

POLITECNICO DI MILANO

COMPUTER SCIENCE AND ENGINEERING



SOFTWARE ENGINEERING 2



MyTaxiService

RASD Document

Authors



SARA PISANI – 854223

LEONARDO TURCHI – 853738

6/11/2015

MyTaxiService-RASD.pdf · V.0.8





Table of contents

1.	Introduction	8
1.1	Recipients.....	8
1.2	Purpose.....	8
1.2.1	Stakeholders.....	8
1.2.2	System Functionality	8
1.2.3	Goals.....	9
1.3	Definitions and acronyms	10
1.3.1	Definitions.....	10
1.3.2	Acronyms.....	10
1.4	References	11
1.5	Overview	11
2	General Description	12
2.1	Product perspective.....	12
2.2	User characteristics.....	12
2.3	Constraints.....	12
2.3.1	Regulatory policies	12
2.3.2	Hardware limitations	12
2.3.3	Interfaces to other applications	12
2.3.4	Parallel operation	12
2.3.5	Documents related	12
2.4	Assumptions and Dependencies.....	13





2.4.1 Assumption	13
2.5 Future possible implementation	14
3 Specific Requirements.....	15
3.1 External Interface Requirements.....	15
3.1.1 User Interfaces	15
3.1.1.1 Login	15
3.1.1.2 Registration form	16
3.1.1.3 Service selection	17
3.1.1.4 Ride Request	17
3.1.1.5 Ride Booking	18
3.1.1.6 Driver area.....	19
3.1.1.7 Popups.....	20
3.1.2 API Interfaces.....	21
3.1.3 Hardware Interfaces	22
3.1.4 Software Interfaces	22
3.1.5 Communication Interfaces.....	23
3.1.6 Memory	23
3.2 Functional Requirements.....	23
3.2.1 Common functionality (every goal need these):.....	23
3.2.1.1 [F1] Allow a visitor to become a registered user.	23
3.2.1.2 [F2] Allow user to log in to application.	24
3.2.2 [G1] The system allows an easy and fast taxi's immediate request.....	24
3.2.2.1 [F3] Allow user to request an immediate taxi ride.....	24





3.2.2.2 [F4] Allow user to delete an existing ride.....	24
3.2.2.3 [F5] Allow user to see their booking data.....	24
3.2.2.4 [F6] After login, application will notify passengers if their booking has been accepted or declined by the taxi driver	25
3.2.3 [G2] The system allows an easy taxi's reservation.	25
3.2.3.1 [F7] Allow user to request a taxi 'booking' ride.....	25
3.2.4 [G3] The system allows to share a ride.....	25
3.2.4.1 [F8] Allow user to share a ride.	26
3.2.5 [G4] The system allows drivers to be organized by a taxi queue.....	26
3.2.5.1 [F9] Allow user to see the state of their booking queue.....	26
3.2.5.2 [F10] Allow user to see data of every booking they had accepted.	26
3.2.5.3 [F11] After login, application will notify taxi drivers if there is a taxi booking for their taxi.....	26
3.3 Scenarios	28
3.3.1 Scenario 1 »	28
3.3.2 Scenario 2 »	28
3.3.3 Scenario 3 »	29
3.3.4 Scenario 4 »	29
3.3.5 Scenario 5 »	29
3.3.6 Scenario 6 »	30
3.4 UML Models.....	31
3.4.1 Use Case	31
3.4.1.1 User registration.....	31





3.4.1.2 User login.....	34
3.4.1.3 User requests for a taxi ride.....	35
1.1.1.1 User requests for a ‘booking’ taxi ride	38
3.4.1.4 Driver management of new rides.....	39
3.4.2 Class Diagram	42
3.4.3 Finite State Machine Diagrams	43
3.5 Non Functional Requirements	45
3.5.1 Performance Requirements	45
3.5.2 Design Constraints.....	45
3.5.3 Software System Attributes	45
3.5.3.1 Availability	45
3.5.3.2 Maintainability	47
3.5.3.3 Portability	47
3.5.4 Security	47
3.5.4.1 External Interface Side.....	47
3.5.4.2 Application Side	49
3.5.4.3 Server Side	50
4 Appendix	Errore. Il segnalibro non è definito.
4.1 Alloy.....	Errore. Il segnalibro non è definito.
4.1.1 Data Type	Errore. Il segnalibro non è definito.
4.1.2 Abstract Entity	Errore. Il segnalibro non è definito.
4.1.3 Abstrac Entity Implementation and Signature.....	Errore. Il segnalibro non è definito.
4.1.4 Fact	Errore. Il segnalibro non è definito.





- 4.1.5 Assert Errore. Il segnalibro non è definito.
- 4.1.6 Predicates Errore. Il segnalibro non è definito.
- 4.1.7 Result Errore. Il segnalibro non è definito.
- 4.1.8 Generated world..... Errore. Il segnalibro non è definito.
- 4.2 Software and tool used..... Errore. Il segnalibro non è definito.
- 4.3 Hours of works..... Errore. Il segnalibro non è definito.
- 5 Revision Errore. Il segnalibro non è definito.
 - 5.1 Changed Assumptions..... Errore. Il segnalibro non è definito.
 - 5.2 Removed goal and functionality Errore. Il segnalibro non è definito.
 - 5.3 Modified Functional Requirements Errore. Il segnalibro non è definito.
 - 5.4 Modified Scenarios and Use Cases..... Errore. Il segnalibro non è definito.
 - 5.5 Modified Diagrams..... Errore. Il segnalibro non è definito.





1. Introduction

1.1 Recipients

This document is applied to professors who teach software engineering 2 in the years 2015/2016, but other users interested in this documents could be the developers of a system that manage requests of taxi rides.

1.2 Purpose

The purpose of this document is the description of a system that offer a taxi service. The software developed will be a web application that allow user to book a taxi ride, immediately or starting from two hours after, and offers a taxi sharing service.

1.2.1 Stakeholders

- **Passengers:** they can book a taxi ride by inserting a departure and a destination, with the chance to share the ride with other passenger to save money.
- **Drivers:** they make available their taxi on-line and they can accept or decline ride requests from passengers.

1.2.2 System Functionality

- Ride's customization





Passengers can personalize a ride simply choosing a button: "Immediate Booking" or "Delayed Booking". Once the choice has been taken, passengers have to fill out a form with two required fields (departure and destination) for the Immediate booking, and three required fields (departure, destination and time) for the Delayed booking. The field "time" allows to book a taxi after a couple of hours or more.

Once completed the form, the button "Taxi Sharing" makes the reservation provided for sharing and shows the other passengers who are interested in the same ride.

- Booking management

The system creates a priority queue of available taxi in each zone, and the current request is sent to the taxi on top of the queue.

