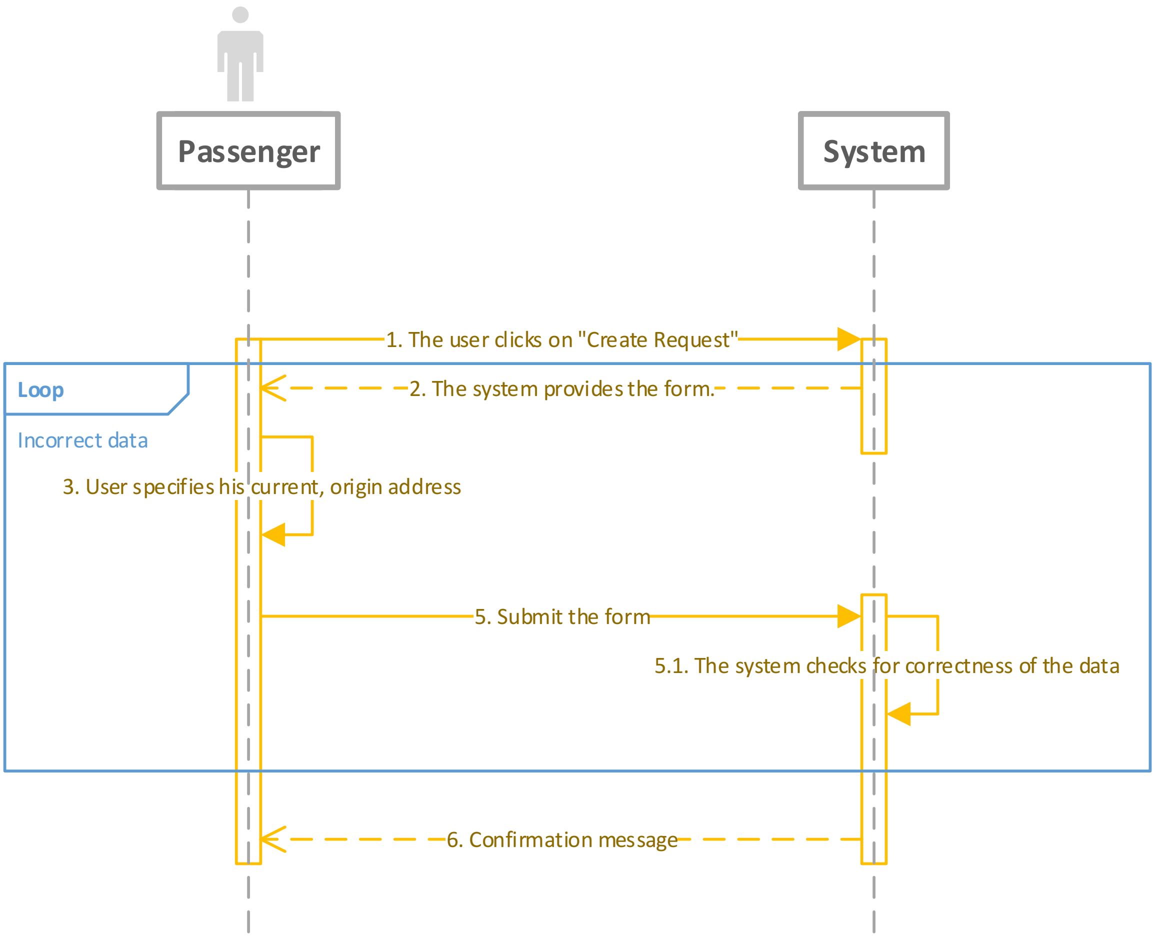
Entry condition: The user is logged in.

Exit condition: User gets notification for confirming the request.

Flow of events:

1. User clicks on “Create request”
2. System provides the form
3. User specifies his current address of origin.
4. User submits the form.
5. User gets confirmation message.

Exeptions: User adds incorrect data as address.


Passenger
System
1. The user clicks on "Create Request"
Loop
2. The system provides the form.
Incorrect data
3. User specifies his current, origin address
5. Submit the form
5.1. The system checks for correctness of the data
6. Confirmation message

Sequence diagram 4: Passenger creates request (USC-004)

Passenger creates reservation

Code: USC-005

Description: Passenger creates reservation.

