Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “myTaxiService”
Requirements **A**nalysis and **S**pecifications **D**ocument
version 1.1

Adrian Mihai Berbieru #854698
Attilio D’Onofrio #789614

21/01/2016

1 Introduction

1.1 Purpose

This document represents the Requirement Analysis and Specification Document (RASD). The main goal of this document is to completely describe the system in terms of functional and nonfunctional requirements, analyse the real needs of the customer to model the system, show the constraints and the limit of the software and simulate the typical use cases that will occur after the development. This document is intended to all developer and programmer who have to implement have to implement the requirements, to system analyst who wants to integrate other system with this one, and could be used as a contractual basis between the customer and the developer.

1.2 Actual System

The government of the city wants to provide a new service to citizens to manage taxi calls and bookings. We suppose until now taxi reservations were made by using a call centre as an intermediary between customer and taxi driver. The call centre will not be dismissed as the new system aims to offer a new feature to customers, without changing the traditional way of booking. On the other hand, call centre agents will have the same new system in order to assign a taxi to every client that uses the "old" booking method.

1.3 Scope

The aim of the project is to simplify the taxi booking process by using a web application or a smartphone app. The system will also increase the efficiency of taxi management (idle time of taxi drivers will be minimized).

The only action visitors can perform is the registration process.

Registered users are able to make reservations and they can choose from two different options: they may want to be immediately picked up or they may decide to make a reservation by fixing origin and destination of the ride.

In the first case, the user will send his/her position manually or by using the GPS automatic detection; then the system will select an available taxi from the queue of that specific city zone. It is up to the driver of the selected taxi whether to accept or not the call. If there is no taxi available in that city zone, the system informs taxis are unavailable and then informs the user when a taxi is dispatched. During this waiting time, user is able to delete the request.

In the second case they can make a long term reservation. It has to be made at least two hours and up to seven days before the ride will take place. Customers are able to delete the reservation made anytime, but at least sixty minutes before the ride would take place.

After the booking, in both cases, the customer will receive a confirm of the reservation containing the code of the taxi assigned to the ride.

1.4 Actors

The kind of people that will interact with the system:

- Visitors: no features are provided for unidentified users, except for the login and the registration;
- Registered user: after the successful login, users can request a taxi, make a reservation or delete a reservation made before;
- Call center operator: user who is a legacy from the traditional system, he uses the same web application to make reservation after receiving a call from the customer.
- Taxi driver: each driver is associated to a city zone, to his availability and the position in the queue of that city zone; he can change its availability and decide whether or not to accept the requests sent by the system.
- System administrator: a technician in charge of adding taxi drivers to DBMS via SQL language.

1.5 Goals

List of goals of myTaxiService system:

- **[G1]** To allow the visitor to register;
- **[G2]** To allow the unidentified user to log in;
- **[G3]** To allow the registered user to make an instant booking;
    - **[G3.1]** To allow the registered user to send his position by writing the address;
    - **[G3.2]** To allow the registered user to send his position by using the GPS detection;
    - **[G3.3]** To allow the registered user to specify the number of passengers;
    - **[G3.4]** To allow the registered user to delete the request, in case taxis are not available;
    - **[G3.5]** To allow the registered user to receive the code of taxi that accepted the request and the estimated time to the arrival of the taxi;
- **[G4]** To allow the registered user to make a reservation in a specified time;
    - **[G4.1]** To allow the registered user to add the origin and destination of the ride;
    - **[G4.2]** To allow the registered user to specify the number of passengers;
- **[G5]** To allow the registered user to delete the reservations he made;
- **[G6]** To allow the taxi driver to change its availability status;
- **[G7]** To allow the taxi driver to accept or decline a call;
- **[G8]** To allow the taxi driver to get GPS directions until he reaches client's position;
- **[G9]** To allow the taxi driver to login.

1.6 Identifying stakeholders

The financial stakeholder is the government of the city, who funded the project. Government's main interest is to improve the quality of the service provided by taxis.

The other stakeholders are customers, call center operator, taxi drivers (all described in section 1.4) who will interact with the system.
Eventually, all the analysts and programmers involved in the development may be considered stakeholders too.

1.7 Definitions, Acronyms, Abbreviations

1.7.1 Definitions

- Visitor: also called unregistered user, is a person who is not registered to the system.
- Customer: also called user or client, is the person who books the ride via the web or smartphone app.
- City zone: the part of the city the taxi is assigned to. Each zone has an area of about 2km^2.
- Availability: referred to a taxi, it says if the driver is able to take calls or not.
- Queue: the list of taxis associated to a city zone.
- Code: the identifier associated to a taxi.
- Booking: also called reservation, the act of asking for a taxi, done by a customer through the system. It may be instant or long term.
- Instant booking: also called instant reservation, it is done when a customer requests a taxi, sending his current position.
- Long term booking: also called long term reservation, it is done when a customer makes a reservation for a taxi in a specific time in the near future.

1.7.2 Acronyms

- **DB:** Database.
- **DBMS:** Database Management System.
- **OS:** Operative System.
- **API:** Application Programming Interface.

1.7.3 Abbreviations

- **[Gn]** n-goal.
- **[Rn]** n-functional requirement.
- **[Dn]** n-domain assumption.

1.8 Reference Documents

- Specification document: Assignments 1 and 2 (RASD and DD).pdf.
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

1.9 Document Overview

This document is basically structured in four part:

- **Section 1** The introduction catches the key points of the system to be developed, clarifies the recurring terms used in the document and gives an high level description of software functionalities;
- **Section 2** The overall description, gives general information about the software product, focusing on constraints and assumptions;
- **Section 3** The specific requirements, containing typical scenarios and use cases. UML diagrams are provided;
- **Section 4** The appendix, this part contains some information about the.als file and some screenshots of the software used to generate it.

2 Overall Description

2.1 Product perspective

The application we're developing will be a web application and also a mobile app, such that the citizens can have more convenient ways of using the taxi service. The application will also replace the existing system: the call center which answers to calls and sends a taxi to each requested location by using the radio.

The application will have two interfaces: the first one for each taxi driver where he/she can take and decide to accept a call, change the taxi's status from occupied to free and vice versa; the second one is for the customer who is able to make instant and long term reservations and also to delete the previously made long term reservations. The call centre operators will use the same interface used by clients. The system will also have to store the queues of taxis from the different city zones in order to assign to each customer the closest vehicle.

2.2 User characteristics

The application will have four users:

- Visitor, who will be able to signup and login for the application;
- Registered user, who will want a simple and reliable application to call for a cab or to make a reservation for a latter time;
- Taxi driver, who will need a simple way of choosing to take or not an order, change the availability of the vehicle;
- Call Center operator, who will have a simple interface where he/she can fill out a form.

2.3 Constraints

2.3.1 Regulatory policies

myTaxiService has to meet the taxi regulatory policies imposed by the city's government and to privacy regulatory policies regarding users data.

2.3.2 Hardware limitations

myTaxiService doesn't have to meet any hardware limitations, since all computations are done on the server(s) and the portable device will have to transmit a limited amount of data.

2.3.3 Interfaces to other applications

myTaxiService doesn't have to meet any interfaces to other applications.

2.3.4 Parallel operation

The application will have to support parallel operations from different users when working with databases and will have to synchronize the communication between client and driver. It is expected that many clients use the application at the same time, especially during holidays, expositions, concerts or any other major event that may take place in the city.

2.3.5 Documents related

- ➔ Requirements and Analysis Specification Document (RASD).
- ➔ Design Document (DD).
- ➔ User's Manual.
- ➔ Testing report.

2.4 Assumptions and Dependencies

2.4.1 Assumption

- Each registered user can reserve up to one taxi on a specific time and date.
- An user can cancel a booked taxi only one hour prior the scheduled departure.
- It must exist at most one account associated to an email address.
- All users cannot cancel an order after they've entered the vehicle.
- Call center operators are obliged to use the new application when they have to book a taxi.
- All taxi drivers who are not available can not receive requests.
- While a driver decides whether or not to accept a call, he/she cannot receive another one.
- Taxi drivers have to be added to the system by manual DB insert.
- Tables binding the address to a city zone already exist.
- Each zone has exactly one queue of available taxis.
- Every taxi can be available only in one zone at a time.
- Each taxi has a unique identifier.
- Long term reservation are possible when the origin and destination of the ride is distant more than 500m.
- Every driver can signal to the system if they're available or not, but not both at the same time.
- Taxi drivers are not able to receive calls when their status is not available.
- The GPS signal will not be lost by taxis.
- In the text it's specified that the government of a city wants to upgrade their taxi service, thus stating that there is already an existing system that handles the transportation service but without giving the details of how it's organized. We can assume that there is a central call centre where different operators receive phone calls from customers, search for a cab through the station and tell the client the details.
- It's assumed that the system doesn't try to redistribute taxi to zones where the queue is empty.

- City is divided into zones which are not overlapping and all the zones cover the entire metropolitan area.
- The taxi service is available only within the metropolitan perimeter, whose limits have been discussed and approved from the government.

2.5 Future possible implementation

- Possibility for further enrichment of the user interface, so that one can have more options using the application. For example it can be developed another interface for taxi sharing where one user can book a taxi or more than one for groups of people, thus changing the tariffs accordingly.
- Enrich the user interface with a section where he/she can send a feedback about the last ride (such as unsatisfactory service, waiting time too big or the taxi never arrived).
- Create a profile for every registered user that is only visible by the owner and by some of the company employees. In the profile can be stored up to the n most recent travels.
- Expand the application so that it can route taxis around the city, so that it can guarantee a more extended management of taxi queues. For instance if the queue for a certain zone of the city is too long then then the system may decide to distribute part of the vehicles to the adjacent zone in order to guarantee a fair distribution of taxis. It will also be able to detect if a zone has no taxis at all, thus it will try to cover the affected zone by sending the last taxis in the queues of the adjacent zones.
- Realize a new system that monitors the efficiency of taxi drivers, so that if a driver surpasses the maximum fixed value of refused calls per week then the system will notify an administrator who will have to talk to him/her in order to find the reason behind this behavior.

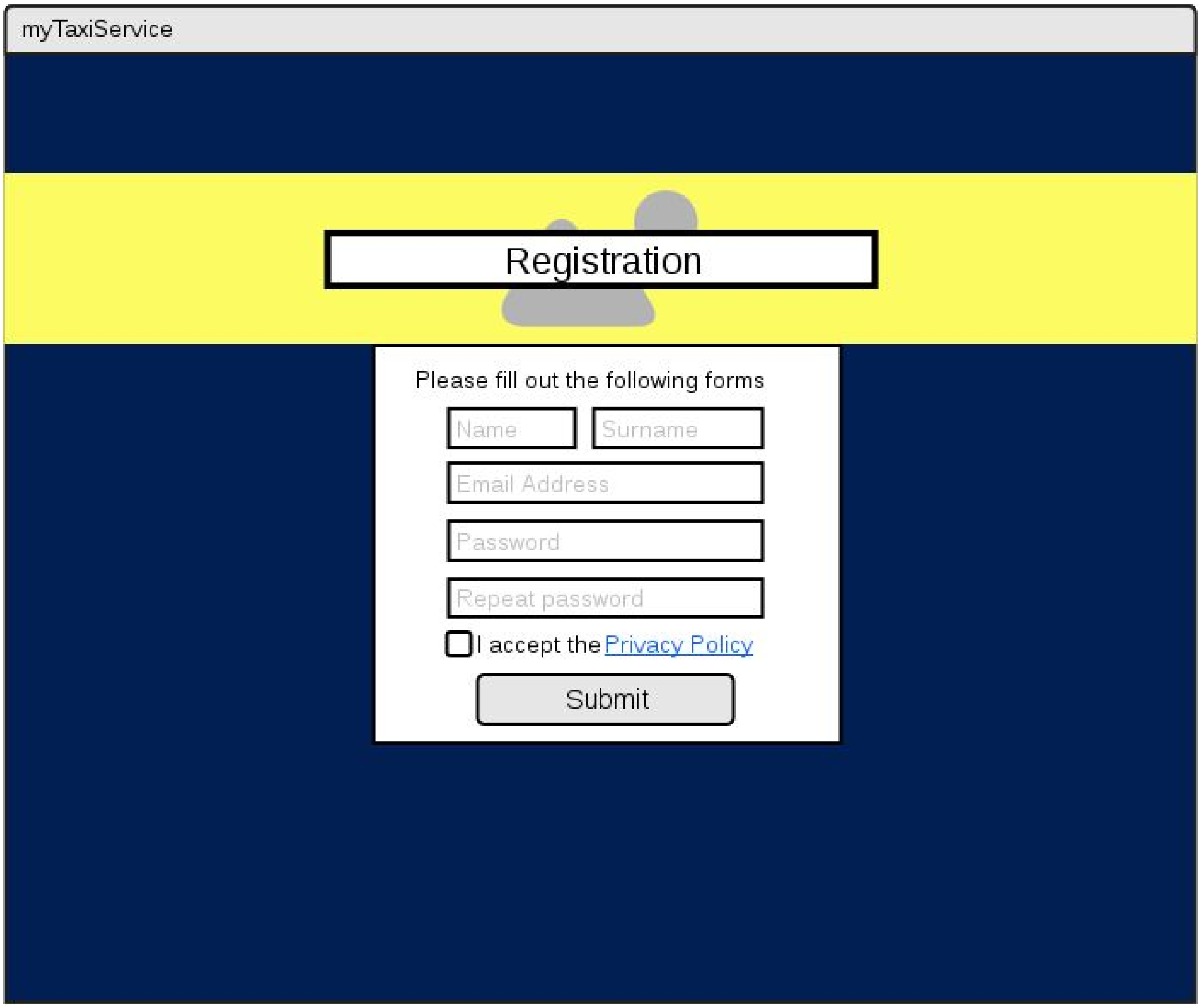
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Here there are some mockups representing an idea of the structure of the application pages. Note that the mockups have been done keeping the smartphone application in mind and that the web application would be quite similar to the app.