If he accepts, the system enqueues the driver in back of the queue and saves the ride's data.

If he declines, the system enqueues the driver in back of the queue and iteratively sends the request to the first taxi in the queue.

- Response to reservation

Drivers on top of the queue receive a popup notification about the ride, with complete details.

If he accepts the passenger receives a confirmation popup notification.

1.2.3 Goals

- The system allows an easy and fast taxi's immediate reservation.

The research is optimized by geographic zone.





- The system allows an easy and fast taxi's delayed reservation.
The research is optimized by geographic zone.
- The system allows to share a ride: passengers could take advantage from this service by saving money and, with a special consideration for the environment, This service reduce the number of automobiles on the road and the traffic.
- The system allows drivers to be organized by a taxi queue: they save time because their rides are optimized by geographic zone.

1.3 Definitions and acronyms

1.3.1 Definitions

- **System:** The software that will be developed.
- **Service:** Everything That will be make available to the users by the system.
- **Ride:** The main service, that concerns passenger's mobility.

1.3.2 Acronyms

- DBMS: Database Management System
- J2EE: Java 2 Enterprise Edition
- JVM: Java Virtual Machine





1.4 References

- Assignments 1 and 2.pdf
- Assignments rules and group registration.pdf
- IEEE standard for requirement specification.pdf
- Alloy reference manual.pdf

1.5 Overview

The paper is organized as follows:

- **Chapter 1:** This section explains the target audience, the purpose and the references of the Project.
- **Chapter 2:** This section provides a general description of the system, its features and its requirements.
- **Chapter 3:** This section contains all the necessary software requirements to allows a proper implementation.





2 General Description

2.1 Product perspective

The application that will be released is a web application which is not integrated with other existing system. That application will not have any internal interface for administration but it will be only user based. The application will not provide any interface or API for integration with future project.

2.2 User characteristics

The user that we expect to use our application is a person who want an easy way to travel in their city. This user must be able to use a web browser and have access to internet.

2.3 Constraints

2.3.1 Regulatory policies

MyTaxiService doesn't have to meet any regulatory policies.

2.3.2 Hardware limitations

MyTaxiService doesn't have to meet any hardware limitations.

2.3.3 Interfaces to other applications

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

2.3.4 Parallel operation

MyTaxiService must support parallel operations from different users when working with database and with all operation done by the user after connection.

2.3.5 Documents related

- Requirements and Analysis Specification Document (RASD)
- Design Document (DD)





- Testing report
- User's Manual ? serve??

2.4 Assumptions and Dependencies

2.4.1 Assumption

- There isn't an administrator or privileged user. Is not necessary a hierarchy of users to keep the system safe.
- Using the simple taxi service there isn't any dependence between users, but using the taxi sharing service there is.
- Ride request's confirm popup notification will be sent only if there's at least an available taxi, otherwise a message or a notice will inform the user.
- If a passenger deletes a taxi reservation (made at least two hours before), he must do this within 30 minutes before and the taxi will be set free by the system.
The driver receive a popup notification.
- If a passenger deletes a taxi reservation (made instantly), he must do this within 5 minutes and the taxi will be set free by the system.
The driver receive a popup notification.
- If a passenger deletes any type of taxi reservation for which has been confirmed the sharing option, his seats will be set free by the system and the other passenger and the driver receive a popup notification.





- The choice to set a ride shareable will not be possible to modify later.
The passenger must delete the reservation and create a new one.
- Assumption about visibility will be described by the following table:

2.5 Future possible implementation

- **Increase of customizing filters:**

MyTaxiService wants to provide more possibility of personalization for the sharing service, so could be inserted more fields to choose the characteristics of the traveling companion (for example smoker or not, chatterbox or not...).

- **Chance to send a feedback:**

from passengers to the driver and vice versa and from passengers to other passengers who share the same taxi.

- **Provide a calendar to the drivers:**

with the handy data of every ride that they have confirmed, the organization could be better.

- **Reduce cheating:**

providing passengers the route and the estimated time of their ride, inefficiency caused by drivers would be reduced.





3 Specific Requirements

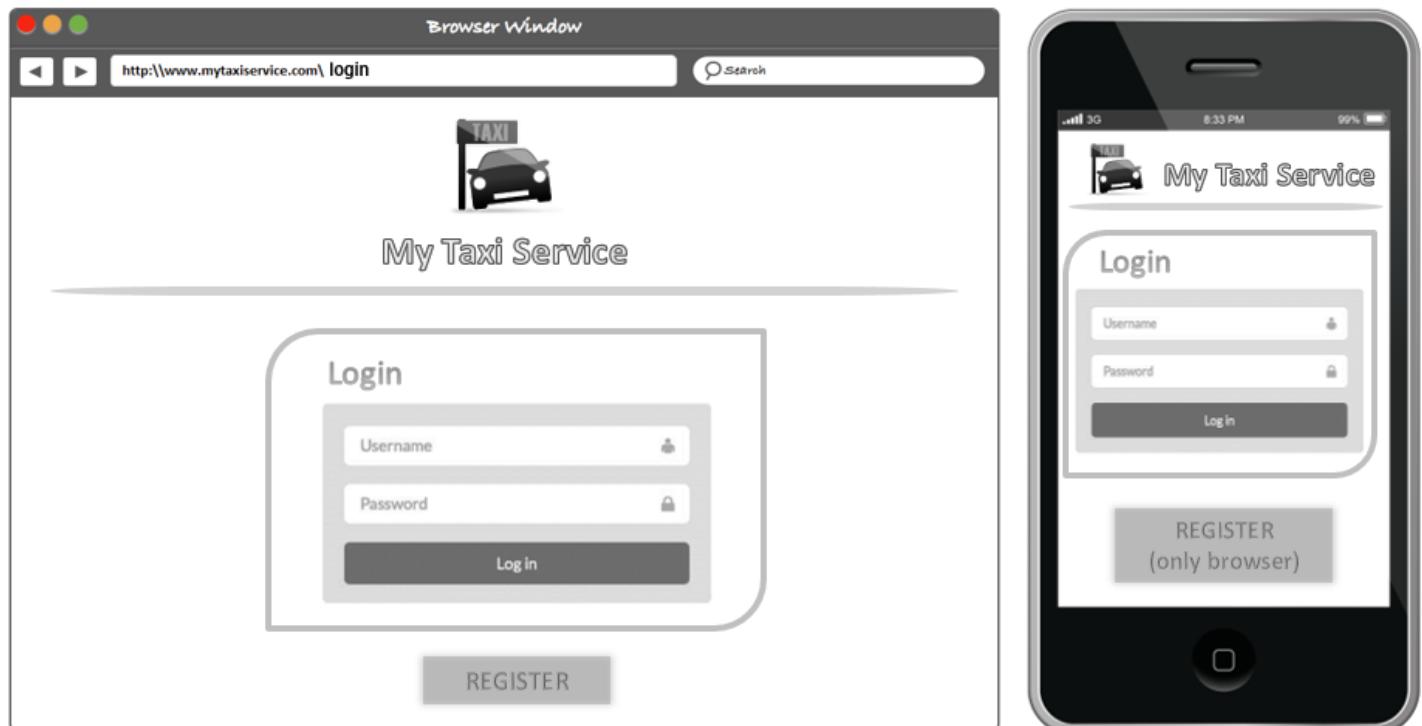
3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 Login

Here is a template of the home page of MyTaxiService.