Goal: [G4]

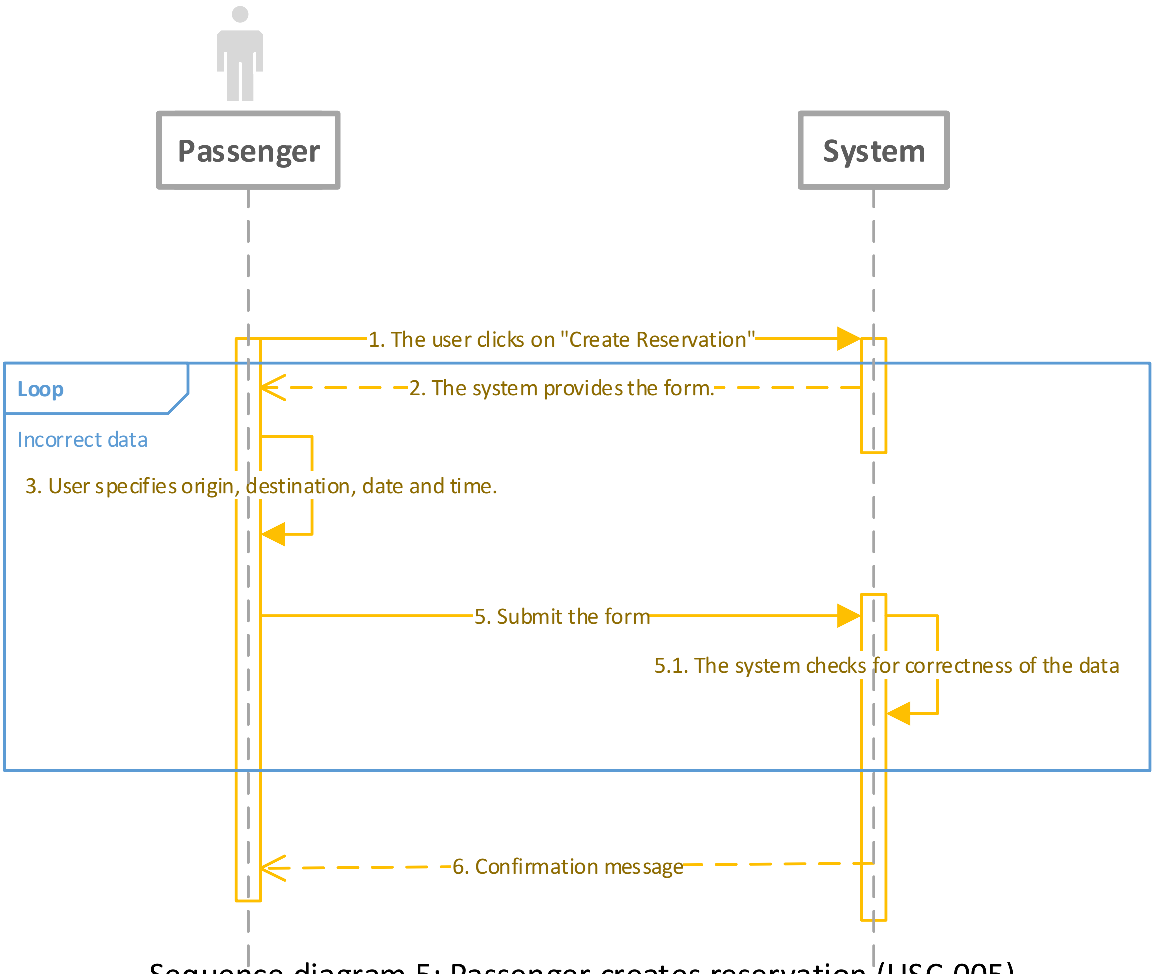
Entry condition: The user is logged in.

Exit condition: User gets notification for confirming the reservation.

Flow of events:

1. User clicks on “Create reservation”
2. System provides the form
3. User specifies origin, destination, date and time of the ride.
4. User submits the form.
5. User gets confirmation message.

Exeptions: User adds incorrect data as address, date or time.


Passenger
System
Loop
Incorrect data
1. The user clicks on "Create Reservation"
2. The system provides the form.
3. User specifies origin, destination, date and time.
5. Submit the form
5.1. The system checks for correctness of the data
6. Confirmation message

Sequence diagram 5: Passenger creates reservation (USC-005)

Driver accepts request

Code: USC-006

Description: Driver accepts request.