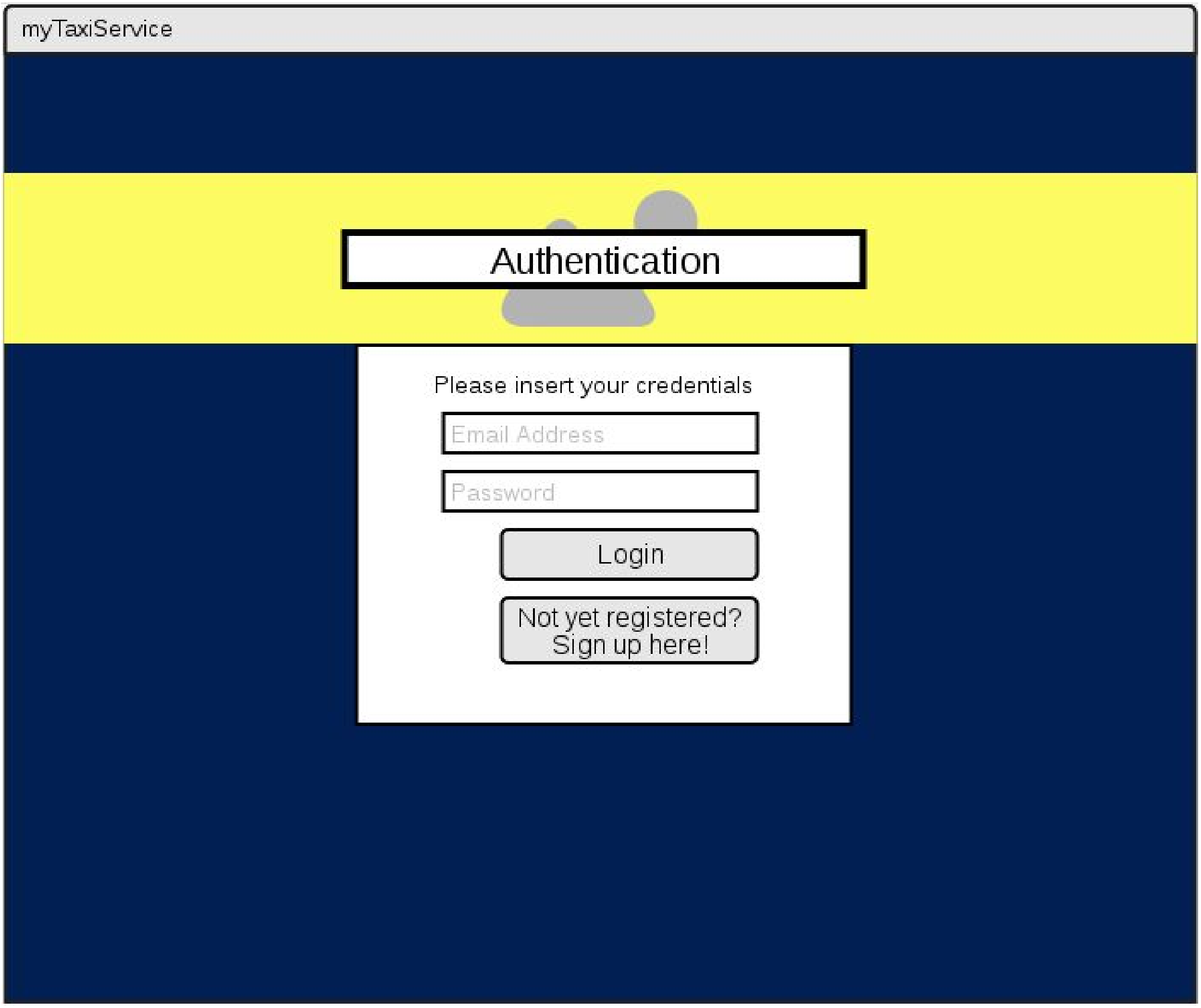
3.1.1.1 User Registration



Please fill out the following forms

□ I accept the Privacy Policy

3.1.1.2 User Login

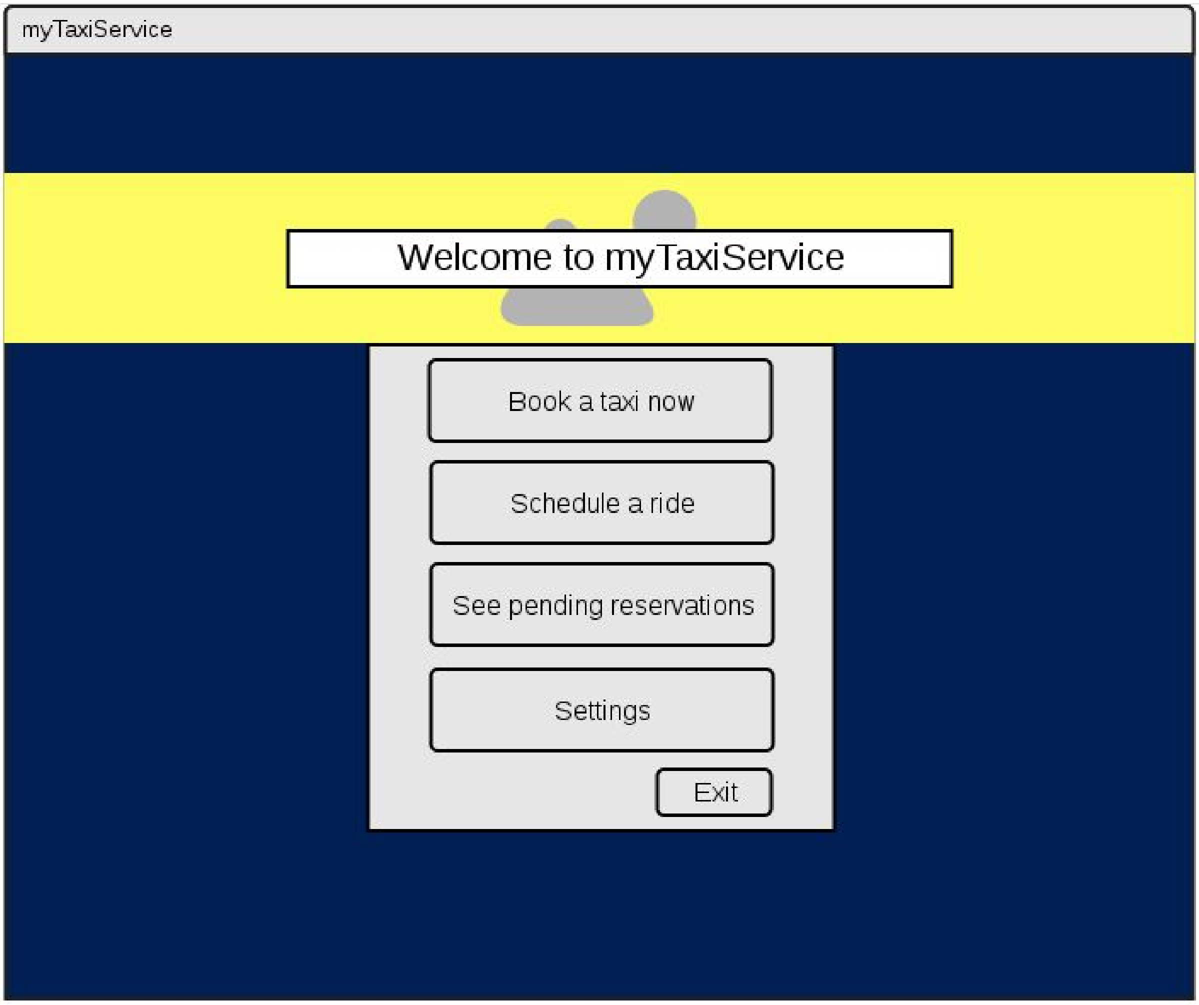


myTaxiService

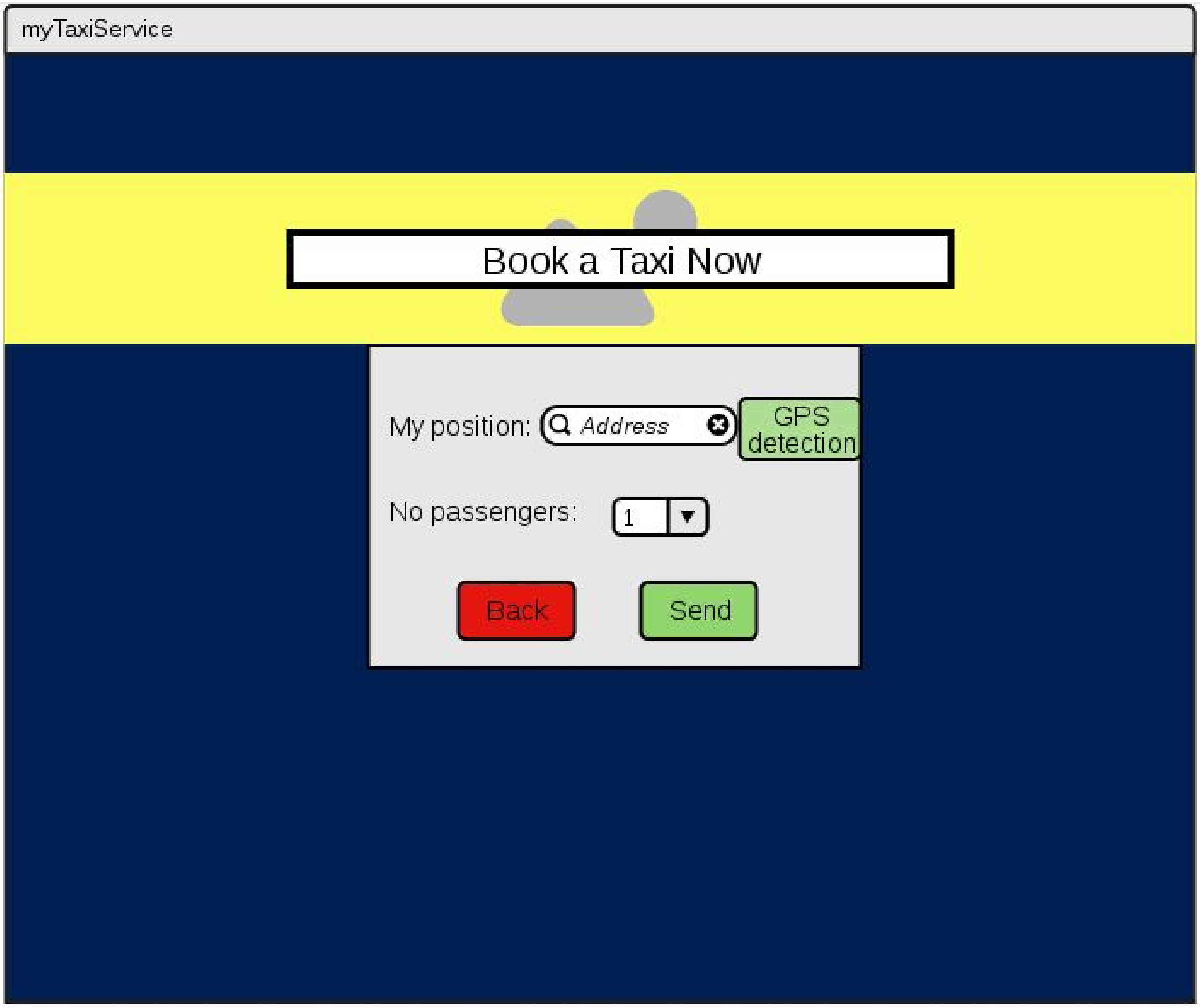
Authentication

Please insert your credentials

3.1.1.3 User Home



3.1.1.4 Book a Taxi Now



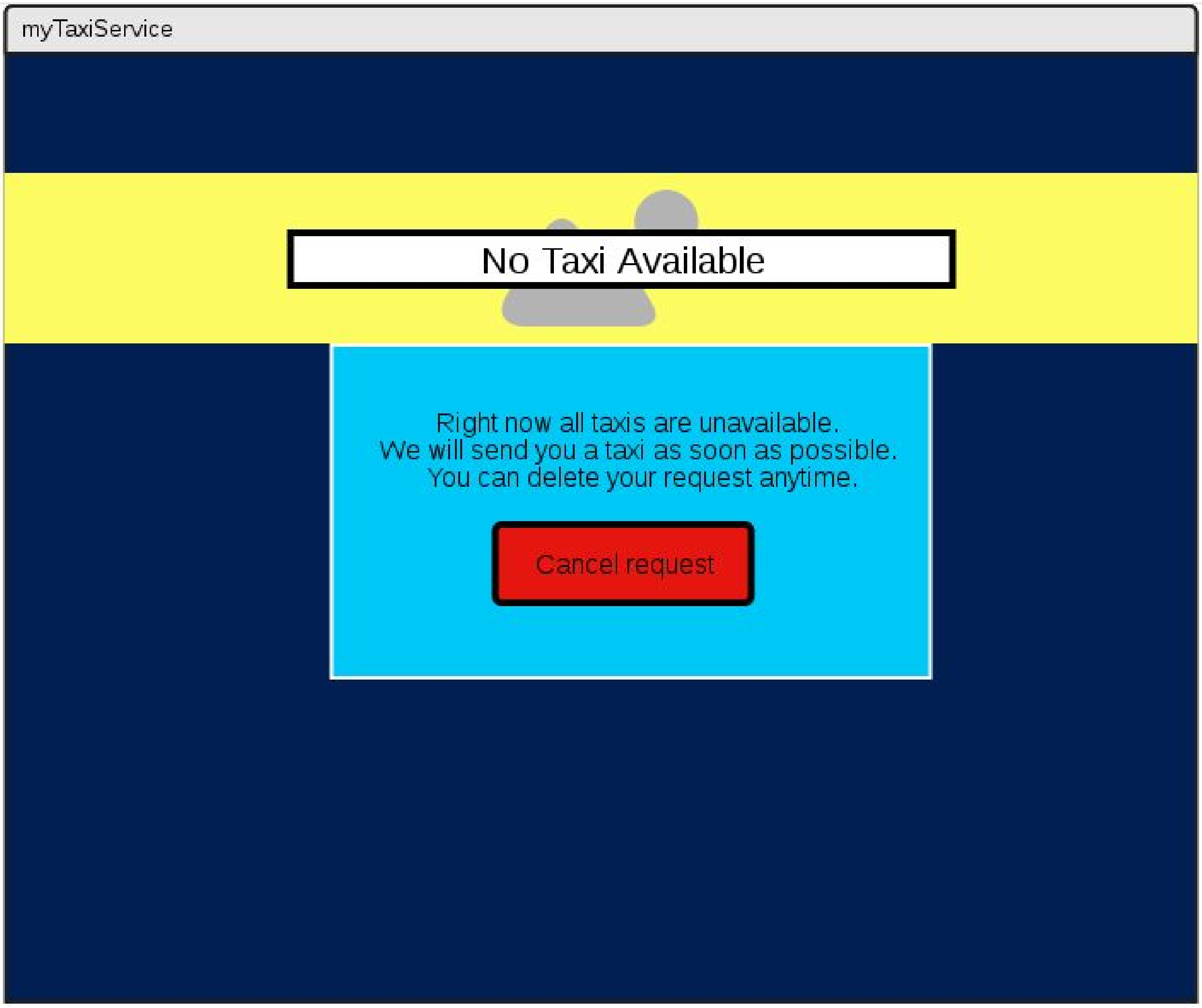
myTaxiService

Book a Taxi Now

My position: Address

No passengers: 1

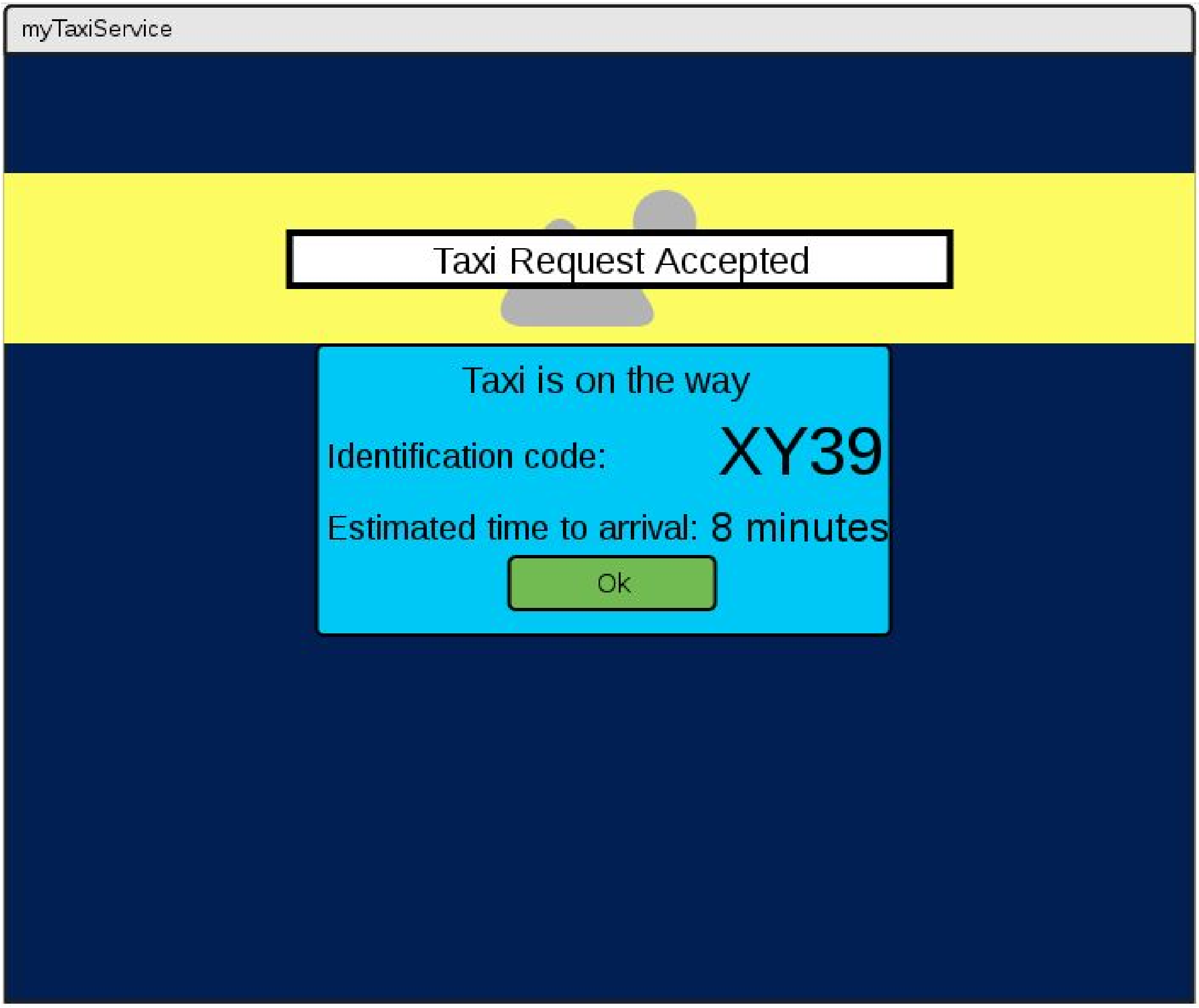
3.1.1.5 No Taxi Available



No Taxi Available

Right now all taxis are unavailable.
We will send you a taxi as soon as possible.
You can delete your request anytime.