Users can log in to the site or access the registration page.





3.1.1.2 Registration form

The first registration to the service must be done via browser.

A generic user becomes a registered user after the submission of a form.

Browser Window

http://www.mytaxiservice.com/registration

Search

TAXI

My Taxi Service

Registration

Email	Birth
Password	Name
Retype Password	Surname
Paypal Account	CF / ID
	Address

CONFIRM





3.1.1.3 Service selection

This mockup shows the main services menu of the site/app.

The user has to click to enter in the requested area.



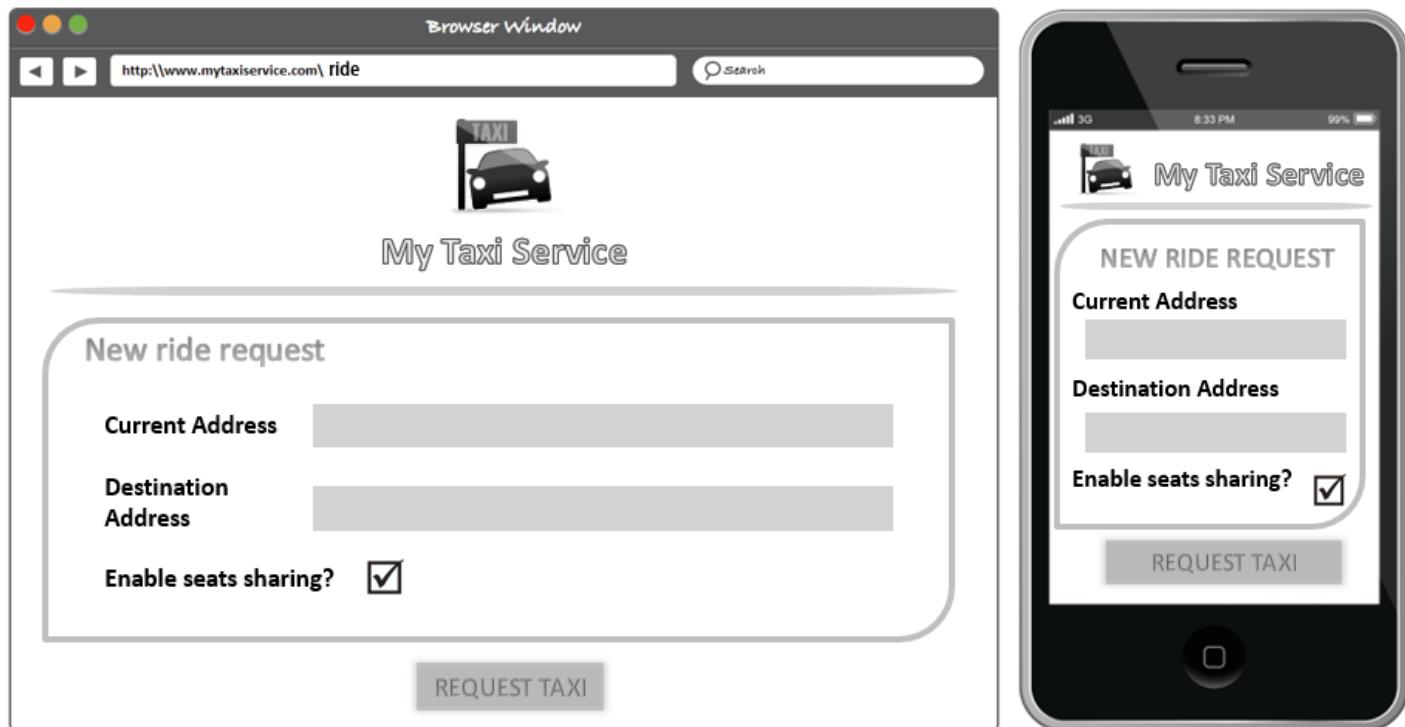
3.1.1.4 Ride Request

This mockup shows the primary page for a taxi request.

In this case is for a single ride without booking.

In addition the user can choose if other seats are shared with other people or not.