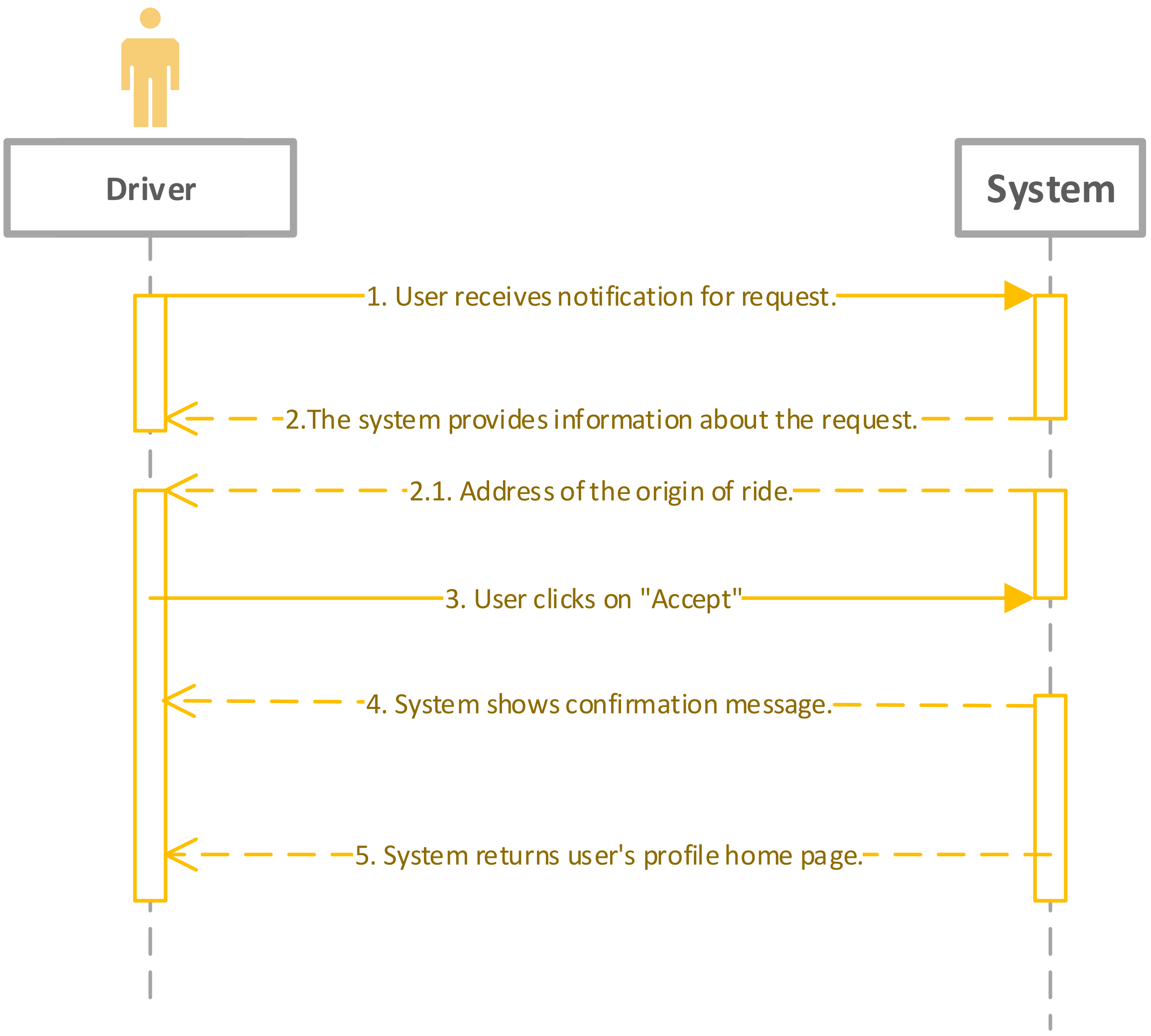
Goal: [G5], [G7]

Entry condition: The user is logged in.

Exit condition: User accepts the request.

Flow of events:

1. User receives notification for request.
2. System provides information about the request: address of the origin of ride.
3. User clicks on "Accept"
4. System shows confirmation message.
5. System returns user's profile homepage.



Driver
System
1. User receives notification for request.
2.The system provides information about the request.
2.1. Address of the origin of ride.
3. User clicks on "Accept"
4. System shows confirmation message.
5. System returns user's profile home page.

Sequence diagram 6: Driver accepts request (USC-006)

Driver accepts request

Code: USC-007

Description: Driver declines request.