3.1.1.7 Taxi is Arriving



myTaxiService

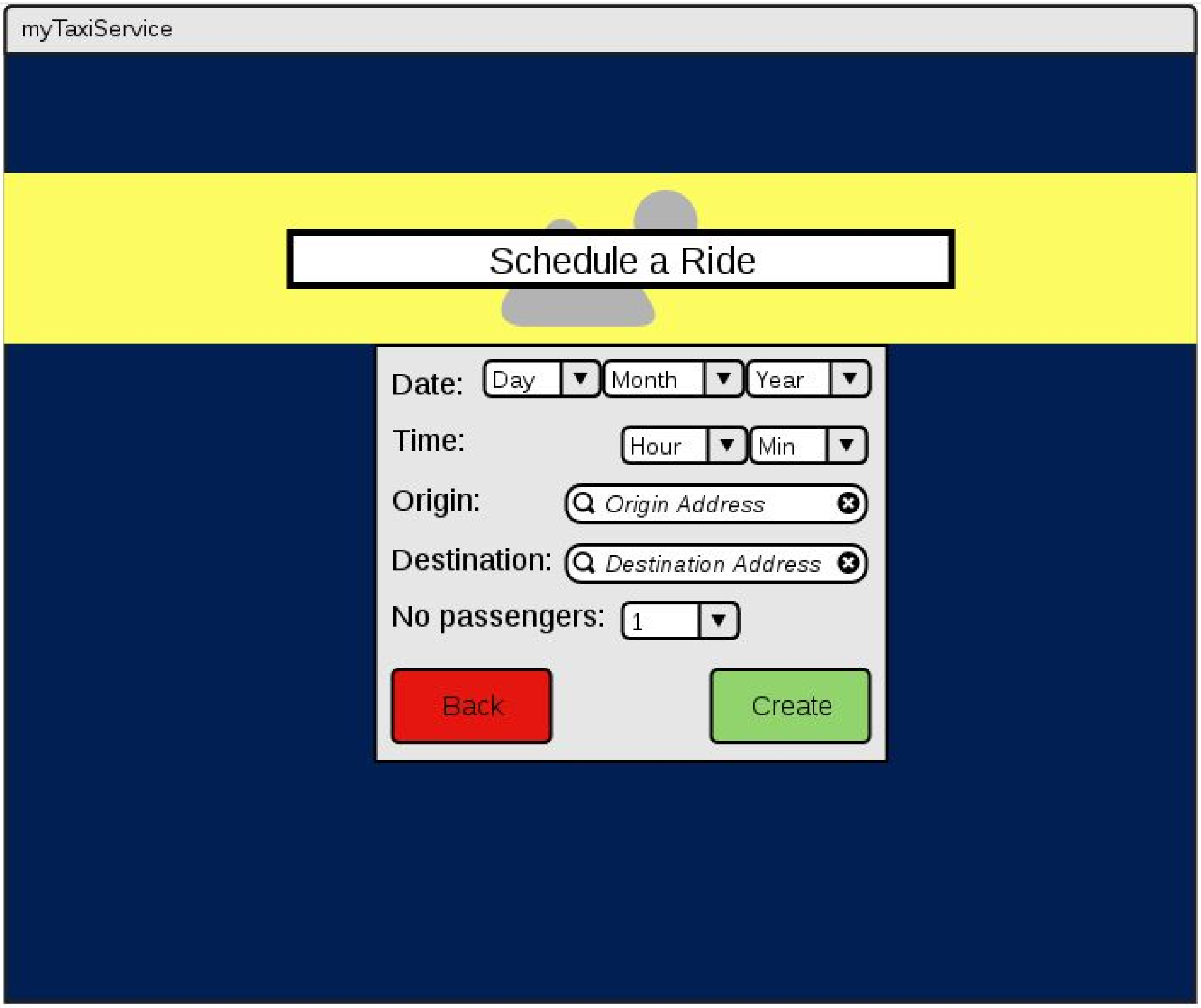
Taxi Request Accepted

Taxi is on the way

Identification code: XY39

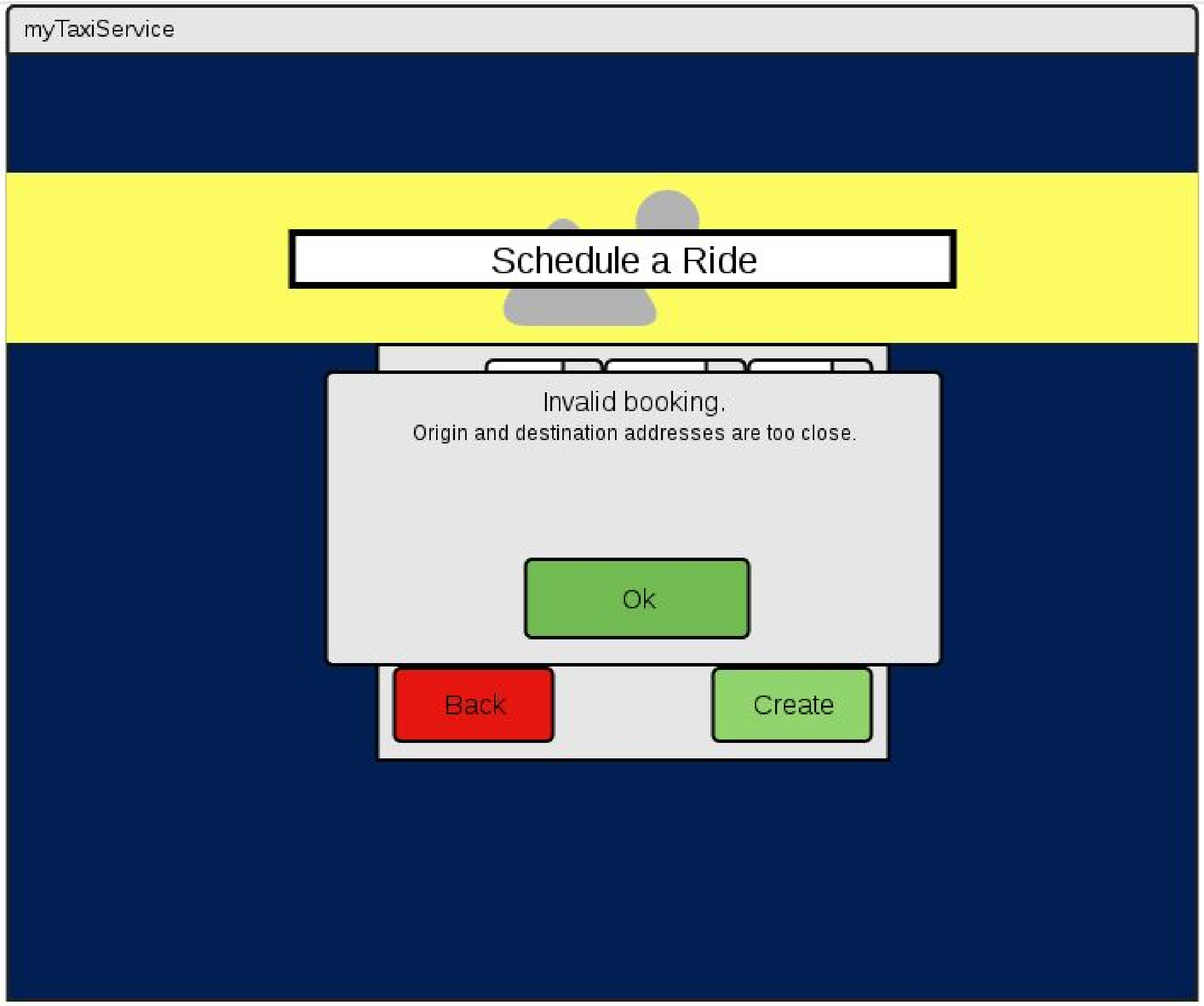
Estimated time to arrival: 8 minutes

3.1.1.7 Schedule a Ride



3.1.1.8 Invalid Booking

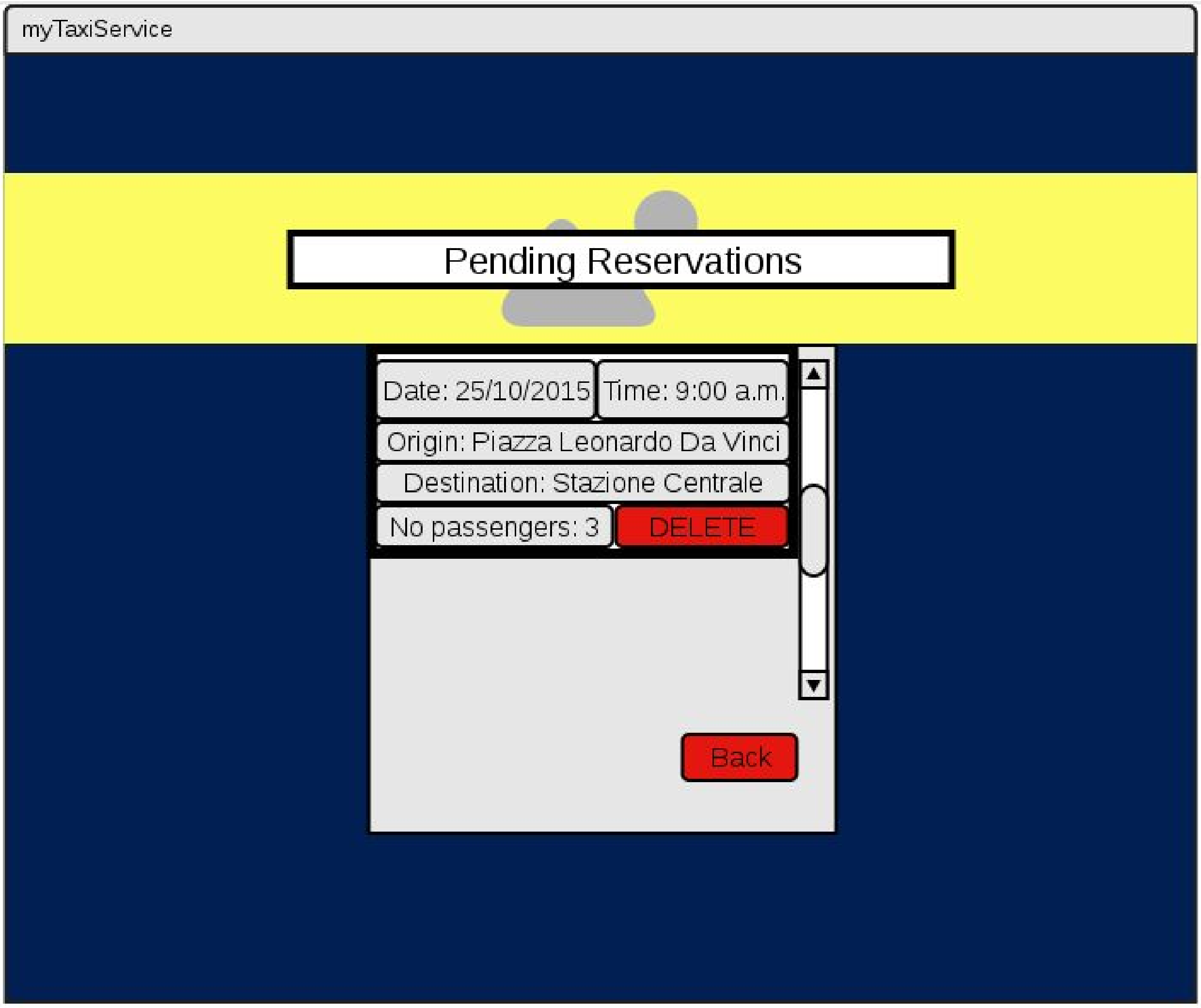
The error message displayed is just an example. In fact, the reservation done could also be too close (less than two hours away), or outside the urban perimeter.



Invalid booking.

Origin and destination addresses are too close.

3.1.1.9 Pending Reservations



Pending Reservations

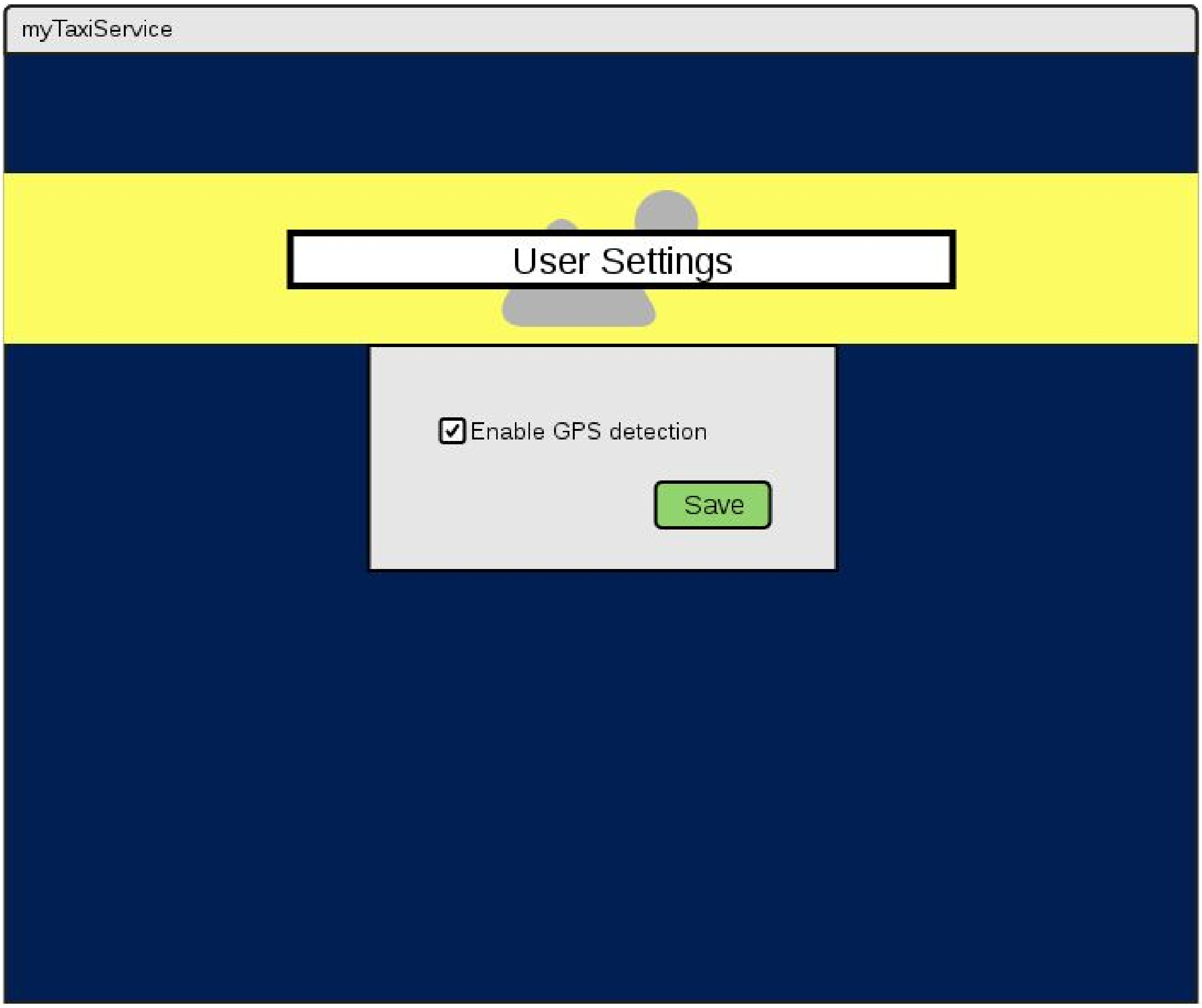
Date: 25/10/2015 Time: 9:00 a.m.