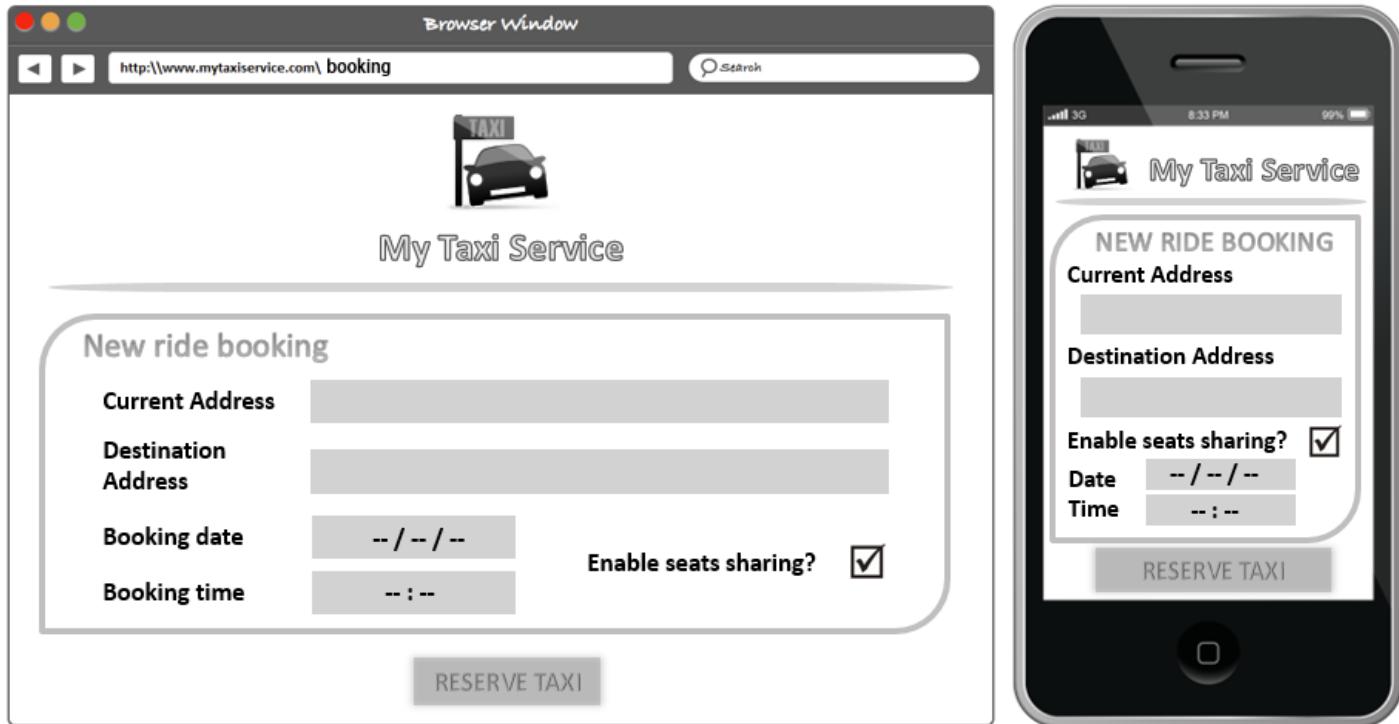
3.1.1.5 Ride Booking

This mockup shows the primary page for a taxi booking request.

This version includes the possibility of a taxi ride in a specific date and hour.

In addition the user can choose if other seats are shared with other people or not.





3.1.1.6 Driver area

Only for drivers, in the app there is a dedicated area.

In this section drivers are able to see the queue in theirs zone and accept or decline new rides.

With a right-slide are also able to manage the reservations: a list of booking rides is shown, drivers can accept or hide reservations.

This section is designed to be used in mobility, so there is no browser version.



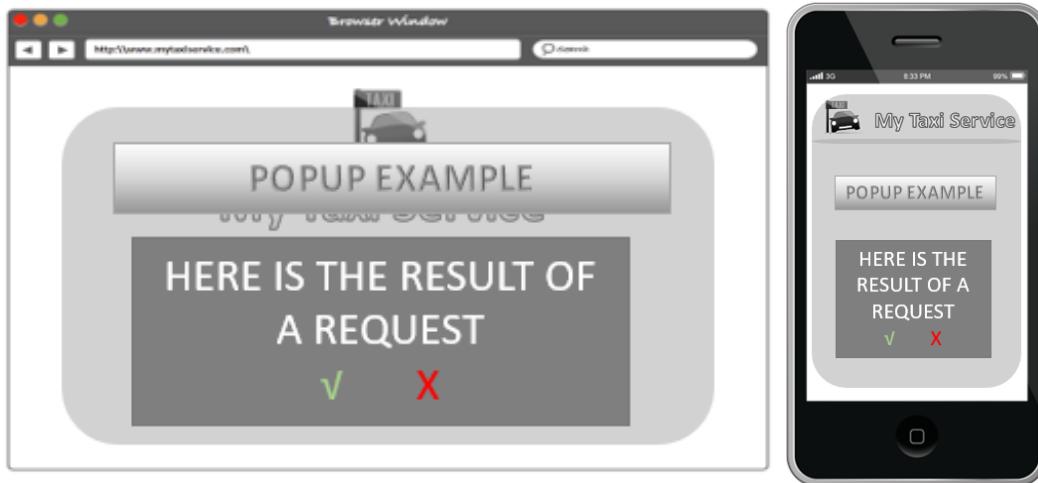


3.1.1.7 Popups

This mock up shows examples of pop-up alert.

Popups are used to inform or alert the user of something.





3.1.2 API Interfaces

To convert addresses to GPS coordinates we use the google Maps Geocoding API. As well described on the website, Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map. Also Data is available in JSON and XML format.

A future implementation could be to make available to the passenger the smartest route:

It will be possible to use Google Maps Roads API.

The Google Maps Roads API allows you to map GPS coordinates to the geometry of the road, and to determine the speed limit along those road segments. The API is available via a simple HTTPS interface, and exposes two services:

- **Snap to roads:** This service returns the best-fit road geometry for a given set of GPS coordinates. This service takes up to 100 GPS points collected along a route, and returns a similar set of data with the points snapped to the most likely





roads the vehicle was traveling along. Optionally, you can request that the points be interpolated, resulting in a path that smoothly follows the geometry of the road.

- **Speed limits:** This service returns the posted speed limit for a road segment. The Speed Limit service is only available to Google Maps API for Work customers. If you are an existing customer, you can contact your account manager or file a ticket in the Maps API for Work support portal to enable the Google Maps Roads API.

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: 5.6.21
 - Source: <http://www.mysql.it/>
- Java Virtual Machine (JVM).
 - Name: JEE
 - Version: 7
 - Source:
<http://www.oracle.com/technetwork/java/javaee/tech/index.html>
- Application server:
 - Name: Glassfish.
 - Version: 4.1.
 - Source: <https://glassfish.java.net/>





- Operating System (OS).
 - Application must be able to run on any SO which supports JVM and DBMS specified before.

3.1.5 Communication Interfaces

Protocol	Application	Port
TCP	HTTPS	443
TCP	HTTP	80
TCP	DBMS	3306 (default)

3.1.6 Memory

The minimum memory requirements are:

- Primary Memory: 2GB+
- Secondary Memory: 32GB+

3.2 Functional Requirements

3.2.1 Common functionality (every goal need these):

3.2.1.1 [F1] Allow a visitor to become a registered user.

- [R1] Visitor must not be already registered to perform registration process.
- [R2] Visitor must choose a username not already used by another user.
- [R3] User cannot sign up twice but only once for session.
- [R4] Visitor can just see login page.
- [R5] Visitor can only access to registration form.
- [D1] Email address used for registration must be formally correct.





3.2.1.2 [F2] Allow user to log in to application.

- [R1] User must be already registered to success login process.
- [R2] User must know his username and password used during registration to success login.
- [R3] Username and password insert during login process must be correct.
- [R4] Wrong credentials will not grant access to user to other pages.
- [R5] Visitor can't access to booking page before registration.
- [R6] Application will not implement retrieve password mechanism.

3.2.2 [G1] The system allows an easy and fast taxi's immediate request.

The research is optimized by geographic zone.

3.2.2.1 [F3] Allow user to request an immediate taxi ride.

- [R1] User must be already registered and logged in the application.
- [R2] User must complete booking form and confirm the ride.

3.2.2.2 [F4] Allow user to delete an existing ride.