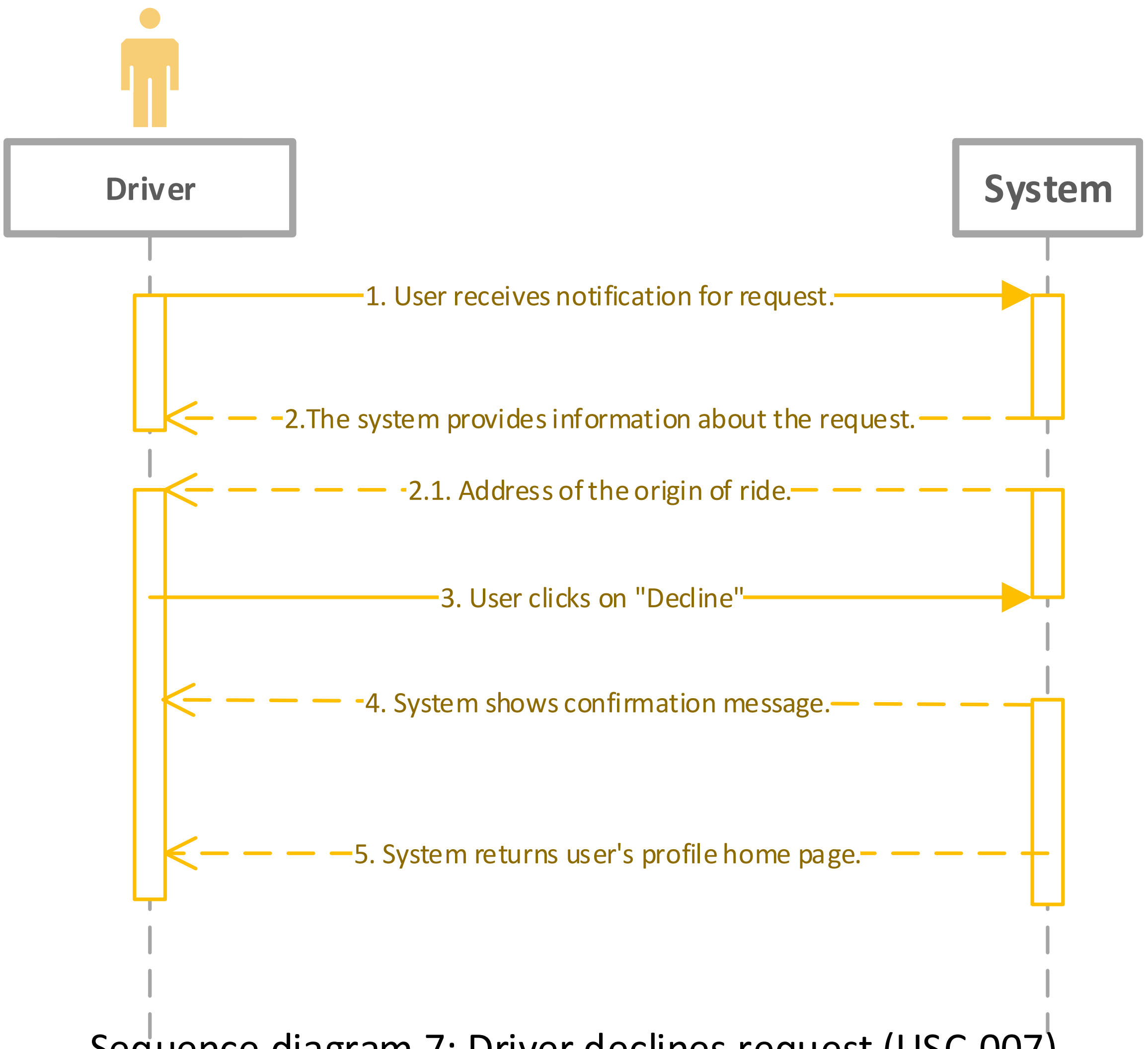
Goal: [G5], [G7]

Entry condition: The user is logged in.

Exit condition: User declines the request.

Flow of events:

1. User receives notification for request.
2. System provides information about the request: address of the origin of ride.
3. User clicks on “Decline”
4. System shows confirmation message.
5. System returns user’s profile homepage.


Driver
System
1. User receives notification for request.
2.The system provides information about the request.
2.1. Address of the origin of ride.
3. User clicks on "Decline"
4. System shows confirmation message.
5. System returns user's profile home page.

Sequence diagram 7: Driver declines request (USC-007)

3.3 Performance Requirements

The software product requires that each web page loads in less than 5 seconds. Also, as mentioned before, it is necessary to support a high level of concurrent users and taxi requests.

3.4 Logical Database Requirements

The database must save the entire entities identified in the analysis diagram, because they are necessary for managing the software product information.