Origin: Piazza Leonardo Da Vinci

Destination: Stazione Centrale

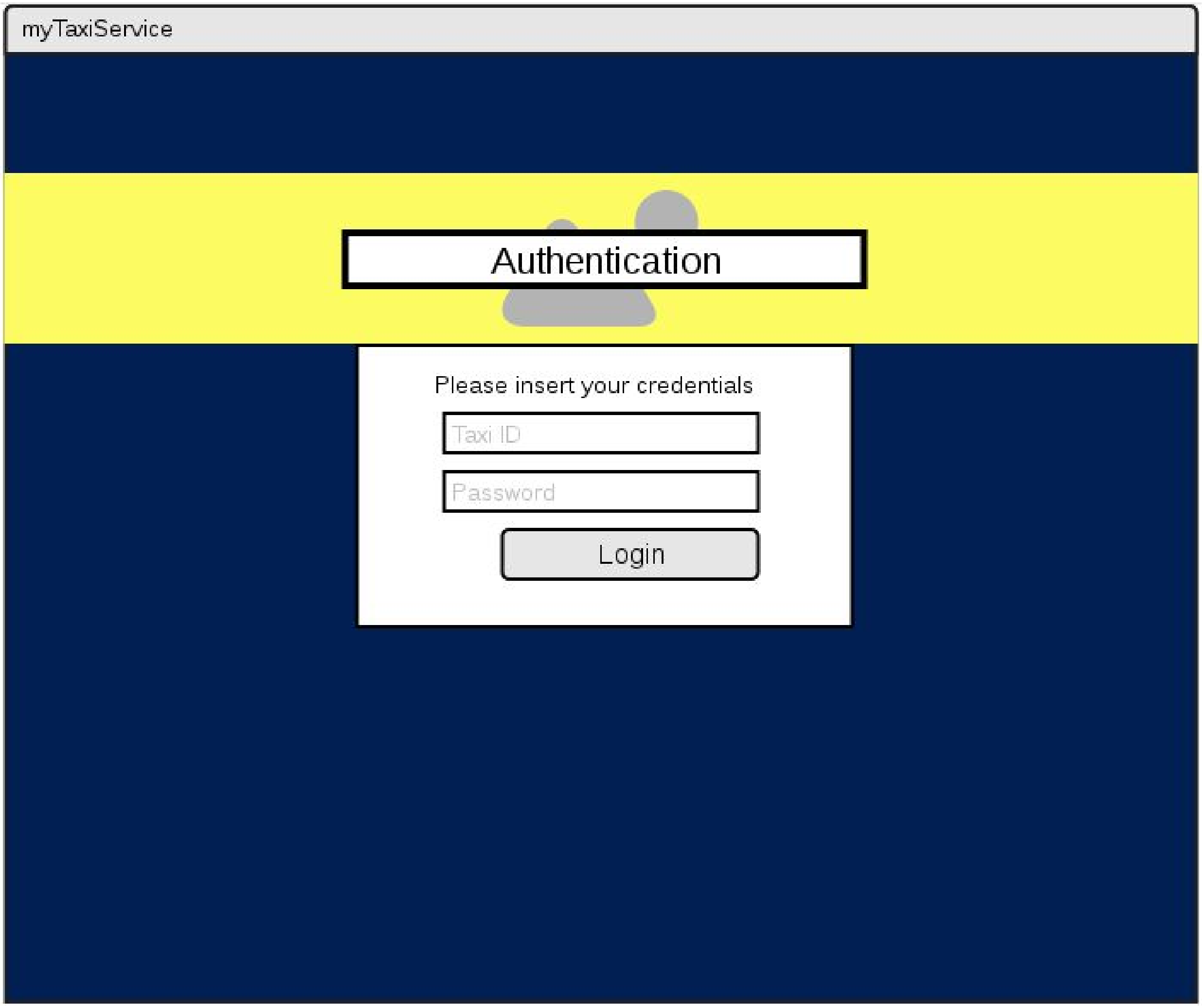
No passengers: 3

3.1.1.10 User Settings



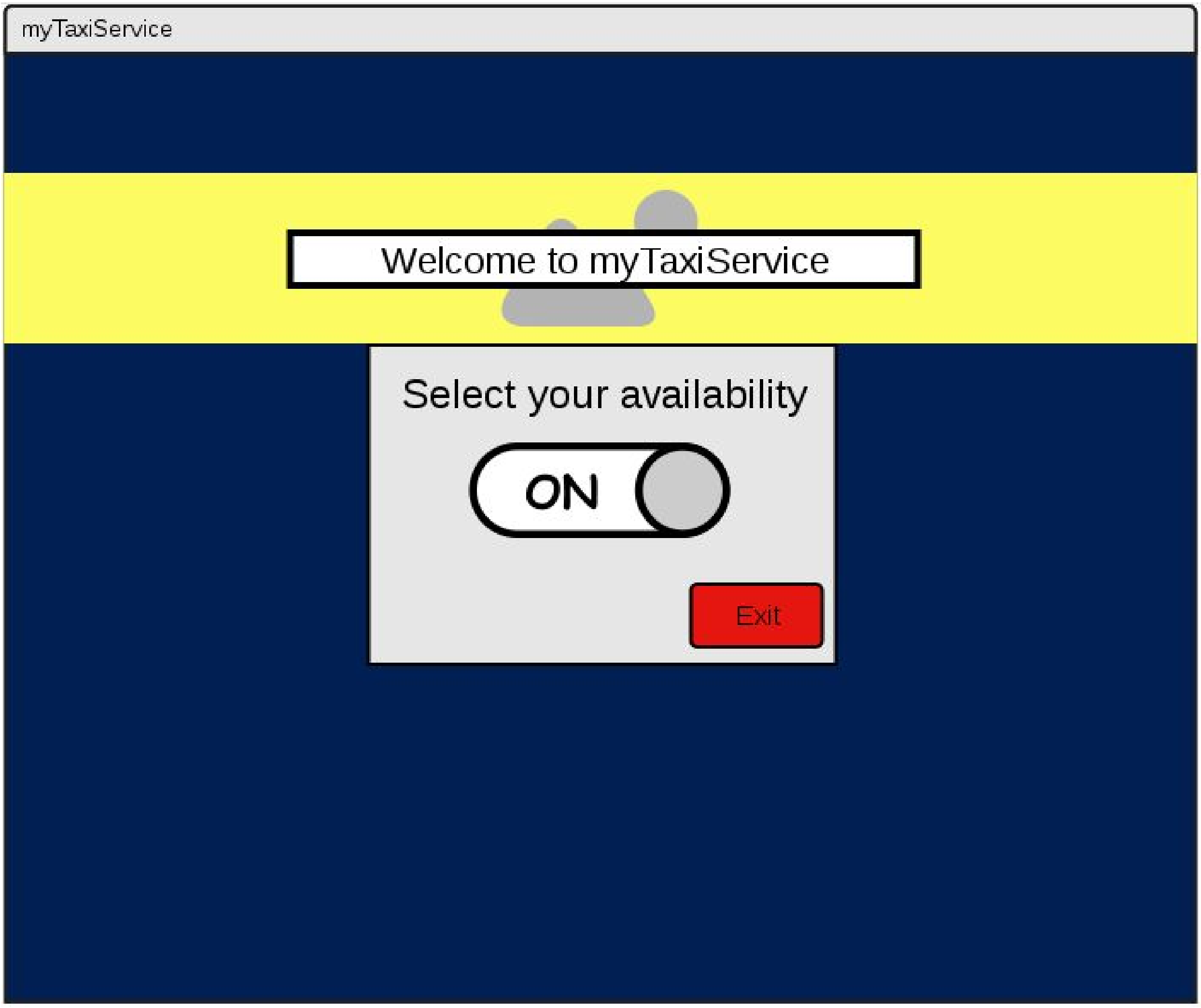
☑ Enable GPS detection

3.1.1.11 Taxi Login

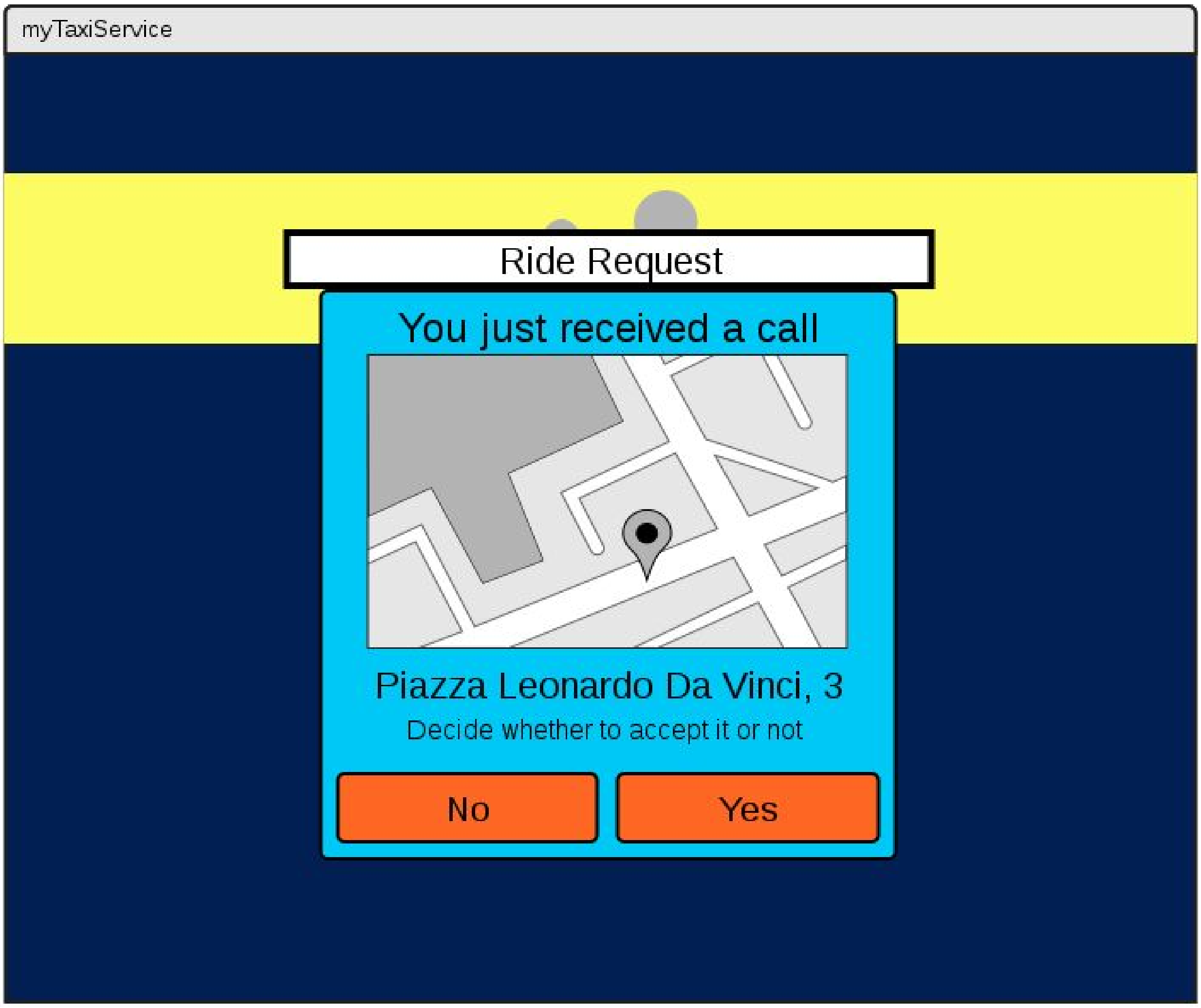


Please insert your credentials

3.1.1.12 Taxi Home



3.1.1.13 Ride Request

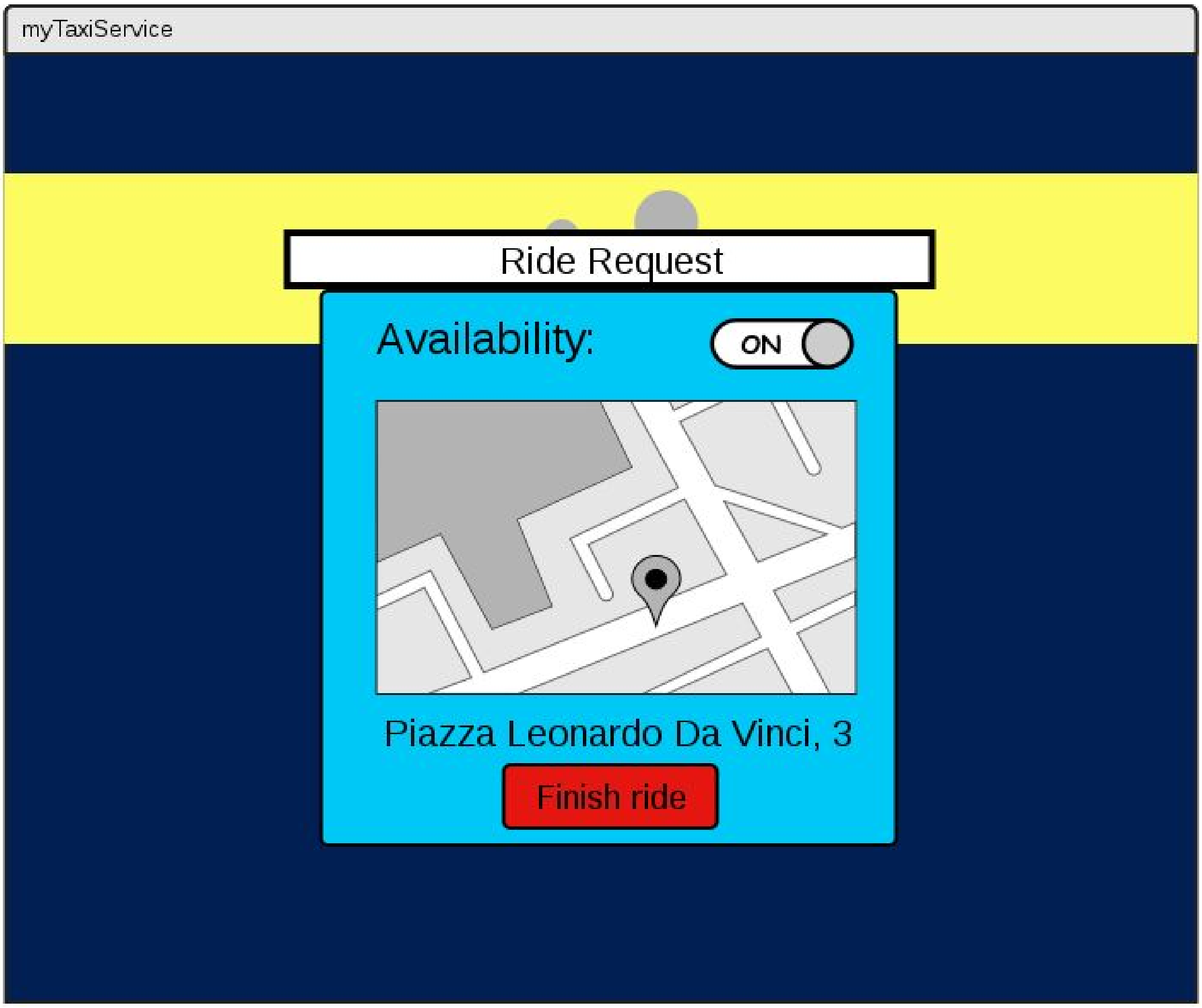


You just received a call

Piazza Leonardo Da Vinci, 3

Decide whether to accept it or not

3.1.1.14 Request Accepted



3.1.2 API Interfaces

myTaxiService will use the Google Maps APIs to auto-complete the text as the user fills the addresses forms out. This will happen during the instant reservation and the long term reservation too. Google Maps GPS will be used by the driver to get directions when he accepts a call. Google Maps is a widely used service and by implementing its APIs we are giving to the usability of our app a strong boost in terms of usability. For more informations on Google Maps APIs see its website (https://developers.google.com/maps/?hl=en).

3.1.3 Hardware Interfaces

This project does not support any hardware interfaces.

3.1.4 Software Interfaces

- Database Management System (DBMS)
    - Name: MySQL
    - Version: 5.7
    - Source: https://www.mysql.com/
- Java Virtual Machine
    - Name: Java Virtual Machine (JVM)
    - Version: 8
    - Source: https://java.com/it/download/
- Application Server
    - Name: GlassFish
    - Version: 4.1.1
    - Source: https://glassfish.java.net/
- Operative System (OS)
    - Application must be able to run on any OS that supports JVM and DBMS specified above.

3.1.5 Communication Interfaces

Protocol	**Application**	**Port**
TCP	HTTPS	443
TCP	HTTP	80
TCP	DBMS	3306

3.1.6 Memory

The minimum memory requirements are:

- Primary Memory: 1GB+
- Secondary Memory: 16GB+

3.2 Functional Requirements

3.2.1 [G1] To allow the visitor to register.

- **[R1]** Visitor must choose an email address not already registered.
- **[R2]** Passwords in both the input forms must be the same.
- **[R3]** Visitor must tick the privacy policy box.
- **[R4]** Password will be hidden while filling out the form.
- **[R5]** No form can be empty.
- **[R6]** By clicking on the submit button, user sends the data provided to the server.
- **[D1]** The email address must be formally correct.

3.2.2 [G2] To allow the unidentified user to log in.

- **[R1]** User will successfully login if the couple email address and password is be valid.
- **[R2]** Wrong credentials will prevent the user from using the application.
- **[R3]** Password will be hidden while filling out the form.
- **[R4]** After three failed login attempts (wrong credentials), device will not be able to login for fifteen minutes.

3.2.3 [G3] To allow the registered user to make an instant booking.

- **[R1]** Using the GPS detection will overwrite any text written in the address box.
- **[R2]** The number of passengers must be within one and seven.
- **[R3]** User can return to the home of the application.
- **[R4]** User can send the request to the server.
- **[R5]** The estimated waiting time is expressed in minutes and it is greater or equal than one.
- **[R6]** User can see the identifier of the taxi that accepted the call.
- **[D1]** The address manually inserted must be within the urban perimeter of the city.

3.2.4 [G4] To allow the registered user to make a reservation in a specified time.

- **[R1]** User can modify the hour, date, origin and destination addresses and number of passengers.
- **[R2]** The number of passengers must be within one and seven.
- **[R3]** Booking has to be made at least two hour before and up to seven days before the ride takes place.
- **[R4]** User can return to the home of the application.
- **[R5]** User can send the request to the server.
- **[D1]** Both origin and destination addresses must be within the urban perimeter of the city.
- **[D2]** Origin and destination addresses must be at least 500m away.

3.2.5 [G5] To allow the registered user to delete the reservations he made.

- **[R1]** User can delete a reservation up to fifteen minutes before the specified time.
- **[R2]** User can see the list of all the rides not already done.
- **[R3]** Reservations deleted by the users must be deleted from the server too.
- **[R4]** User can return to the home of the application.
- **[R5]** When user deletes a reservation, a popup asks if he wants to confirm the deletion.

3.2.6 [G6] To allow the taxi driver to change its availability status.

- **[R1]** Taxi driver can change the availability status anytime.

3.2.7 [G7] To allow the taxi driver to receive a call.

- **[R1]** The driver can see the map with the position of the customer.
- **[R2]** The driver can see the address of the customer.
- **[R3]** The driver can accept or decline the call.

3.2.8 [G8] To allow the taxi driver to get GPS directions until he reaches client's position.

- **[R1]** The driver can change the availability status.
- **[R2]** The driver can see a map and use GPS directions to reach the client.
- **[R3]** The driver can finish the ride.

3.2.9 [G9] To allow the taxi driver to login.

- **[R1]** Taxi will successfully login if the couple taxi ID and password will be valid.
- **[R2]** Wrong credentials will prevent the taxi from using the application.
- **[R3]** Password will be hidden while filling out the form.

3.4 Scenarios

3.4.1 Scenario 1

Ann has just landed in X city after a long flight and is excited to visit the grand art gallery of modern art that the city hosts during this period. She immediately rushes from the airport terminal to grab a cab and go check in at the hotel so she can leave her luggage in her room and go visit the gallery. When she arrives at the taxi reservation stand she finds out that there is a long queue caused by the malfunctioning computer which made taxi reservation at the airport,thus forcing Ann to wait until the workers fix it, delaying all her plans. Looking around she finds an ad about the app myTaxiService that offers an easy way to call for a taxi; so she takes out her smartphone, scans the QR code on the add and downloads the app. After the completing the installation process she can now move on to create an account providing name, email and password; after the app has verified that her email address is valid she logins on the app and asks for a cab at the airport Terminal 1. After five minutes she receives a notification with the code of the reservation, taxi number and that it'll arrive in front of the terminal in about 10 minutes. Ann is very pleased with myTaxiService, because it helped her save precious time.

3.4.2 Scenario 2

Bob has invited Jenny out for dinner at the new Japanese restaurant downtown that just opened last week. After a pleasant evening they decide to go back home, but unfortunately Jenny breaks a heel from one of her shoes and can't properly walk to the closest taxi station. Bob reassures Jenny that he will not walk such a long distance in such conditions, so he takes out his smartphone opens myTaxiService app and asks for cab for his current location. He receives a notification containing the reservation code, taxi number and that it will arrive in 5 minutes. Jenny praises Bob for being so kind with her.

3.4.3 Scenario 3

Charlie and Dan are heading back home after spending a wonderful week in city X, after they checked out from the hotel where they booked they go to the coach station to get to the airport. Unfortunately they've missed the 13:00 one and the next one departs in 30 minutes, thus making them lose their 14:30 flight. After discussing a bit they decide that they have to take a taxi if they want to get to the airport on time, so Charlie goes in a nearby pub and ask the owner if he can use the telephone to call for a cab. After getting in touch with an operator Charlie asks for a taxi to come pick them up at the coach station, so the operator opens the application and fills out the form with origin and destination of the ride. The system checks the queue of taxi of that particular zone, after the system has found a driver it forwards the details to the operator which in turn communicates them to Charlie. Luckily they arrive just in time at the airport.

3.4.4 Scenario 4

Eric and some of his friends want to see the rugby match between Y city RedBulls and X city WildCats that will hold next week at Grand Stadium in X city. So Eric decides to search for means of transportation on the internet on how to get to the stadium from the central railway station; so he stumbles upon myTaxiService application and decides with his buddies that reserving a taxi would be the fastest, easiest and most efficient way of getting to the stadium. So Eric decides to sign up in to the application and successfully completing the signup and login procedure, he then proceeds to make a long term booking. The application requests the details of the ride like origin, destination, number of passengers, date and time Eric also chooses to receive the confirmation containing the taxi number and waiting time through email as he does not want to install myTaxiService app on his smartphone. After completing all requested fields and after the system has validated the data, Eric receives an email containing the code details of the reservation and that an email will be sent to him 10 minutes prior to departure with the taxi number as stated in the reservation procedure.

3.4.5 Scenario 5

Frances has to meet up with a client downtown to discuss the details of their contract as she usually does she opens up myTaxiService app on her smartphone and request for cab to come at her location (using the GPS coordinates offered by her smartphone). But this time she receive a notification that there are no available taxis in her area and will be put on hold until the system can find a taxi. Frustrated by the fact that she forces her client to wait, she tries to search with other possible means of transportation in order to make it in time at the meeting. While frantically searching with local city transportation app she notices a cab was letting a customer get off, she approaches the vehicle and ask the driver if he's free and if he can drive her downtown to an important meeting. The driver answers positively and Frances gets in the car. She then proceeds to clearing all the opened applications on her smartphone, thus cancelling the order on myTaxiService app.

3.4.6 Scenario 6

Henry is a taxi driver in X city and is currently working his shift for today, he is eager to see how the new system works. He is the first in line at the taxi station of a zone and waits for his first clients, a family approaches Henry and asks him if he can drive them to the airport and he accepts. After he gets in the taxi he receives a notification of a call from the system, but he already answered another call and thus refuses, he then proceeds to change the status of the vehicle and drives to the airport.

3.4.7 Scenario 7

Isaac has just arrived in X city by train from an important meeting held in Y city and can't wait to get back home to his family. Unfortunately he discovers that the local transportation system is on strike today and can't take the bus. So he decides to call home and ask his wife Laura to come pick him up, but his daughter Katie answers and tells her father that Laura is still at work and doesn't know when she will be back. Katie suggests his dad to take a cab, but Isaac tells her that all of the taxis were taken already; she tells him that there's no problem as that she can book a taxi through myTaxiService application on her computer easily, as she often does when she goes out in the evenings. So she logs in the application and makes a reservation by manually inserting where the cab has to pick up the customer and tells his father the taxi number. Isaac arrives safely at home and thanks his daughter for helping him out of that situation.

3.4.8 Scenario 8

Mark is a taxi driver in X city, today is his first shift after his skiing accident. While he was recovering, the government of X city decided it was time to give new life to the city transportation system so they decided to upgrade the taxi service. Mark is anxious about how he will interact with the new system, as he never used one. He enters his car and starts it, the tablet also turned on with the car and asked for the driver's credentials. He inserts them and the application welcomes Mark, after just 10 minutes the application notifies Mark about an incoming call: a customer wants to be picked up on First Street get to Y Avenue. The application asks Mark if he accepts or declines the request, he gladly accepts because it's his first time after a very long time. The application also asks Mark if he wants a GPS system guide him where the client is; he gladly accepts the GPS assistance because X city has undergone a huge street redesign changing dual carriageway to single carriageways and vice versa and he is worried that he may lose himself and thus making the client wait too much.