- [R1] User must be already registered and logged in the application.
- [R2] User must be the owner of the ride he wants to delete.
- [R3] User must confirm deleting process.
- [R4] Deleting process is not reversible, all event data will be lost.

3.2.2.3 [F5] Allow user to see their booking data.

- [R1] User must be already registered and logged in the application.





- [R2] Booking must exist, owner of the booking has correctly created it.

3.2.2.4 [F6] After login, application will notify passengers if their booking has been accepted or declined by the taxi driver

- [R1] User must be already registered and logged in the application.
- [R2] Booking must exist, owner of the booking has correctly created it.
- [R3] Application will notify passengers of the taxi only when they perform login.

3.2.3 [G2] The system allows an easy taxi's reservation.

The research is optimized by geographic zone.

NOTE:

This goal has the functionality [F4] [F5] [F6] of [G1] plus the next:

3.2.3.1 [F7] Allow user to request a taxi 'booking' ride.

- [R1] User must be already registered and logged in the application.
- [R2] User must complete booking form and confirm the ride.
- [D1] Time, for a delayed booking, must be included between 00.00 and 23.59.

3.2.4 [G3] The system allows to share a ride.

Passengers could take advantage from this service by saving money and, with a special consideration for the environment, this service reduce the number of automobiles on the road and the traffic.





3.2.4.1 [F8] Allow user to share a ride.

- [R1] User must be already registered and logged in the application.
- [R2] User must complete booking form and confirm the ride.
- [R3] User must press the button "Sharing" to divide the cost of the ride and to reduce the number of cars on the road.

3.2.5 [G4] The system allows drivers to be organized by a taxi queue.

They save time because their rides are optimized by geographic zone.

3.2.5.1 [F9] Allow user to see the state of their booking queue.

- [R1] User must be already registered and logged in the application.
- [R2] Bookings must exist, owner of the booking has correctly created it.

3.2.5.2 [F10] Allow user to see data of every booking they had accepted.

- [R1] User must be already registered and logged in the application.
- [R2] Bookings must exist, owners of every single booking has correctly created it.

3.2.5.3 [F11] After login, application will notify taxi drivers if there is a taxi booking for their taxi

- [R1] User must be already registered and logged in the application.
- [R2] Booking must exist, owner of the booking has correctly created it.
- [R3] Application will notify taxi drivers only when they perform login.







3.3 Scenarios

3.3.1 Scenario 1 »

Bianca has finally finished her afternoon shift and she needs a taxi because it's very late for a dinner with some friends.

The restaurant is across town and she wants to save money, because she's a very stingy genovese girl: she decides to enable the sharing service.

Celeste, a Bianca's childhood friend from Genova, lives a block away from Bianca's office and she's late for the dinner too!

Celeste decides to use her new app MyTaxiService, and she finds out the option that allows passengers to share a ride: she could save a lot of money!

Celeste and Bianca have the same destination, so Mr. Rossi with his taxi picks up Bianca and, a few minutes later, Celeste.

The ligurian girls arrive on time for the dinner spending less!

3.3.2 Scenario 2 »

Rossella has a job meeting in the company headquarters this afternoon at 2 p.m. o'clock, but has a lunch meeting at 12 a.m.

The restaurant is in the Italian district, but the office that she must reach two hours later is in the financial district: they're too far apart!

The best solution to be punctual is to book a taxi, so she does it while is eating her breakfast.

At 11:30 a.m. she receives an e-mail in which it is written that the meeting has been moved to 15 because her boss has another obligation.

She booked the taxi for 1:45 p.m., so she is in time to delete the reservation, run some errands and make another request for the 14:45 p.m..





3.3.3 Scenario 3 »

Mr. Verdi has just parked his taxi after a stressful ride: the traffic was heavy and the customer was very unlikable, so he decide to take a break with a cappuccino and a croissant.

Soon as he bite the croissant a pop up warns him that he must pick up a passenger near there.

He's too nervous and needs at least 10 minutes to cool off, so he selects the option to decline the ride and continues to drink his cappuccino.

3.3.4 Scenario 4 »

Viola and her colleague Rosa have had a little chat on lunch break.

Rosa was very satisfied because she tried MyTaxiService and it has been a very efficient service, so invites Viola to use the same service.

Viola is not very good with technology, but Rosa said that MyTaxiService is simple to use and the registration is a fast process, so she decide to try.

The form that she must compile is intuitive and the fields require information easy to find: her name, her surname, her age, her address, her email, a password, her fiscal code and her PayPal account. Initially she has inserted a not valid e-mail address, so a pop up warns Viola about the error, but after her correction the registration has been a success!

3.3.5 Scenario 5 »

Verdiana must reach her boyfriend's house after dinner. He has moved in her same city just two days ago, but his apartment is not near the metro station and Verdiana is afraid of being out alone in the night. The fastest and safest way to arrive is a taxi, so Verdiana compiles the MyTaxiService's form with her start





address and her destination. At this point a pop up warns her about an nonexistent destination address: she has inserts the old address of her boyfriend! She corrects the mistake and is able to book the taxi.

3.3.6 Scenario 6 »

Mr. Bianchi and Mr. Neri must meet themselves for a job lunch. Mr. Bianchi is at home, so use his PC to book a taxi with MyTaxiService: he inserts his address and the destination's address and books the ride. Mr. Neri is in the city center, so use his mobile to book the ride: MyTaxiService's system suggests him the start address because it can find his position with the GPS data. He's very satisfied because he hates to write with the touchscreen keyboard and a simple touch has been sufficient to indicate his position.