3.5 Design Constraints

The system must be designed and implemented in JEE technology and its related technologies.

3.6 Standard Compliance

The software product must be developed following recommended standards in order to be easily readable and updatable.

3.7 Software System Attributes

3.7.1 Reliability

The system shall assure the integrity of the users.

3.7.2 Availability

The system should be available 24 hours per day, 7 days per week, and 365 days per year.

3.7.3 Security

The software product must encrypt users' password. This can be achieved using cryptographic techniques.

3.7.4 Maintainability

The database of the software product should be backed up periodically, so that in case of failure existing data can be reconstructed easily.

The system needs maintainability in clearing the users that are inactive for longer periods, server files system backup and periodical cleaning of old requests, in order not to affect the performance of the system.

3.7.5 Portability

The software product can be installed on any operating system which supports Java Virtual Machine and its dependent components.

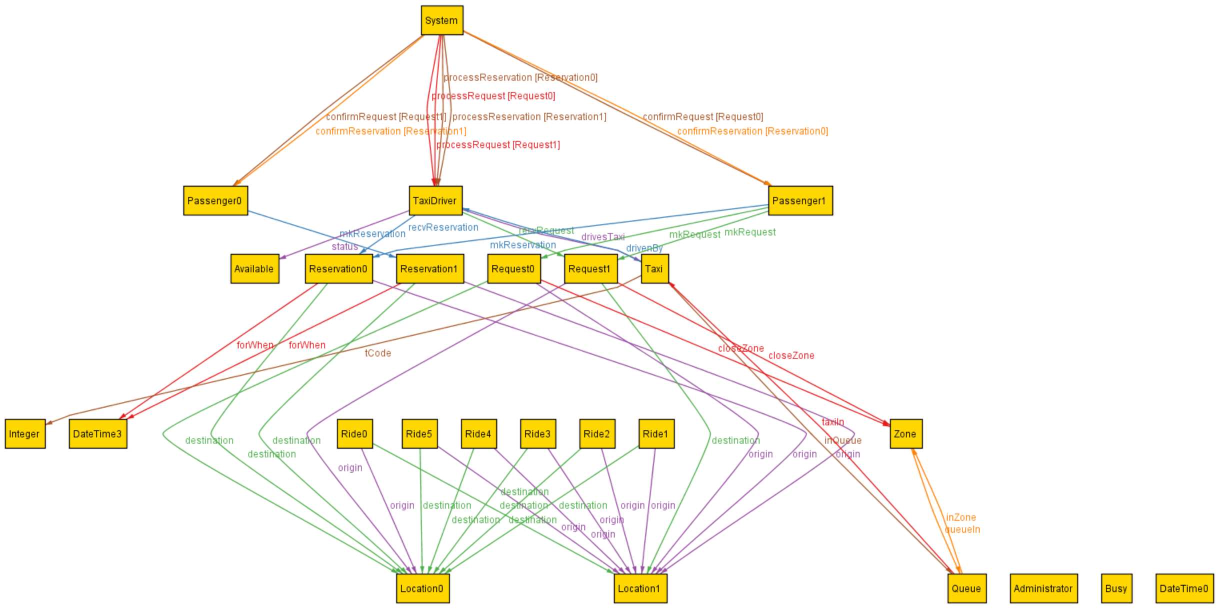
3.8 Other Requirements

The software product must provide understandable messages in text form in the event of an error and instruct the user on what to do.

4 Appendices

4.1 Alloy model

Here is a screenshot of the world generated by the Alloy model.


System
processReservation [Reservation0]
processRequest [Request0]
confirmRequest [Request1]
processReservation [Reservation1]
confirmReservation [Reservation1]
processRequest [Request1]
confirmRequest [Request0]
confirmReservation [Reservation0]
Passenger0
TaxiDriver
Passenger1
mkReservation
recvReservation
status
recvRequest
drivesTaxi
mkReservation
drivenBy
mkRequest
mkRequest
Available
Reservation0
Reservation1
Request0
Request1
Taxi
forWhen
forWhen
tCode
closeZone
closeZone
Integer
DateTime3
destination
destination
destination
Ride0
Ride5
Ride4
Ride3
Ride2
Ride1
destination
origin
origin
origin
taxiIn
inQueue
Zone
origin
origin
destination
destination
destination
destination
origin
origin
destination
destination
origin
origin
origin
inZone
queueIn
Location0
Location1
Queue
Administrator
Busy
DateTime0