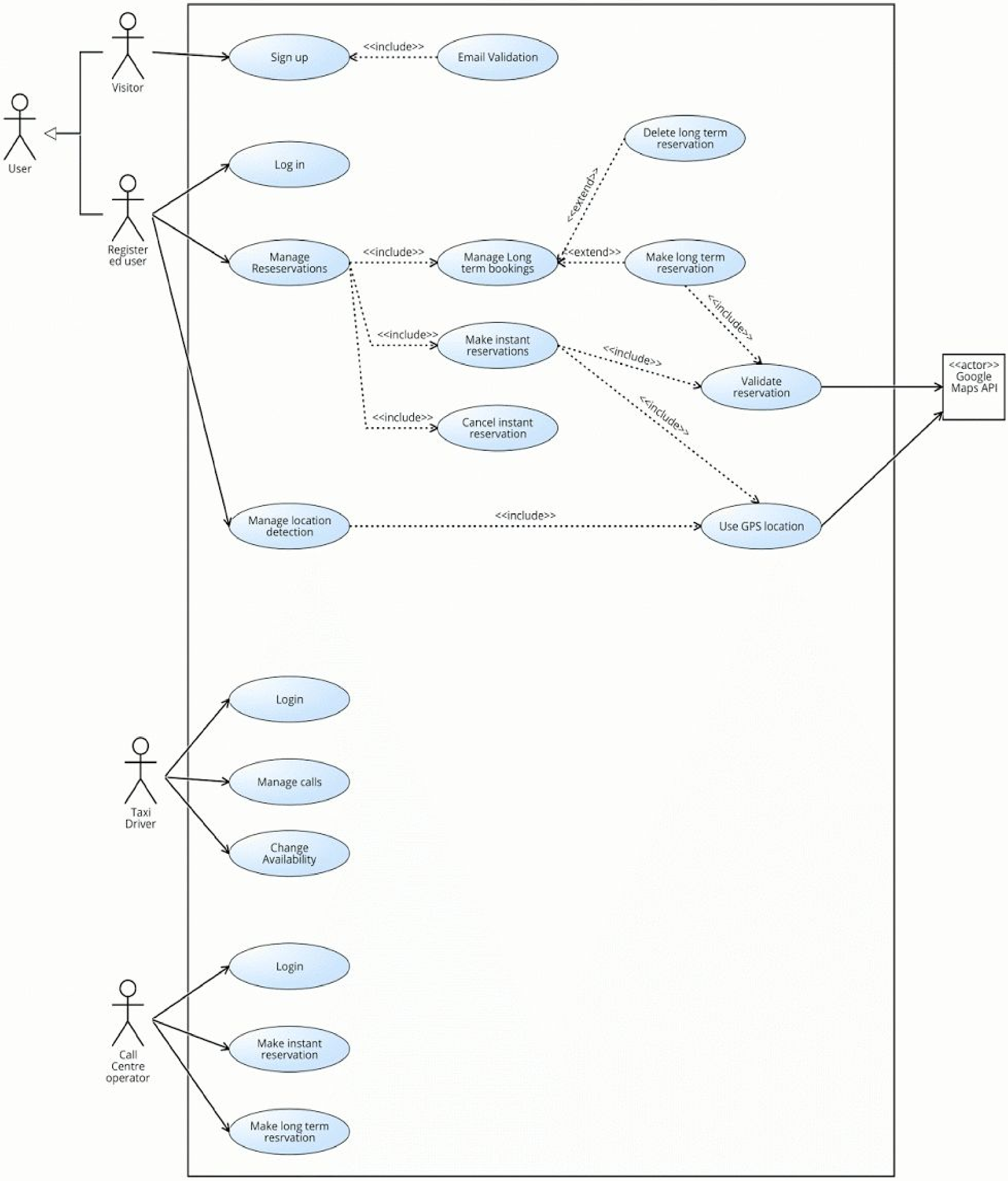
3.5 UML models

3.5.1 Use Case

We can derive some use cases from the scenarios identified in the previous paragraph:

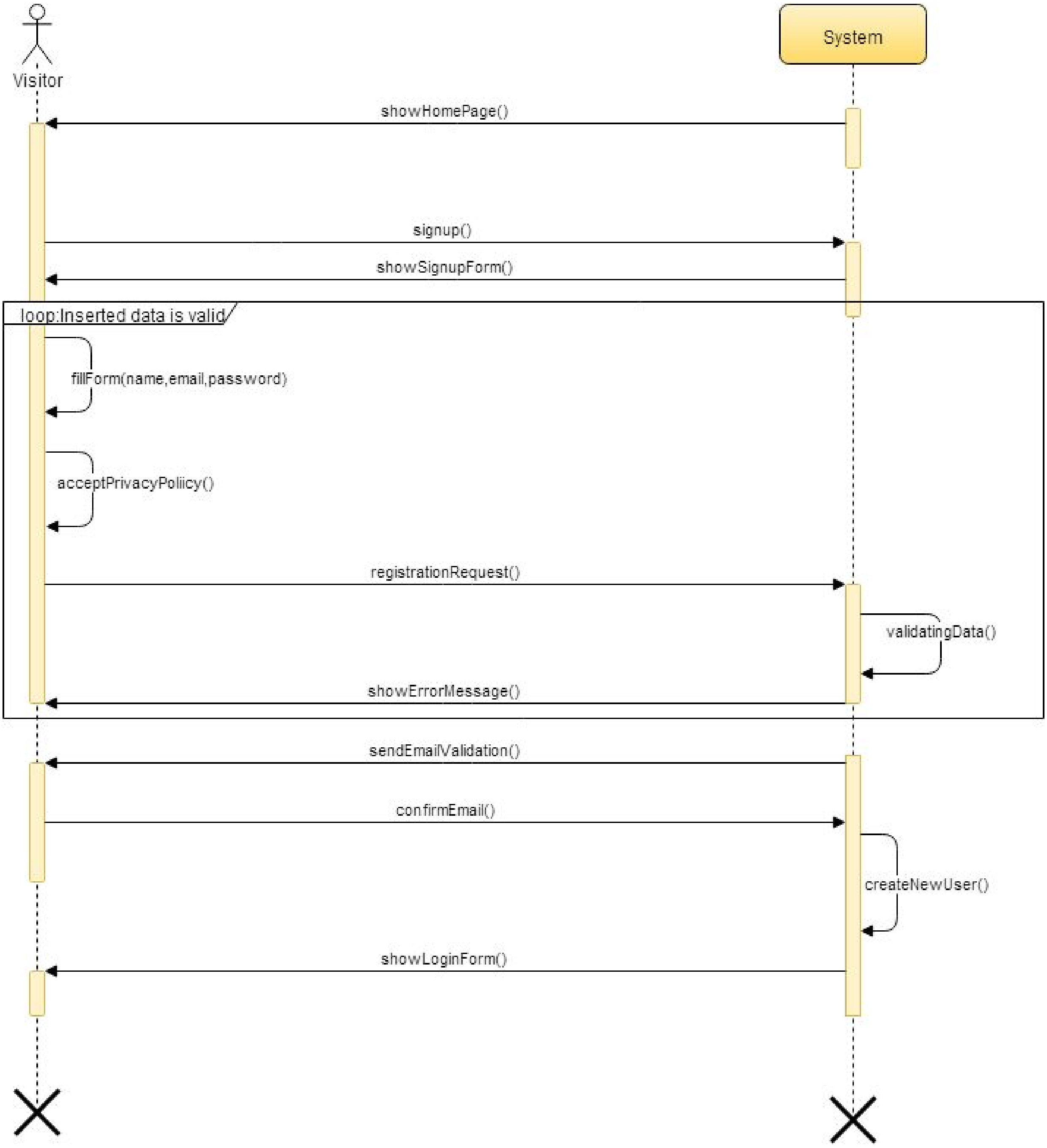
- Sign up for the application;
- User login;
- Taxi driver login;
- User makes an instant reservation;
- User makes a long term reservation;
- User management long term reservation;
- User changes the way the application gets his/her position;
- Taxi driver notification about a call;
- Change availability of the taxi;

myTaxiService Use Cases



Sign up use case

Actors	Visitors
Goal	G1
Input Condition	The visitor is not yet registered.
Flow of events	1. The user enters the home page of the application and clicks on the 'Signup' button, in order to enter the registration form. 2. The user must fill out all the forms: 2.1. Name; 2.2. Email; 2.3. Password; 2.4. Accept the application Privacy policy; 3. The visitor clicks on 'Submit'. 4. The system sends a message to the inserted email in order to validate it. 5. The user receives a confirmation request email and clicks on the provided link. 6. The system saves the profile of the new user on the data base. 7. Registration process complete: 7.1. The user is prompted on to the home page application where he/she can login. 7.2. The user is automatically logged into his/her account when registering through the app and he/she can make a reservation.
Output Condition	The visitor is now a registered and can start using the new service.
Exception	Exceptions may arise such as: ● if the email inserted is not a valid one; ● the email inserted is already associated to another account; ● if the user clicks 'Submit' even though he/she hasn't filled out all the forms; In all exceptions presented above the system alerts the user through pop ups with the error message.

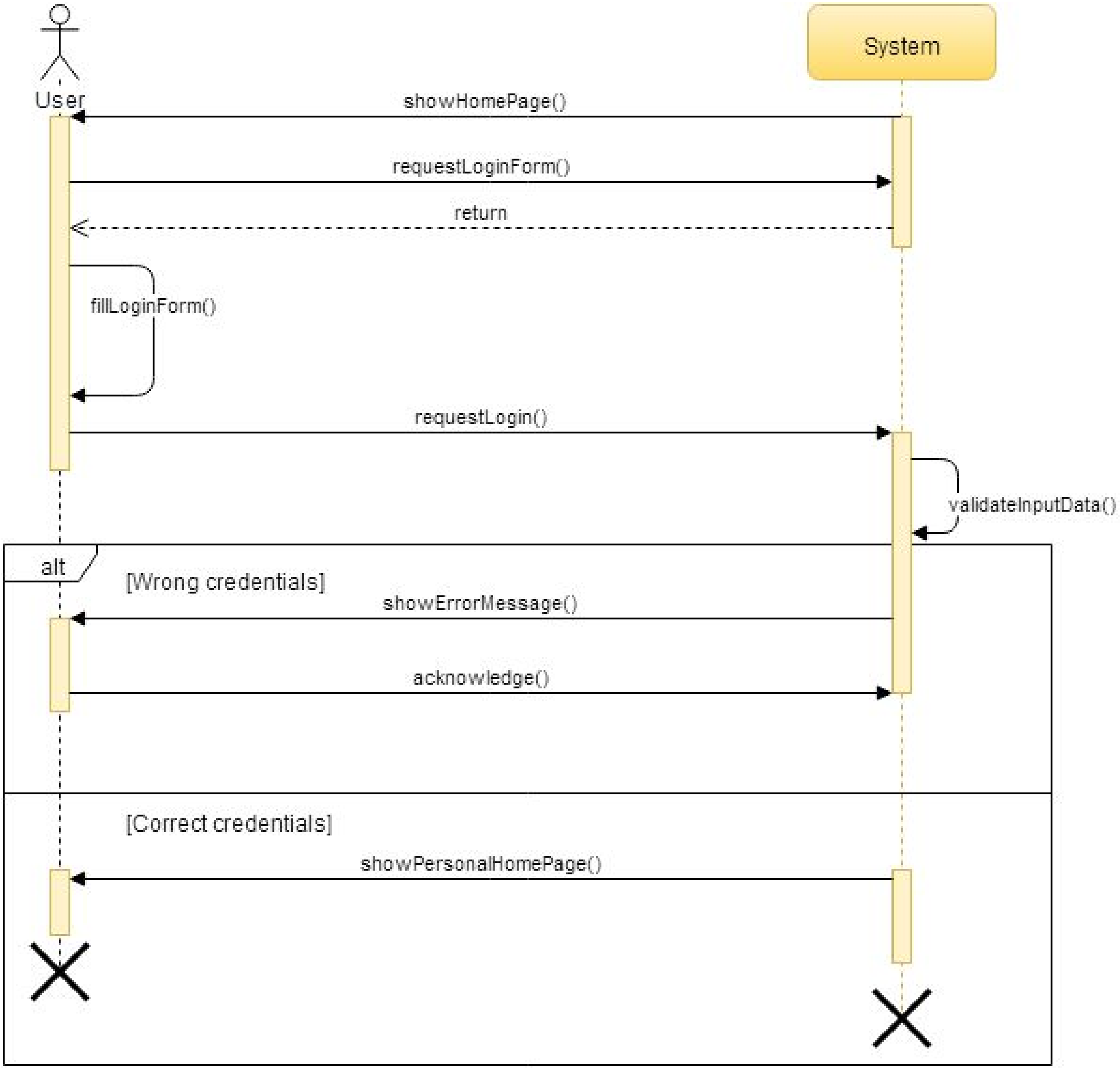

Visitor
System
showHomePage()
signup()
showSignupForm()
loop:Inserted data is valid
fillForm(name,email,password)
acceptPrivacyPoliicy()
registrationRequest()
validatingData()
showErrorMessage()
sendEmailValidation()
confirmEmail()
createNewUser()
showLoginForm()

User login

Actors	Registered user, Call Centre operator, taxi driver
Goal	G2
Input Condition	The user must be already registered.
Flow of events	1. The user enters the home page of the application and clicks on the ‘Login’ button. 2. He/She fills out the required forms and clicks the ‘Submit’ button: 2.1. Email and password for registered users; 2.2. Driver ID for taxi drivers; 2.3. Worker ID for Call Centre operators; 3. The system verifies that the inserted data is correct. 4. The user is presented with his personal home page.
Output Condition	The user has successfully logged into his/her account and can start using the functionalities the system offers.
Exception	Exceptions may arise such as: ● the inserted password may not be correct; ● the inserted email is not contained in the database; In all exceptions presented above the system alerts the user through pop ups with the error message.

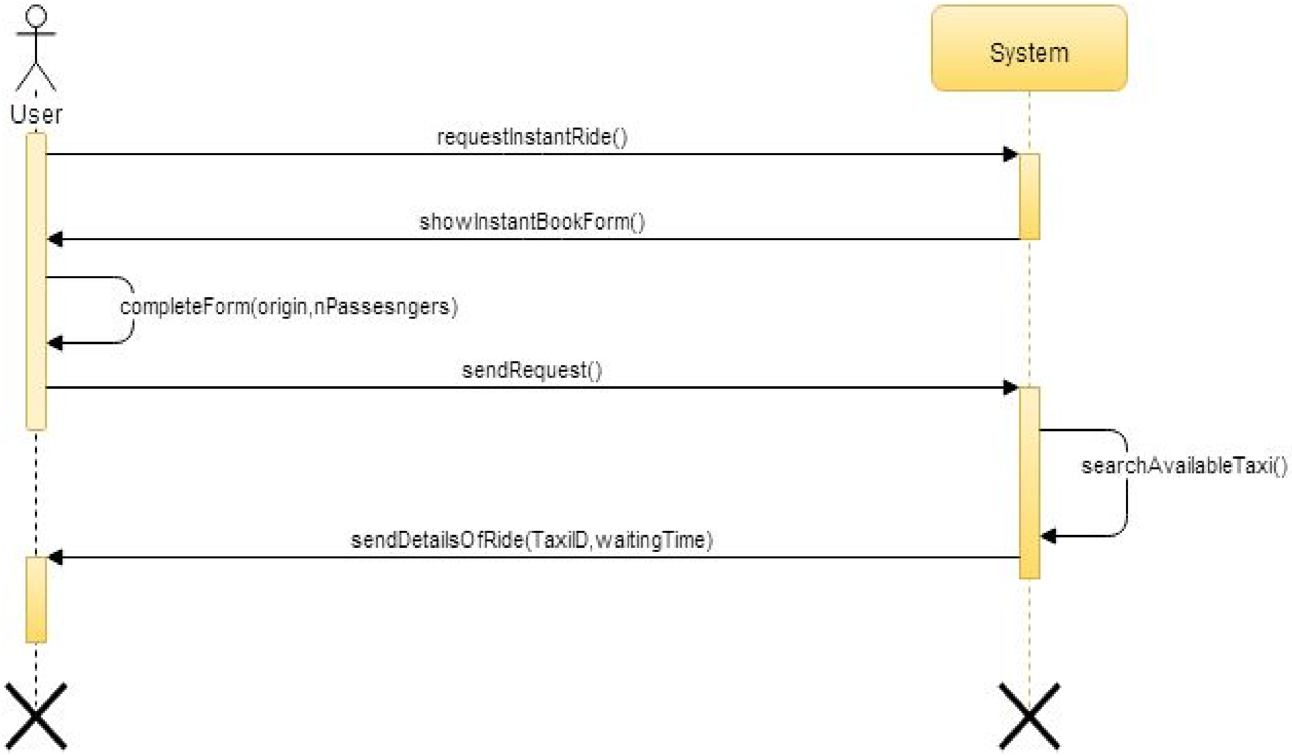
Taxi driver login

Actors	Taxi driver
Goal	G9
Input Condition	The driver must be already be registered.
Flow of events	5. The driver enters the home page of the application and clicks on the ‘Login’ button. 6. He/She fills out the required forms and clicks the ‘Login’ button: 6.1. Driver ID; 6.2. Password; 7. The system verifies that the inserted data is correct. 8. The driver is presented with his personal home page.
Output Condition	The driver has successfully logged into his/her account and can start using the functionalities the system offers.
Exception	Exceptions may arise such as: ● the inserted password may not be correct; ● the inserted driver ID is not contained in the database; In all exceptions presented above the system alerts the user through pop ups with the error message.


User
System
showHomePage()
requestLoginForm()
return
fillLoginForm()
requestLogin()
validateInputData()
alt
[Wrong credentials]
showErrorMessage()
acknowledge()
[Correct credentials]
showPersonalHomePage()

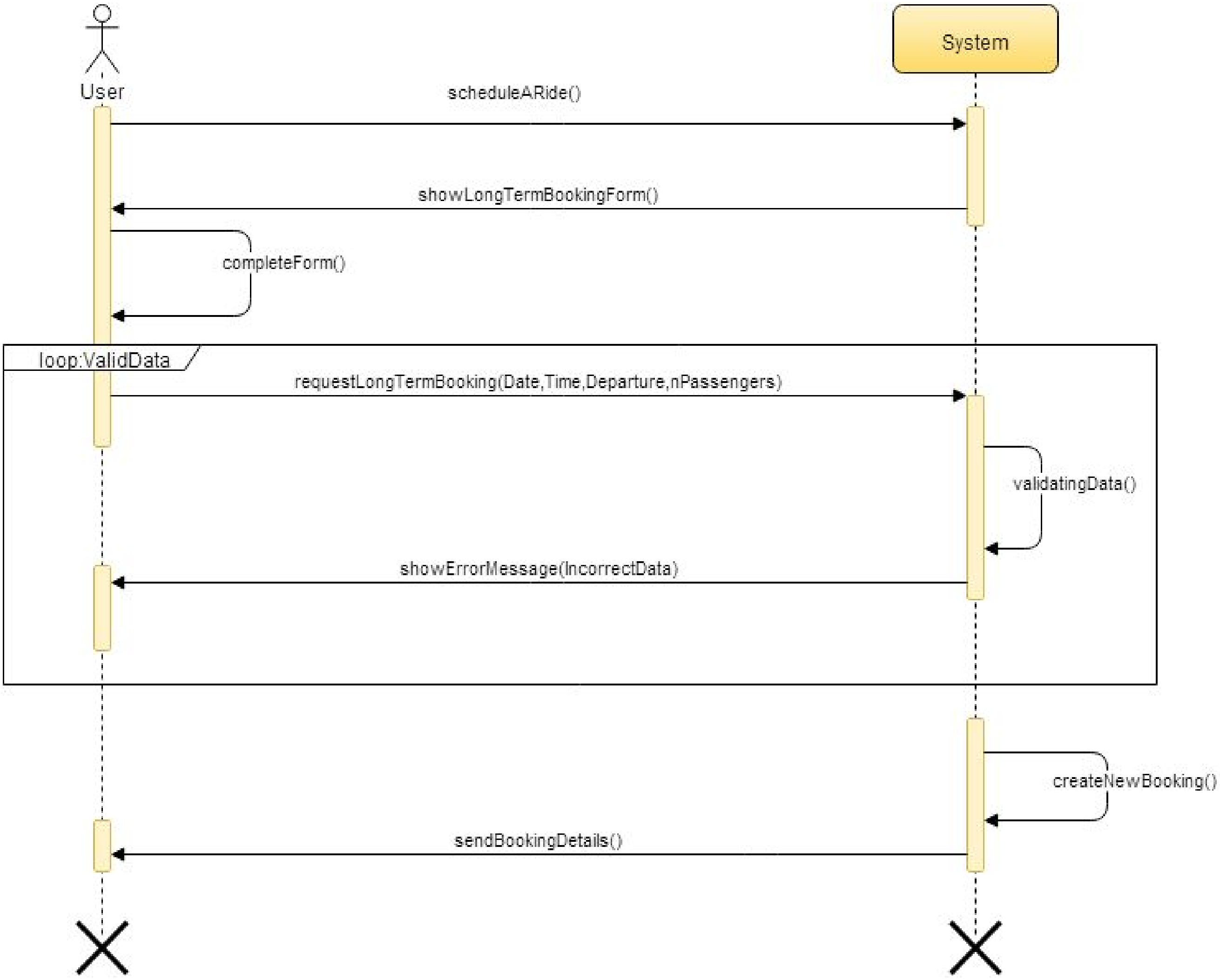
Instant reservation

Actors	Registered user, Call Centre operator
Goal	G3, G3.1, G3.2, G3.3, G3.5
Input Condition	The user must be already logged in the application.
Flow of events	1. The user must select the ‘Book a taxi now’ from his/her personal homepage. 2. The user will fill out the requested forms: 2.1. Address of where to pick up the client: 2.1.1. The Call Centre operator has to manually insert the location of the client; 2.1.2. The registered user can also manually insert the position or let the application get his/her position with the GPS of the device by clicking ‘GPS detection’; 2.2. Number of passengers; 3. The user clicks the button ‘Send’. 4. The system searches for a taxi that meets the inserted requirements through the queue of the zone where the client is. 5. The reservations ends in two different ways: 5.1. The system sends a confirmation with the taxi code, and waiting time to the user that made the reservation. 5.2. The Call Centre operator tells the details of the ride to the client and ends the call.
Output Condition	The user successfully made an instant reservation and the customer waits for his/her taxi to arrive.
Exception	Exceptions may arise such as: ● the inserted origin may not be correct; ● there are no available taxi for the zone where the client is; In all exceptions presented above the system alerts the user through pop ups with the error message.