3.4 UML Models

3.4.1 Use Case

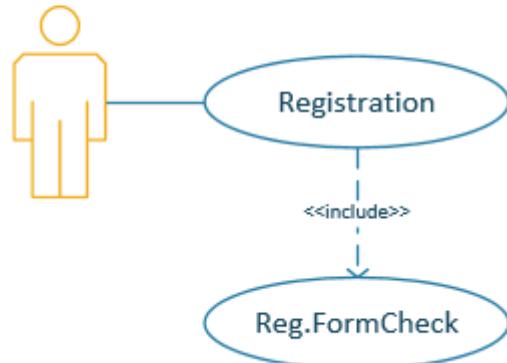
3.4.1.1 User registration

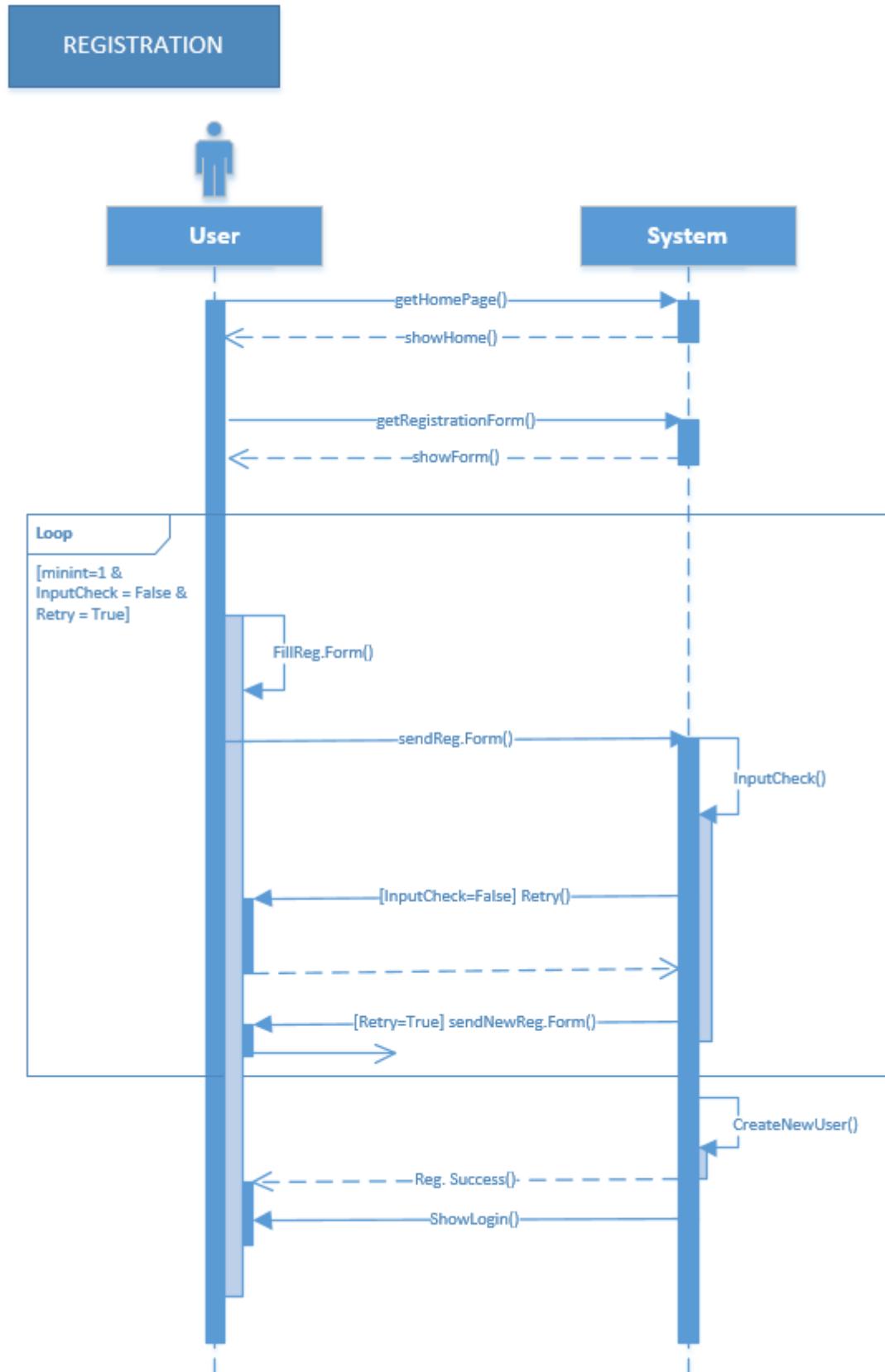
Actor	Visitor
Functionality	[F1]
Input Condition	NULL
Event Flow	<ol style="list-style-type: none">1. Visitor on the login page clicks on “register” button to reach the registration form.2. Visitor fills in all mandatory fields.3. Visitor clicks on “confirm” button.4. The application will save the credentials in the DB.5. A popup inform the user for the registration output.
Output Condition	Visitor successfully ends registration process and become a User. Now he/she is redirected to the login page and can log in to the application using his/her credential.
Exception	<ol style="list-style-type: none">1. The visitor is already a user.2. One or more mandatory fields are not valid.3. Email chosen is already associated to another user. <p>For every notification or exception a popup is shown over other elements of the page or app.</p>