User
System
requestInstantRide()
showInstantBookForm()
completeForm(origin,nPassesngers)
sendRequest()
searchAvailableTaxi()
sendDetailsOfRide(TaxiID,waitingTime)

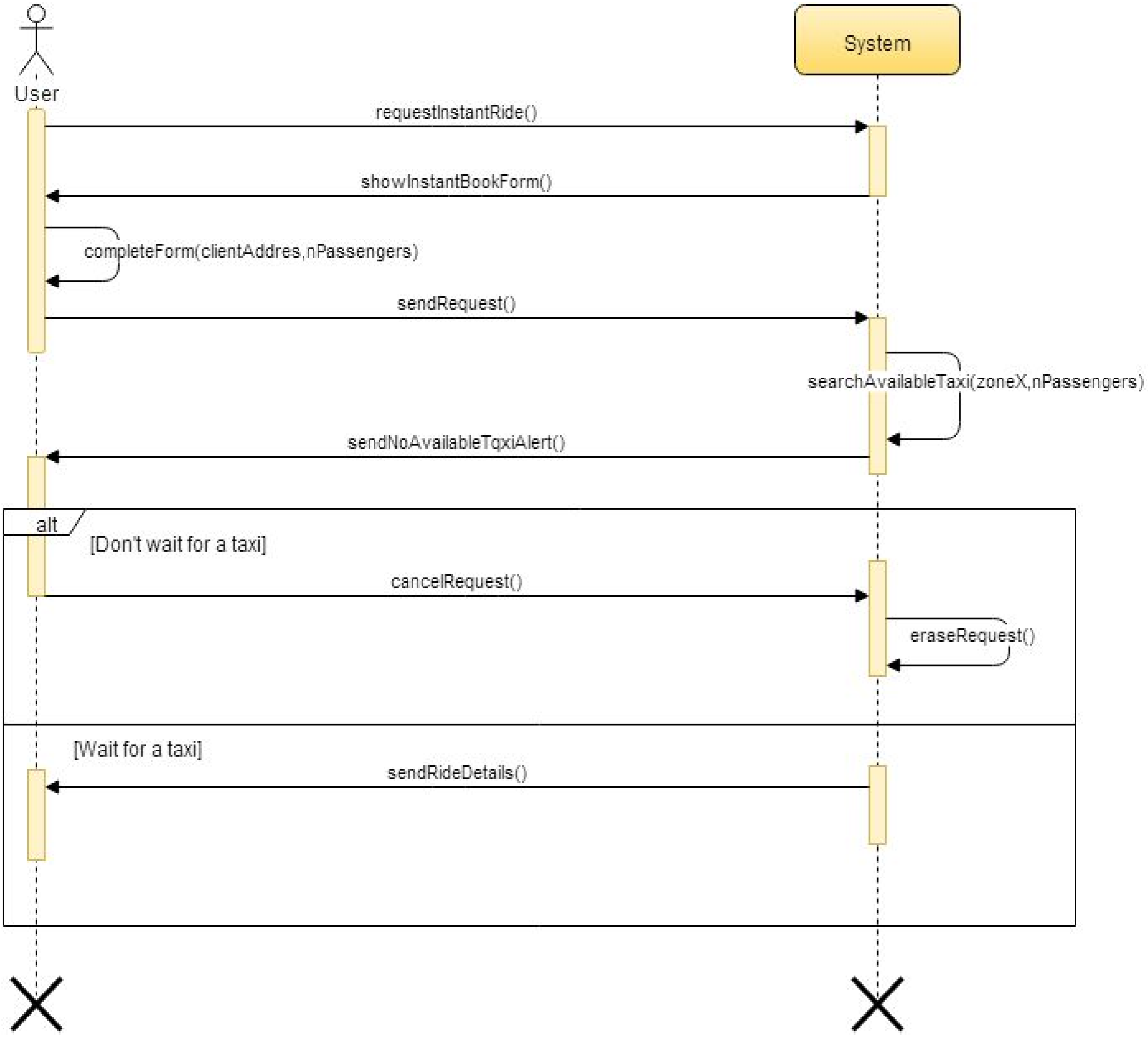
Long term reservation

Actors	Registered user, Call Centre operator
Goal	G4, G4.1, G4.2
Input Condition	The user must be logged into the application.
Flow of events	1. The user from his/her personal home page clicks on ‘Schedule a ride’. 2. The user will fill out the requested forms: 2.1. Position of where to pick up the client; 2.2. Destination of the ride; 2.3. Number of passengers; 2.4. Date and time of the ride; 3. The user clicks the button ‘Create’. 4. The system verifies if the reservation respects the requirements of a valid long term reservation. 5. The system sends a confirmation with code reservation to the user that created it. 6. The reservation procedure ends in two ways: 6.1. The Call Centre operator tells the details of the ride to the client and ends the call. 6.2. The system saves the new reservation in the user’s section ‘See pending reservations’.
Output Condition	The user successfully made an long term reservation and if the reservation has been made by a registered user he/she can modify it on later time.
Exception	Exceptions may arise such as: ● the inserted origin may not be correct; ● the inserted destination may not be correct; ● the distance between the two addresses is less than 500m; ● the date is in the past; ● the date and time of the departure are not two hours before and a week after the departure; In all exceptions presented above the system alerts the user through pop ups with the error message.


User
System
scheduleARide()
showLongTermBookingForm()
completeForm()
loop:ValidData
requestLongTermBooking(Date,Time,Departure,nPassengers)
validatingData()
showErrorMessage(IncorrectData)
createNewBooking()
sendBookingDetails()

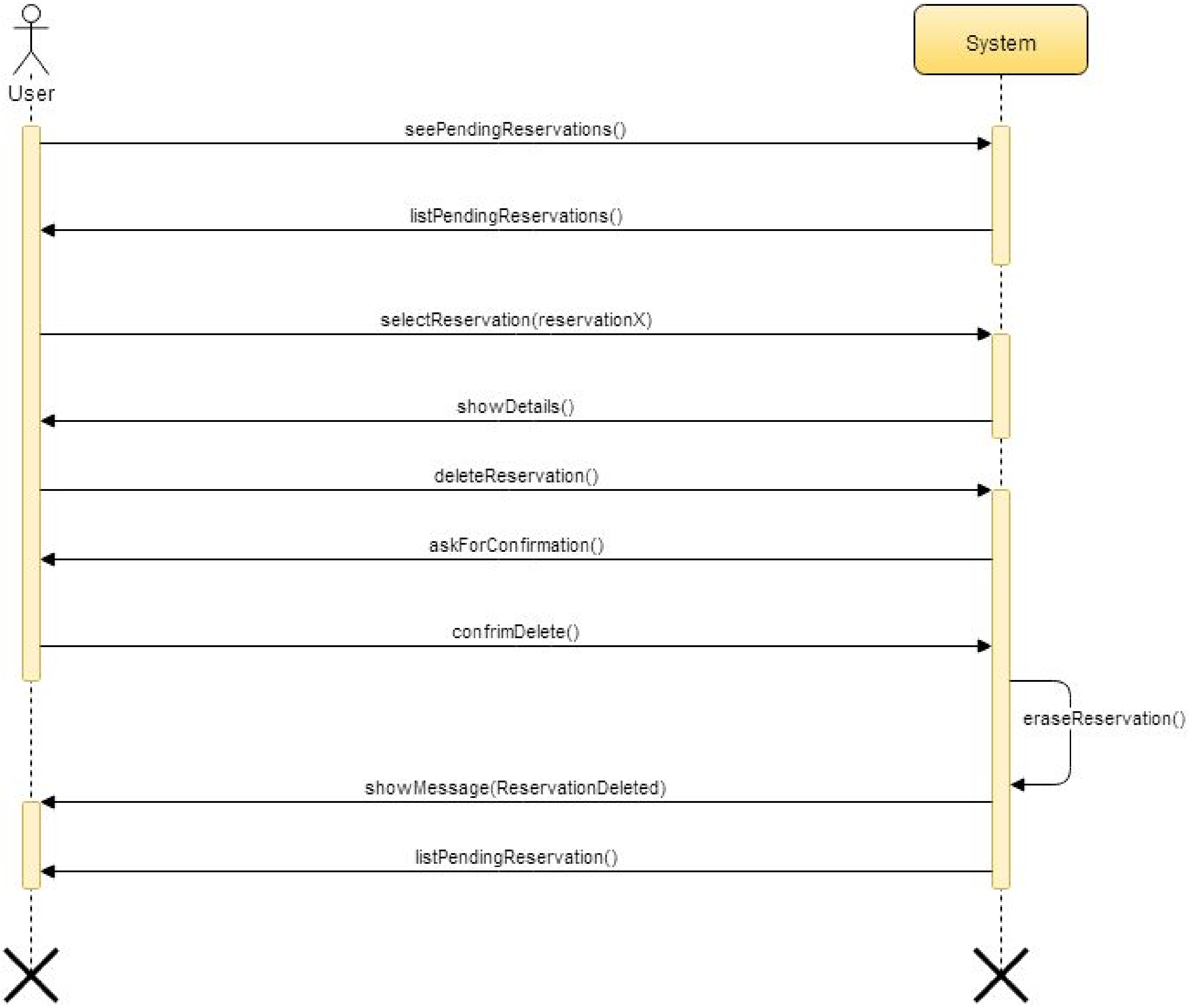
Instant reservation cancellation

Actors	Registered user
Goal	G3.4
Input Condition	The user must have made an instant reservation and there are no available taxis in the zone where he/she is.
Flow of events	1. After the user has made an instant reservation, he is presented with a message that there aren't any taxi available for the zone he/she is and will have to wait until the system finds one ; 2. The user clicks on the button 'Cancel request' shown underneath the message. 3. The system deletes the request.
Output Condition	The user cancelled an instant reservation he/she made, thus he/she is no longer obliged to wait for a taxi and can find alternative ways of reaching his/her destination.
Exception	There are no possible exception that we know of.


User
System
requestInstantRide()
showInstantBookForm()
completeForm(clientAddres,nPassengers)
sendRequest()
searchAvailableTaxi(zoneX,nPassengers)
sendNoAvailableTqxiAlert()
alt
[Don't wait for a taxi]
cancelRequest()
eraseRequest()
[Wait for a taxi]
sendRideDetails()

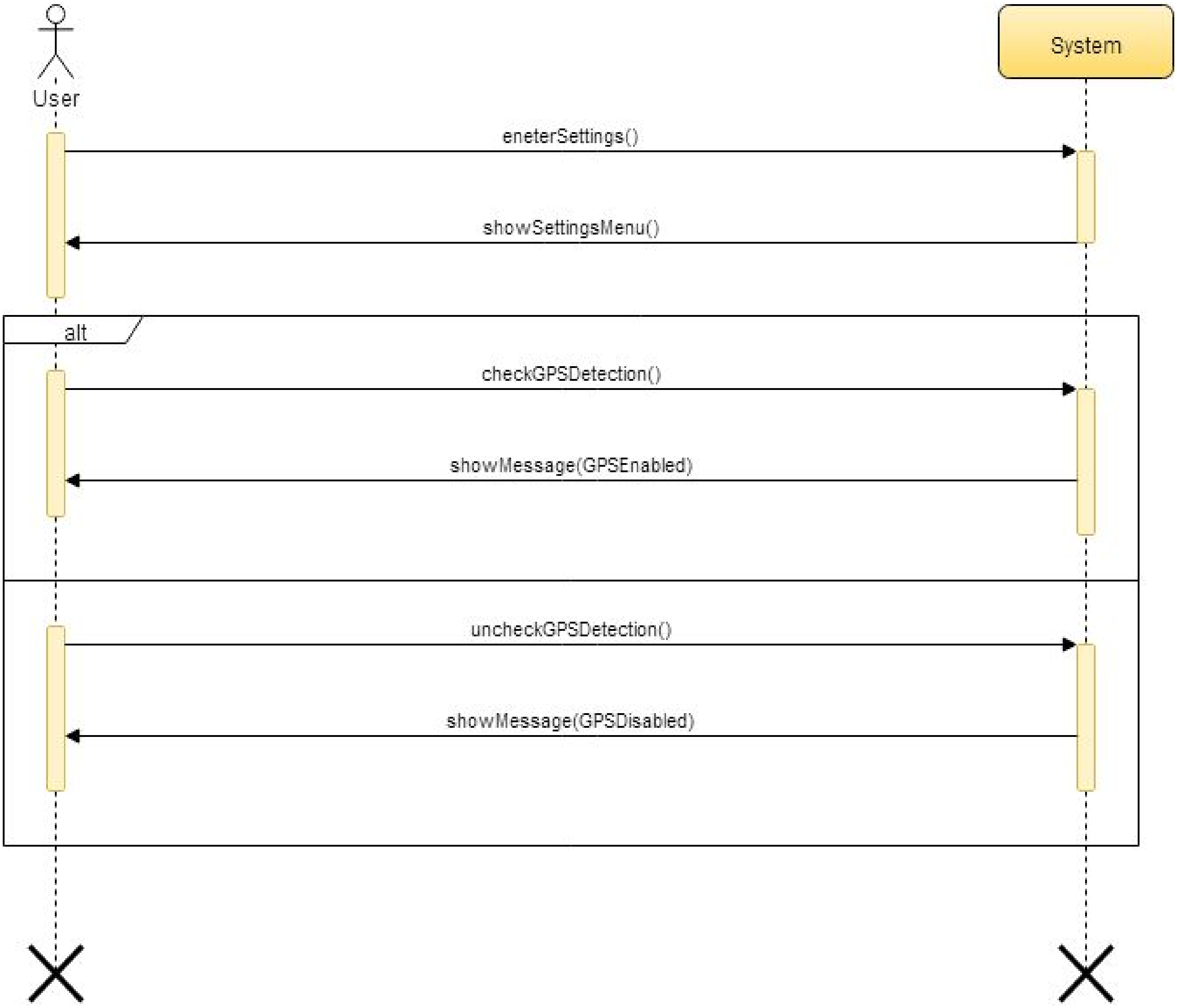
Long term reservation management

Actors	Registered user
Goal	G5
Input Condition	The user must be logged into the application and he/she must have made a long term reservation and there are more than two hours from the scheduled departure.
Flow of events	1. From his/her personal home page the user clicks on the button 'See pending reservations'. 2. The systems lists all the long term reservation he/she has made up until now (if there are any at all); 3. The user can click on one of the reservation to expand it , in order to see the details regarding it; 4. After expanding the selected reservation, he/she clicks on 'Delete reservation'; 5. The system validates if such action can be performed and deletes the reservation;
Output Condition	The user successfully deleted a long term reservation.
Exception	Exceptions may arise such as: ● the present time isn't two hours before the scheduled departure, therefore he/she cannot delete the reservation; In all exceptions presented above the system alerts the user through pop ups with the error message.


User
System
seePendingReservations()
listPendingReservations()
selectReservation(reservationX)
showDetails()
deleteReservation()
askForConfirmation()
confrimDelete()
eraseReservation()
showMessage(ReservationDeleted)
listPendingReservation()

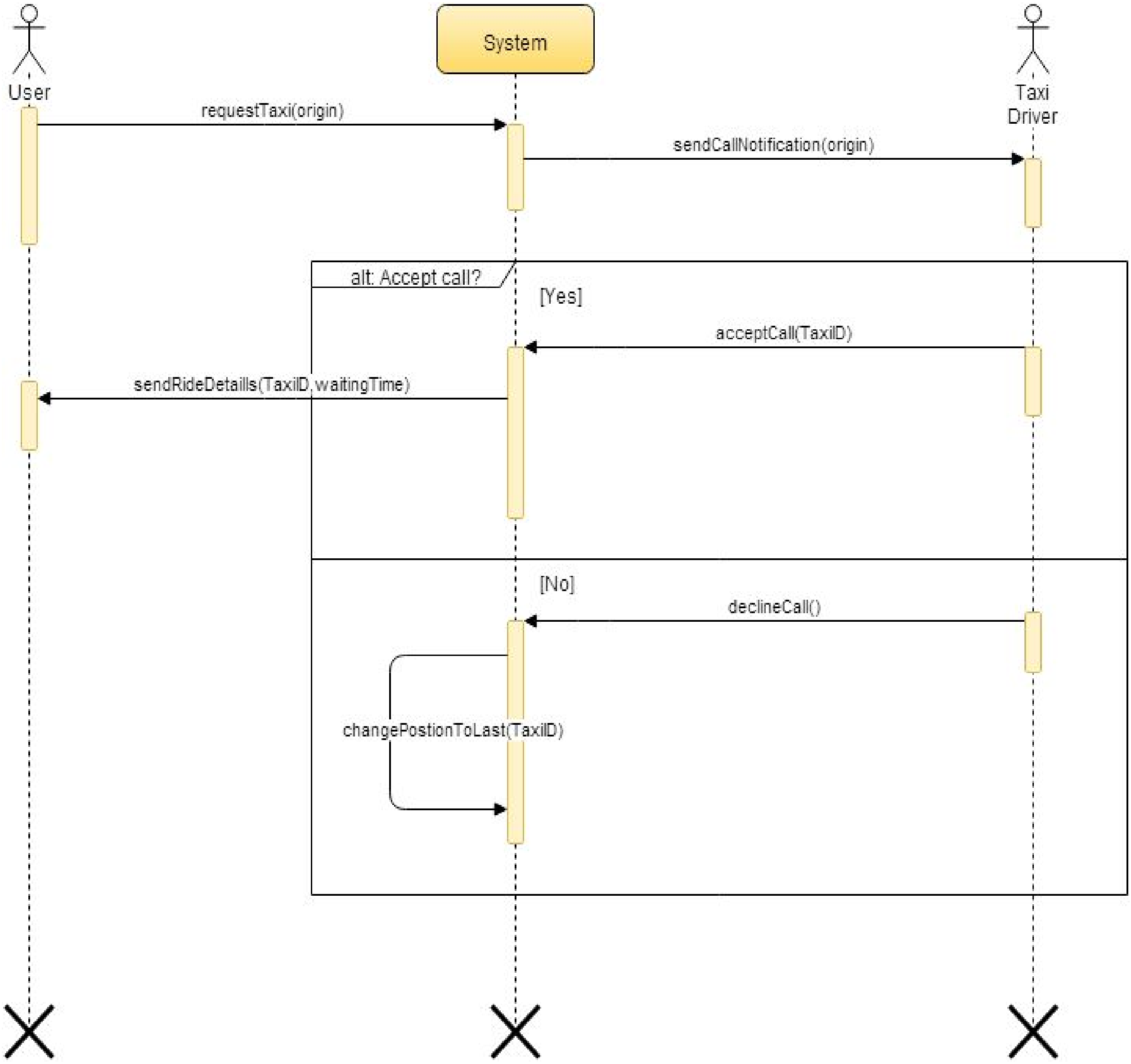
User location detection management

Actors	Registered user
Goal	G3.2
Input Condition	The user must be logged into the application.
Flow of events	1. The user form his/her personal homepage can click on the button 'Settings'. 2. The system presents the user with a new page showing the current choosed option to determine the departure location of a ride. 3. The user unchecks/checks 'Enable GPS detection'. 4. In each case the system asks for a confirmation from the user and then changes it.
Output Condition	Successfully changed the way of inputting the departure location of a ride.
Exception	Exceptions may arise such as: ● the privacy settings on the device do not permitted such action (this exception can arise only if the user wants to change from manual to GPS detection); ● the user's device does not have a GPS system; In all exceptions presented above the system alerts the user through pop ups with the error message.


User
System
eneterSettings()
showSettingsMenu()
alt
checkGPSDetection()
showMessage(GPSEnabled)
uncheckGPSDetection()
showMessage(GPSDisabled)

Notification of a call

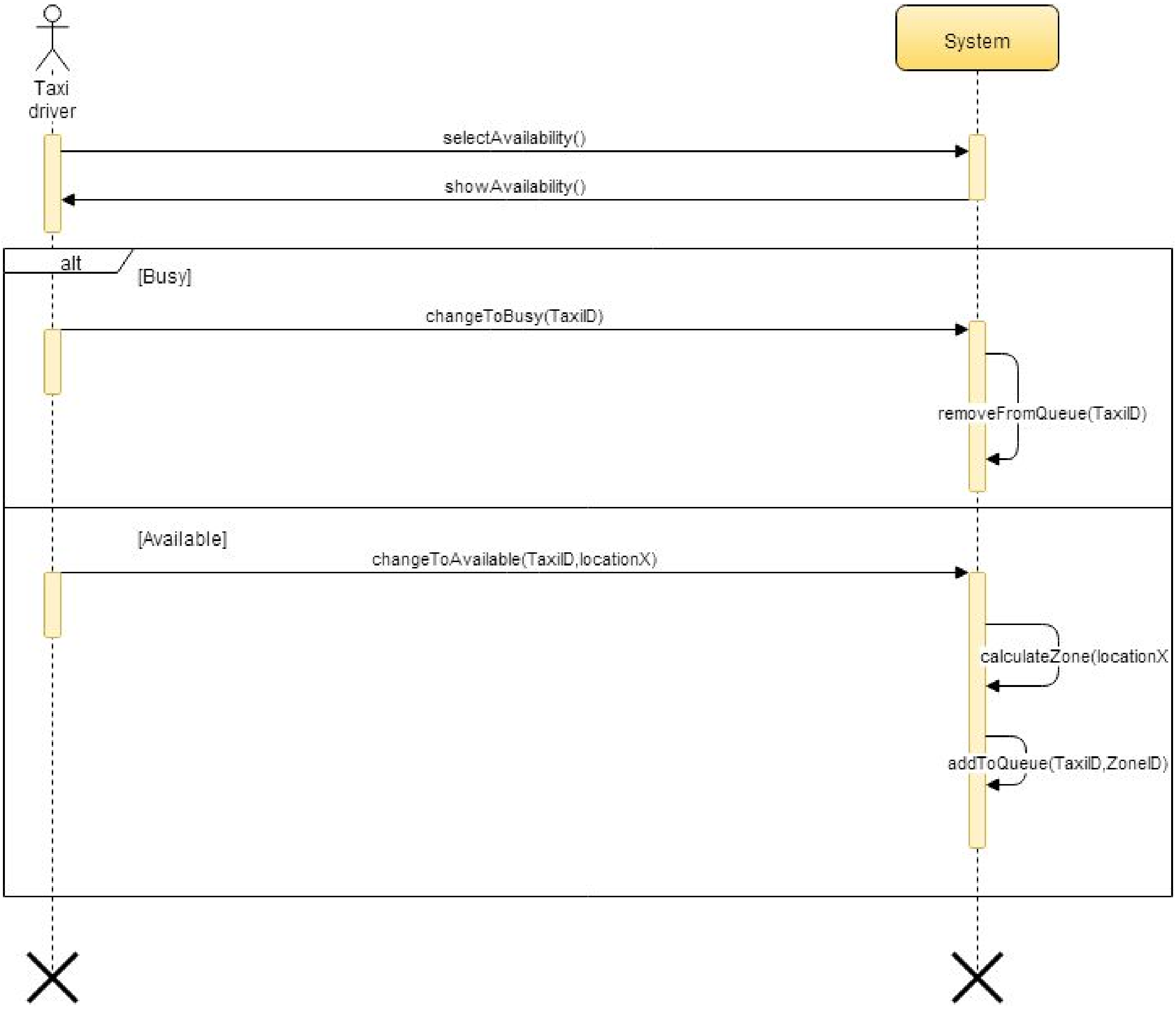
Actors	Taxi driver
Goal	G7, G8
Input Condition	The driver must be logged in the application.
Flow of events	1. The system notifies the driver of an incoming call showing him/her where the client is; 2. The system asks the driver to decide take or not the ride by choosing 'Yes' or 'No'; 3. The driver chooses one of the options; 4. The driver tapped on the 'Yes' button: 4.1. The pop disappears and on the screen is shown the location of where to pick up the client: 4.1.1. He uses the GPS navigator to get to where the client is; 5. The driver tapped on 'No' : 5.1. The driver can return to what he/she was previously doing;
Output Condition	The taxi driver was notified of a call and he/she was allowed to accept or decline it.
Exception	There are no possible exception that we know of.



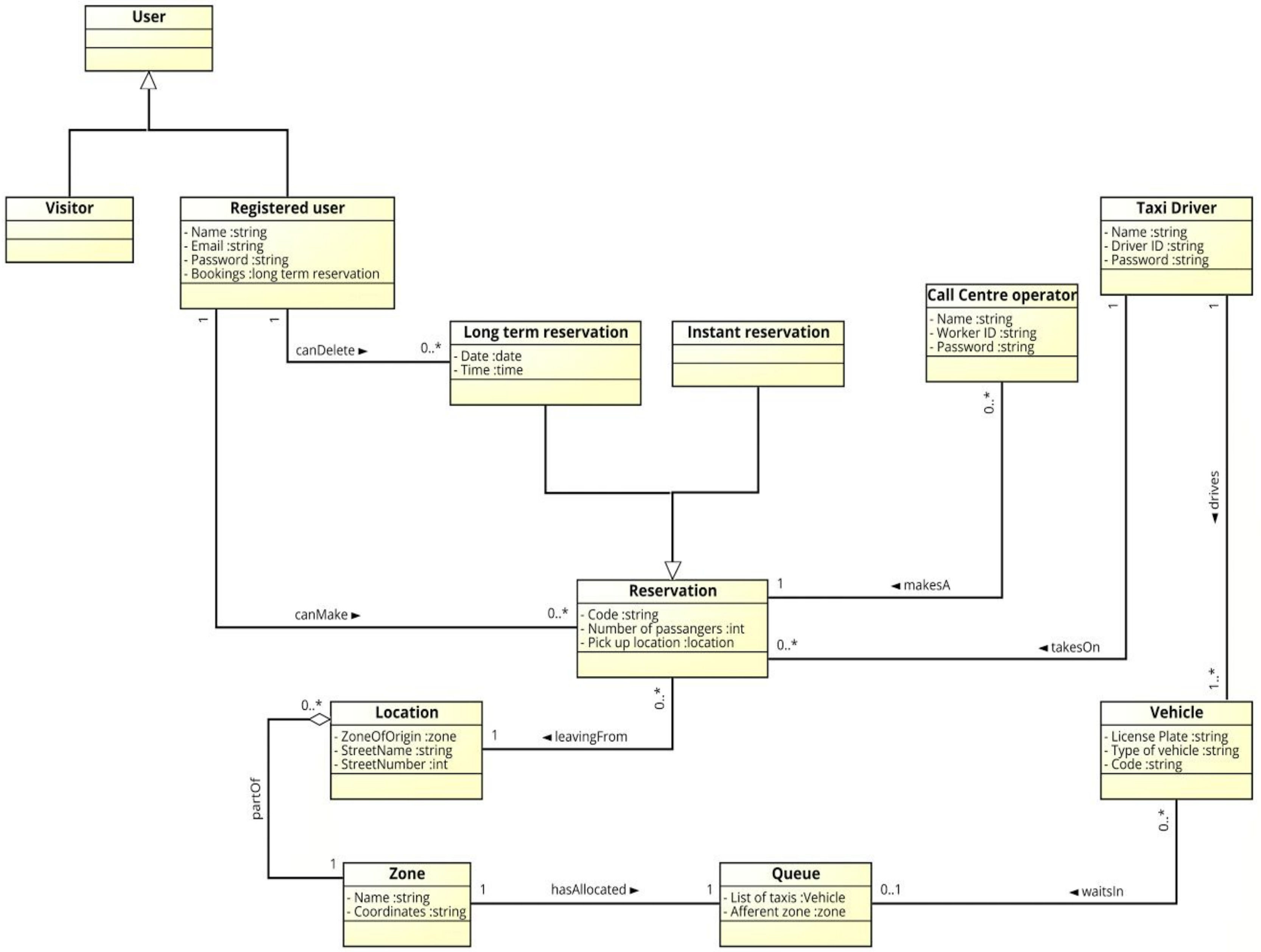
User
System
Taxi Driver
requestTaxi(origin)
sendCallNotification(origin)
alt: Accept call?
[Yes]
acceptCall(TaxiID)
sendRideDetaills(TaxiD,waitingTime)
[No]
declineCall()
changePostionToLast(TaxiD)

Taxi Availability

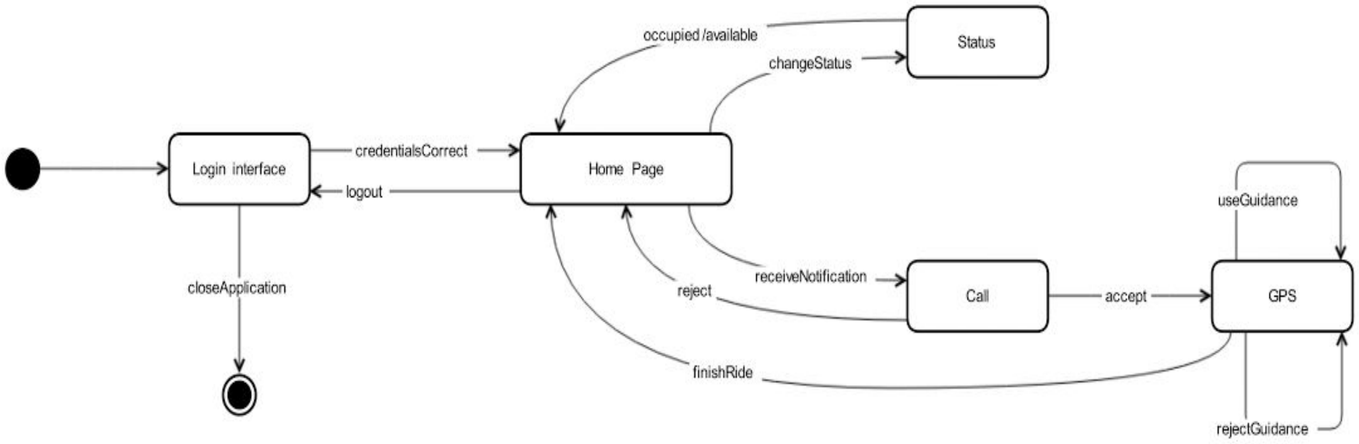
Actors	Taxi Driver
Goal	G6
Input Condition	The driver must logged into the application.
Flow of events	1. On his/her personal home page the driver taps on the slider ‘Select you availability’ accordingly; 2. The system acts accordingly to the chosen option: 2.1. The taxi became available, thus the system inserts it in the queue of the zone it is in; 2.2. The taxi became busy, thus the system deletes it from the queue it previously was inserted;
Output Condition	Successfully changed the availability of a vehicle, and updated the queues.
Exception	There are no possible exception that we know of.

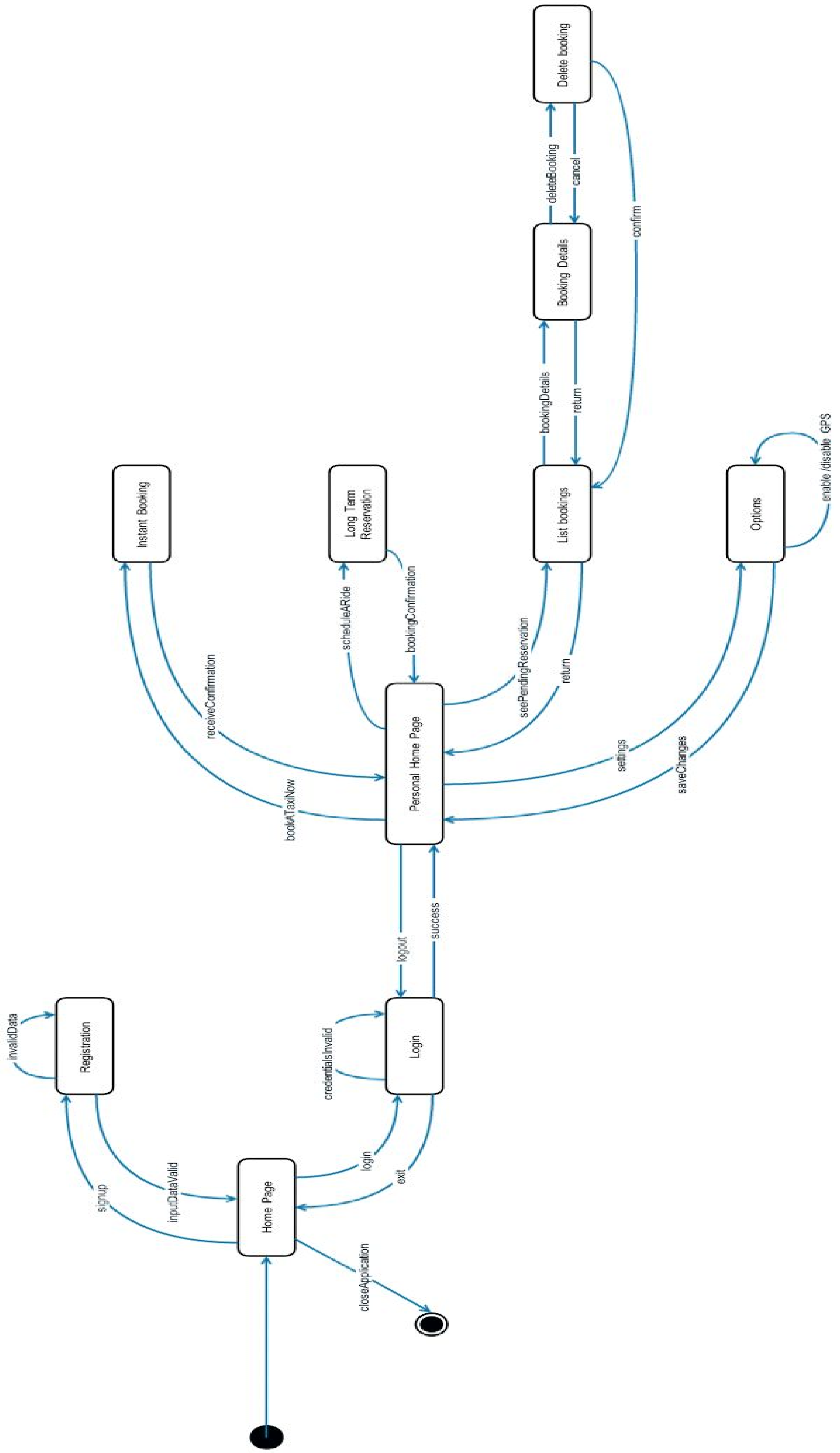


myTaxiService - Class Diagram



3.5.3 State machine diagrams




Home Page
signup
inputDataValid
Registration
invalidData
login
exit
Login
credentialsInvalid
closeApplication
logout
success
Personal Home Page
bookATaxiNow
receiveConfirmation
Instant Booking
scheduleARide
bookingConfirmation
Long Term Reservation
seePendingReservation
return
List bookings
bookingDetails
return
Booking Details
deleteBooking
cancel
Delete booking
confirm
settings
saveChanges
Options
enable /disable GPS

3.6 Non functional requirements

3.6.1 Performance Requirements

Performance must be acceptable to guarantee a good grade of usability. We assume the response time of the system is within 10 to 60 seconds and that the performance is essentially affected by user's internet connection. The performance of the system is also affected by number of users that will interact with the application, we therefore suppose that the system can cope with up to 1000 users connected at the same (where users include registered clients, taxi drivers, call centre operators) in order to offer a good service especially during events that the city may host.