REGISTRATION

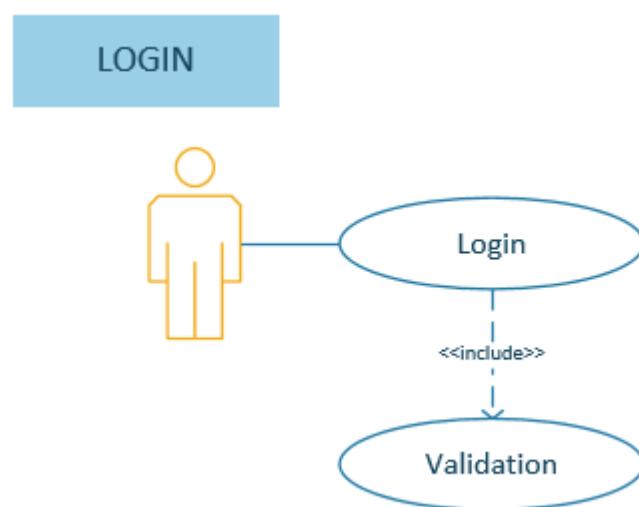


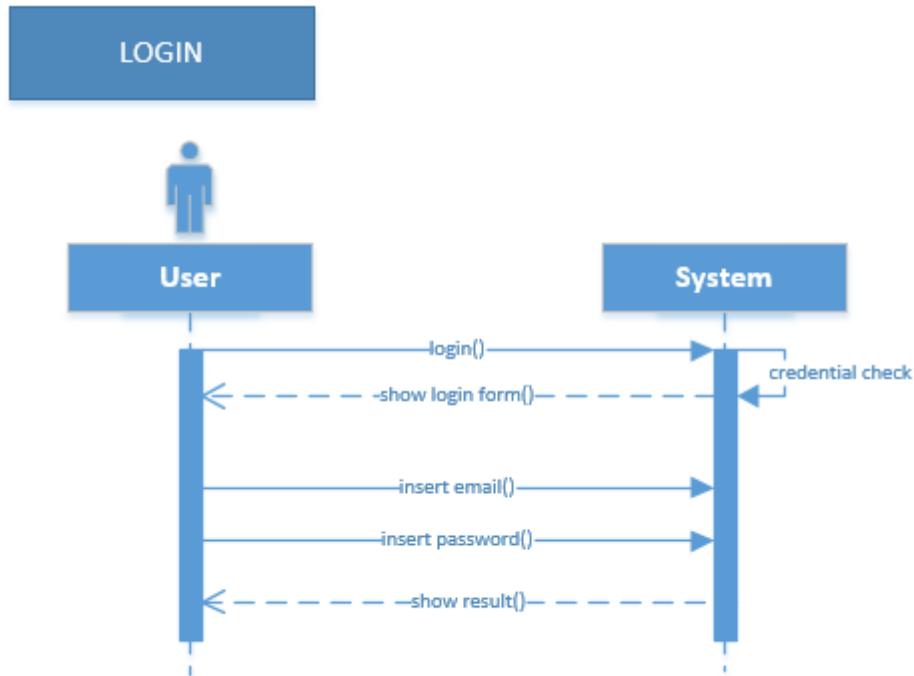




3.4.1.2 User login

Actor	Visitor, Registered User
Functionality	[F2]
Input Condition	Visitor is registered into the system.
Event Flow	<ol style="list-style-type: none">1. The site/app shows the login page.2. Visitor complete the form inserting a correct email and password.
Output Condition	The site/app verifies the credential of visitor and if these are correct, shows a menu with all services.
Exception	If email and/or password are incorrect the system notifies the visitor with a popup and show a redirect button the login page (to try again).





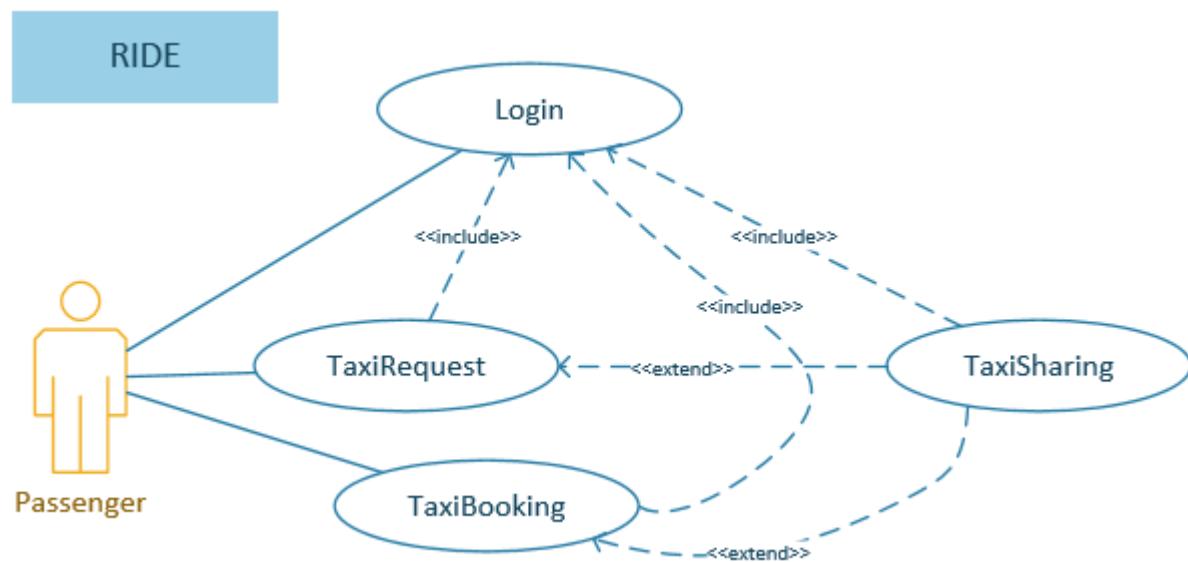
3.4.1.3 User requests for a taxi ride

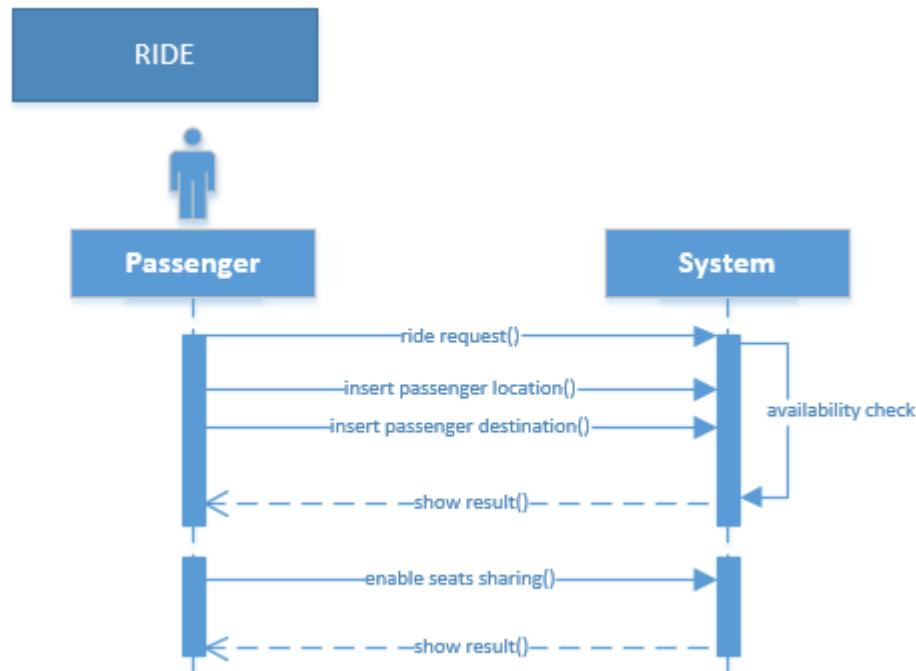
Actor	Registered User
Goal	[G1] [G3]
Input Condition	Registered User is already logged in into the site/app.
Event Flow	<ol style="list-style-type: none">1. Registered User navigate to the simple ride request area.2. User insert current address and destination address.3. User activate or not the 'sharing' function.4. User click on 'Request Taxi'.
Output Condition	The site/app shows a popup with the result of the request.
Exception	If the User wants to cancel the request process: -can go back in the browser