3.6.2 Design Constraints

The computer application and the Android app will be developed using Java EE so it will inherit all language's constraints, while the IOS app will be developed using Swift.

3.6.3 Software System Attributes

3.6.3.1 Availability

The application will be accessible at any given time. To achieve this goal we will use the cloud hosting service offered by Google. This solution will significantly reduce costs for creating a dedicated server and maintaining it; it also offers a guarantee that the system will be able to offer its services on big loads.

3.6.3.2 Maintainability

The application does not provide any specific API,but the whole application code will be documented to inform future developers of how application works and how it has been developed.

3.6.4 Security

3.6.4.1 External Interface Side

myTaxiService offers its services only to registered users only, thus the system requires every visitor that wants to register to the system to create a strong password in order to secure his/her account. The password must be at least 8 characters long and can be up to 12 characters, it must contain high and lower cases letters, number and at least one special symbol. The passwords are encrypted with a hash algorithm and saved on the Google hosting service which also offers an additional degree of security of its own. In the future it could be wise to use a two-factor authentication: as to associate a phone number to the account, in order to be one hundred percent sure that the changes made to the account are made by owner and not by some third party.

While for the other two users, taxi drivers and call centre operators, they are given cards containing the password with which they can log into the system. Any problem they may have regarding their credentials they must go to the administrative office located in the call centre building.

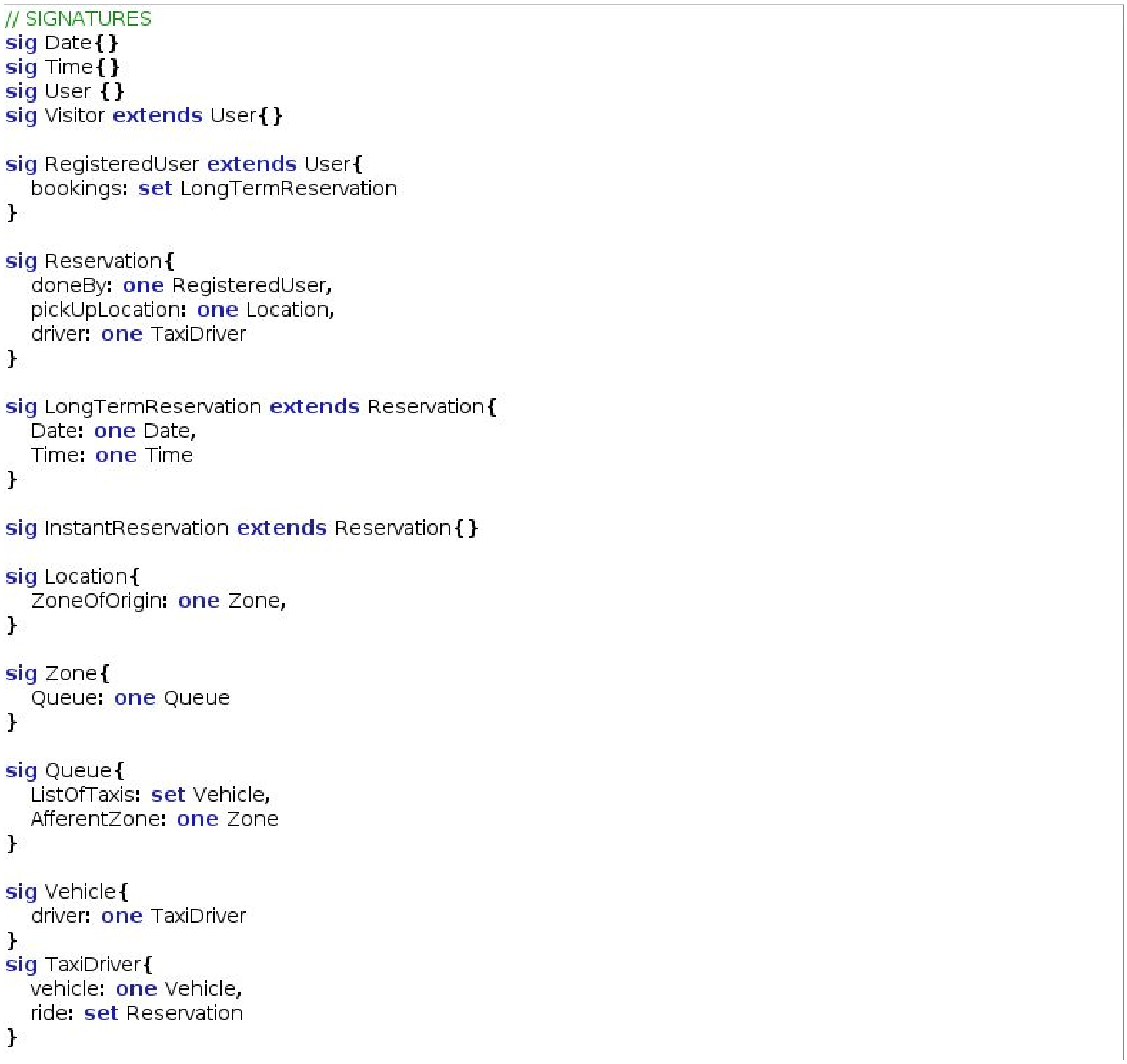
3.6.4.2 Application Side

On the application side will be implemented a filtering system in order to prevent malicious attacks (mainly SQL injection). Another important security measure is the implementation of a secure connection in order to guarantee communication confidentiality and integrity and also mutual authentication.

4 Appendix

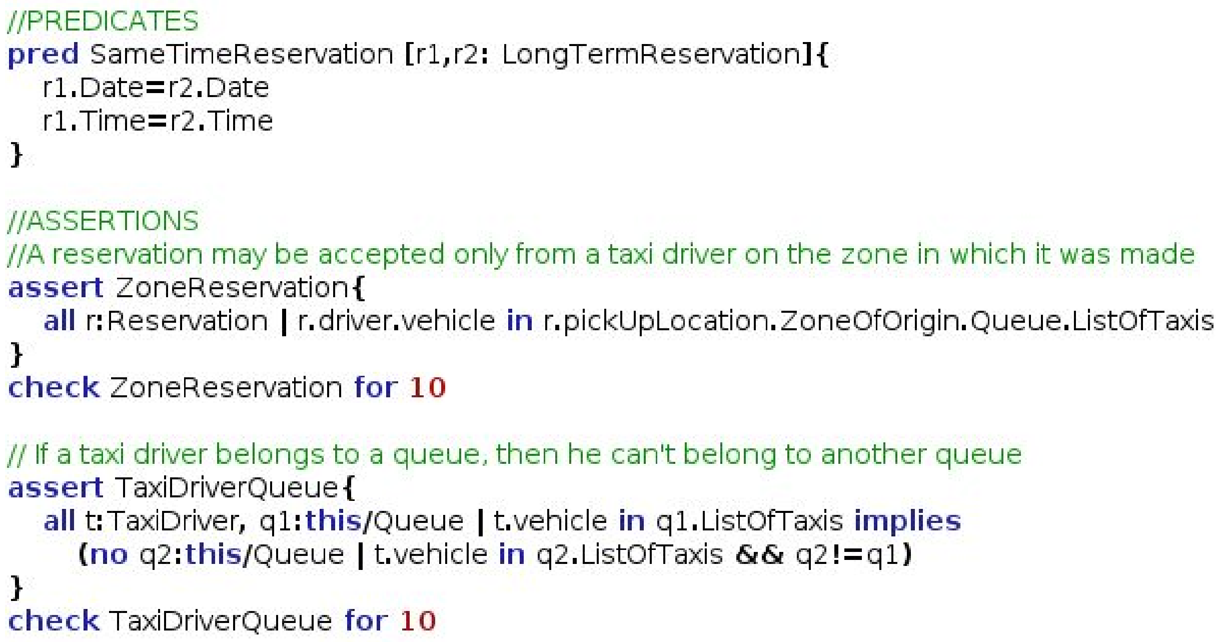
4.1 Alloy

In the following page there is the Alloy code used to check the consistency of the model produced. The model is quite similar to the one represented in the UML class diagram, however the call center operator are not included (as they are part of the legacy system). Then the output shows the result of the validation process. Eventually there is a graphic representation of the model (but for a small number of instances, for readability purposes.

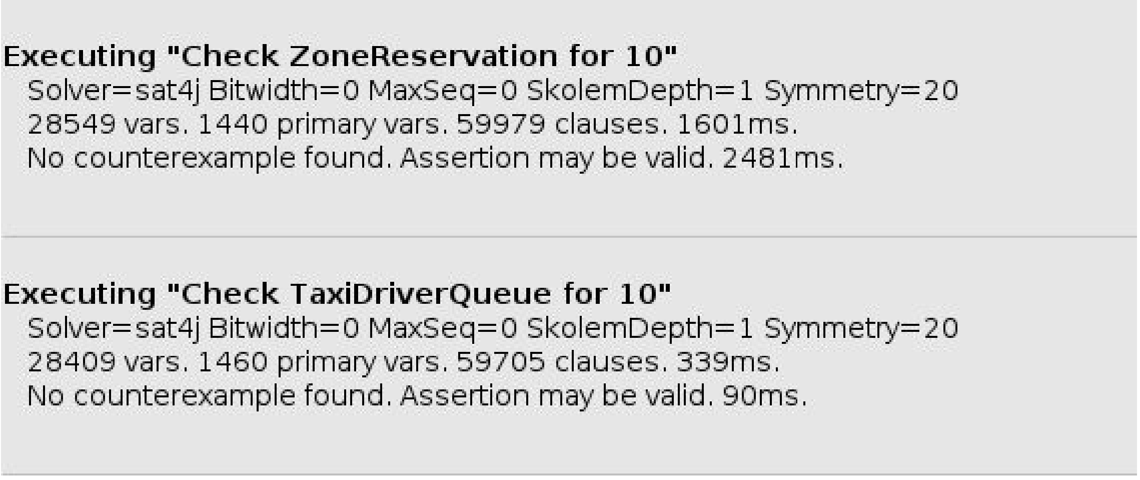


```
// SIGNATURES
sig Date{}
sig Time{}
sig User {}
sig Visitor extends User{}

sig RegisteredUser extends User{
   bookings: set LongTermReservation
}

sig Reservation{
   doneBy: one RegisteredUser,
   pickUpLocation: one Location,
   driver: one TaxiDriver
}

sig LongTermReservation extends Reservation{
   Date: one Date,
   Time: one Time
}

sig InstantReservation extends Reservation{}

sig Location{
   ZoneOfOrigin: one Zone,
}

sig Zone{
   Queue: one Queue
}

sig Queue{
   ListOfTaxis: set Vehicle,
   AfferentZone: one Zone
}

sig Vehicle{
   driver: one TaxiDriver
}
sig TaxiDriver{
   vehicle: one Vehicle,
   ride: set Reservation
}
```

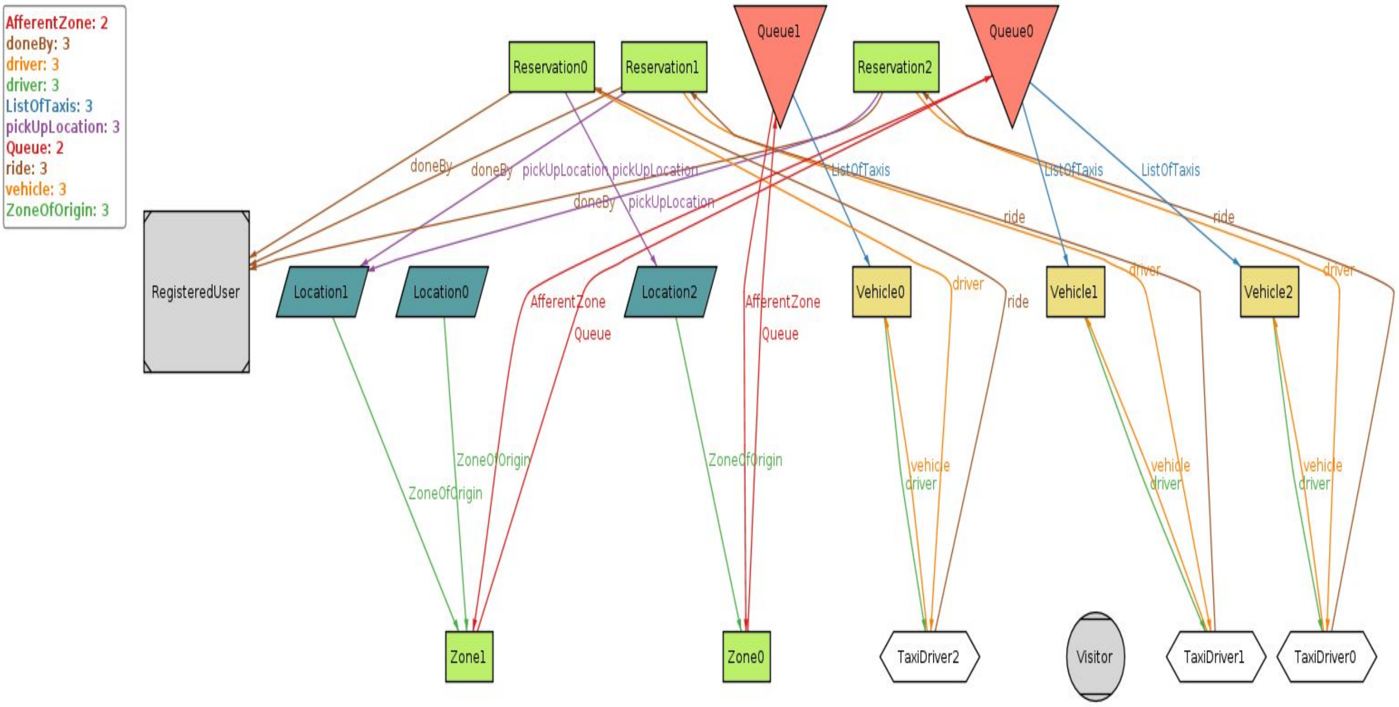
```
//FACTS
// Two taxi drivers can't own the same vehicle
fact VehicleHasSingleOwner{
   no t1,t2: TaxiDriver | !(t1= t2) && t1.vehicle=t2.vehicle
}

// A driver has exactly one vehicle
fact SymmetricTaxiDriverVehicle{
   all v: Vehicle, t:TaxiDriver |
   t in v.driver implies v in t.vehicle
}
// Symmetric relation between reservation and taxi driver: a reservation is taken by a taxi if and only if
// the taxi takes that reservation
fact SymmetricTaxiDriverReservation{
   all r: Reservation, t:TaxiDriver | (t=r.driver implies r in t.ride) &&
   (r in t.ride implies t=r.driver)
}
// Symmetric relation between Queue and AfferentZone
fact SymmetricZoneQueue{
   all q:this/Queue, z: Zone |
   (q.AfferentZone=z implies z.Queue=q) &&
   (z.Queue=q implies q.AfferentZone=z)
}

// The reservation with pick up location in a Zone are done by taxis whose taxi is in that Zone
fact {
   all r:Reservation, q:this/Queue | r.pickUpLocation.ZoneOfOrigin=q.AfferentZone implies
      (one t:TaxiDriver | r in t.ride && t.vehicle in q.ListOfTaxis)
}

// Vehicles belongs to exactly one queue
fact VehicleInSingleQueue{
   all q1,q2: this/Queue, v: Vehicle | v in q1.ListOfTaxis implies !(v in q2.ListOfTaxis && !(q2=q1))
}
// The same user can't make two reservations with the same time and date
fact NoDoubleLongTermReservation{
   all ltr1,ltr2: LongTermReservation | SameTimeReservation[ltr1][ltr2] implies
   ltr1.doneBy!=ltr2.doneBy
}
//The same taxi driver can't serve two customers whose booking are at the same time
fact NoUbiquity{
   no r1,r2:Reservation | (r1.driver=r2.driver && SameTimeReservation[r1][r2] && r1!=r2)
}
```



```
//PREDICATES
pred SameTimeReservation [r1,r2: LongTermReservation]{
   r1.Date=r2.Date
   r1.Time=r2.Time
}

//ASSERTIONS
//A reservation may be accepted only from a taxi driver on the zone in which it was made
assert ZoneReservation{
   all r:Reservation | r.driver.vehicle in r.pickUpLocation.ZoneOfOrigin.Queue.ListOfTaxis
}
check ZoneReservation for 10

// If a taxi driver belongs to a queue, then he can't belong to another queue
assert TaxiDriverQueue{
   all t:TaxiDriver, q1:this/Queue | t.vehicle in q1.ListOfTaxis implies
      (no q2:this/Queue | t.vehicle in q2.ListOfTaxis && q2!=q1)
}
check TaxiDriverQueue for 10
```



```
Executing "Check ZoneReservation for 10"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   28549 vars. 1440 primary vars. 59979 clauses. 1601ms.
   No counterexample found. Assertion may be valid. 2481ms.

Executing "Check TaxiDriverQueue for 10"
   Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
   28409 vars. 1460 primary vars. 59705 clauses. 339ms.
   No counterexample found. Assertion may be valid. 90ms.
```


AfferentZone: 2
doneBy: 3
driver: 3
driver: 3
ListOfTaxis: 3
pickUpLocation: 3
Queue: 2
ride: 3
vehicle: 3
ZoneOfOrigin: 3
Reservation0
Reservation1
Queue1
Reservation2
Queue0
RegisteredUser
Location1
Location0
Location2
Vehicle0
Vehicle1
Vehicle2
Zone1
Zone0
TaxiDriver2
Visitor
TaxiDriver1
TaxiDriver0
doneBy
doneBy
pickUpLocation
pickUpLocation
doneBy
pickUpLocation
ListOfTaxis
ListOfTaxis
ListOfTaxis
ride
ride
ride
driver
driver
driver
AfferentZone
AfferentZone
Queue
Queue
ZoneOfOrigin
ZoneOfOrigin
ZoneOfOrigin
vehicle
driver
vehicle
driver
vehicle
driver

5 Other info

5.1 Hours spent

- Adrian Mihai Berbieru: ~8 hours
- Attilio D'Onofrio: ~ 8 hours

5.2 Software used

- Google Docs (http://www.drive.google.com): to redact and to format this document.
- Signavio (http://academic.signavio.com/): to create Use Cases Diagrams and Class Diagrams;
- Draw.io (http://www.draw.io/): to create Sequence Diagrams and State Machine Diagrams;
- Alloy Analyzer (http://alloy.mit.edu/alloy/): to prove the consistency of our model.
- Moqups(https://moqups.com/): to create mockups.

6 Revision

This is 1.1 version of RASD that contains update of the document after the entair development of the application. In the follow are listed the difference between the previous version:

6.1 Added actors

- System administrator: a technician in charge of adding taxi drivers to DBMS via SQL language.

6.2 Added assumption

- The GPS signal won't be lost by taxis.
- Tables binding the address to city zone already exist.

6.3 Changed diagrams

- Added the correct sequence diagram at Section **3** "*Long term booking management*".