-in the popup can push the X button

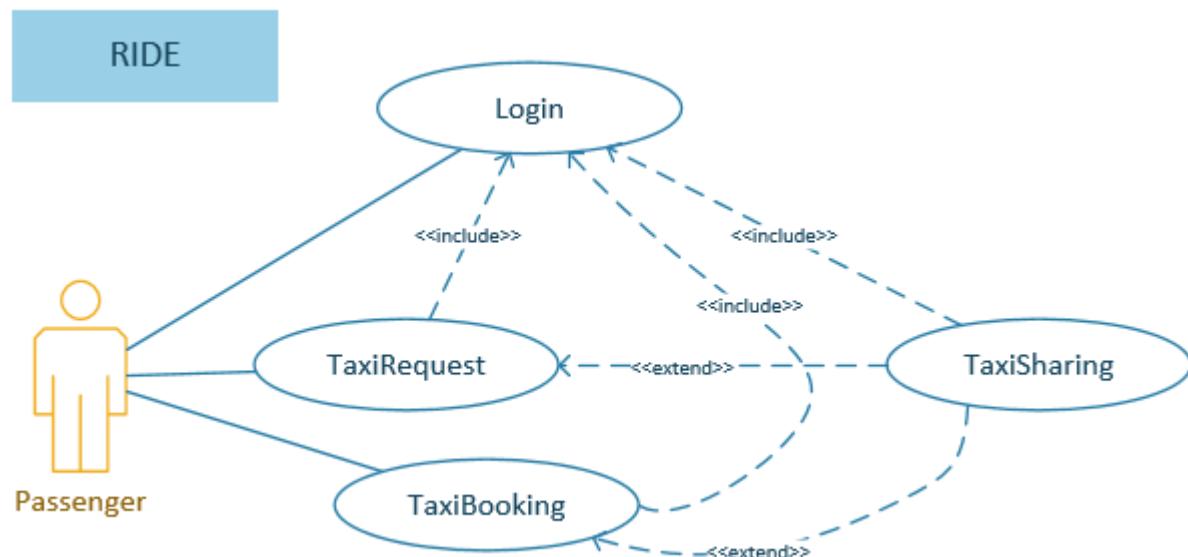


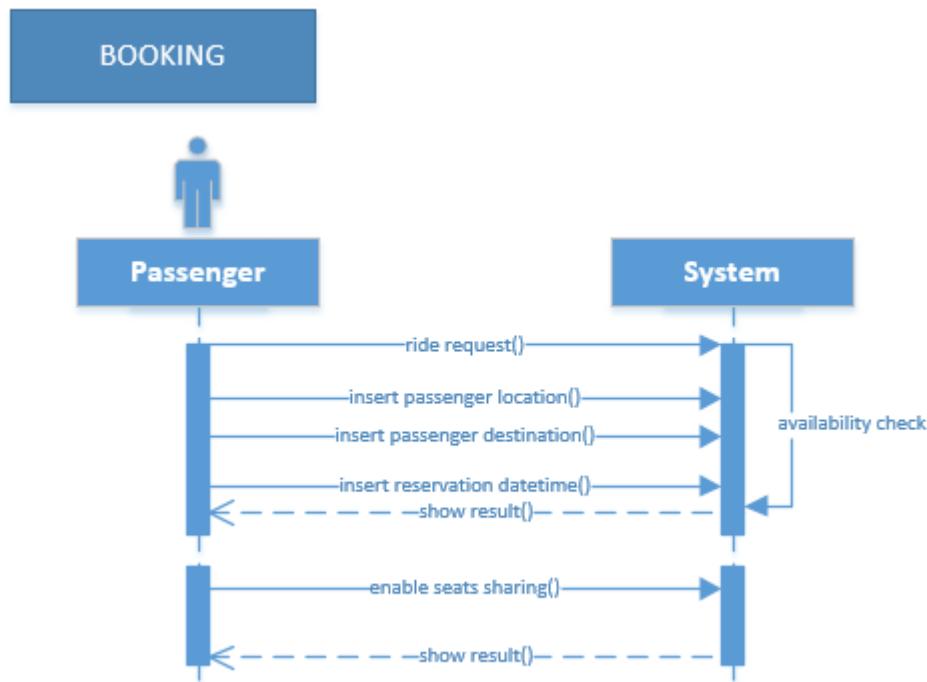




1.1.1.1 User requests for a ‘booking’ taxi ride

Actor	Registered User
Goal	[G2] [G3]
Input Condition	Registered User is already logged in into the site/app.
Event Flow	<ol style="list-style-type: none">1. Registered User navigate to the ‘booking’ ride request area.2. User insert current address and destination address.3. User select the date and the time for which he/she wants to have the taxi.4. User activate or not the ‘sharing’ function.5. User click on ‘Reserve Taxi’.
Output Condition	The site/app shows a popup with the result of the request.
Exception	If the User wants to cancel the reservation process: -can go back in the browser -in the popup can push the X button





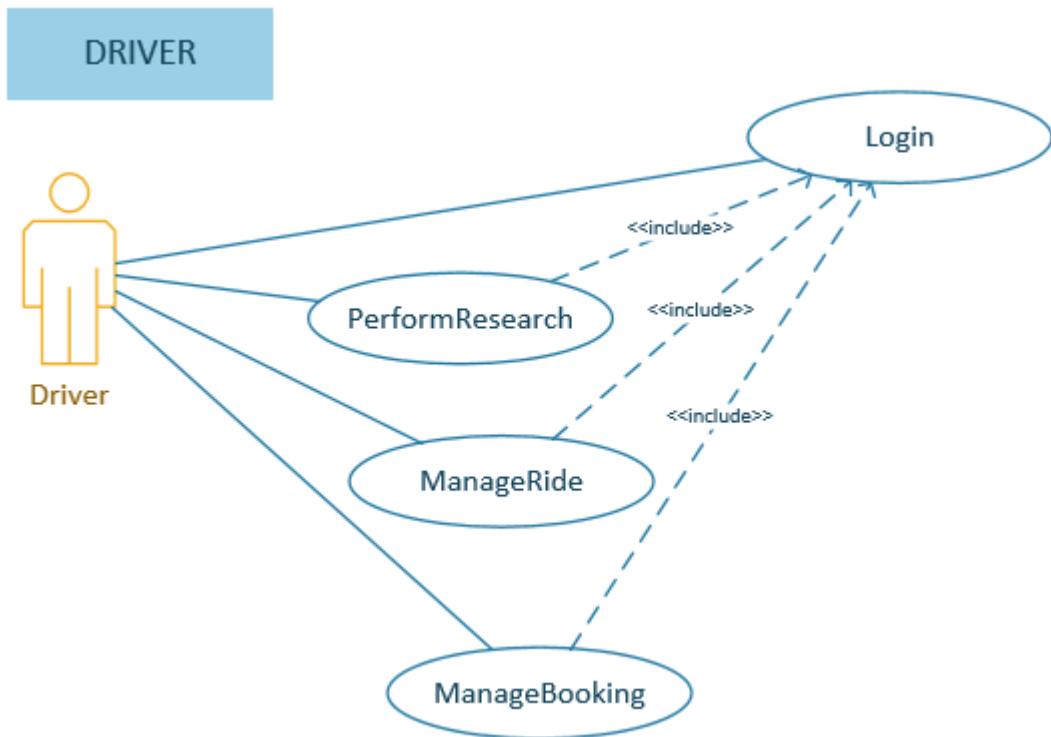
3.4.1.4 Driver management of new rides

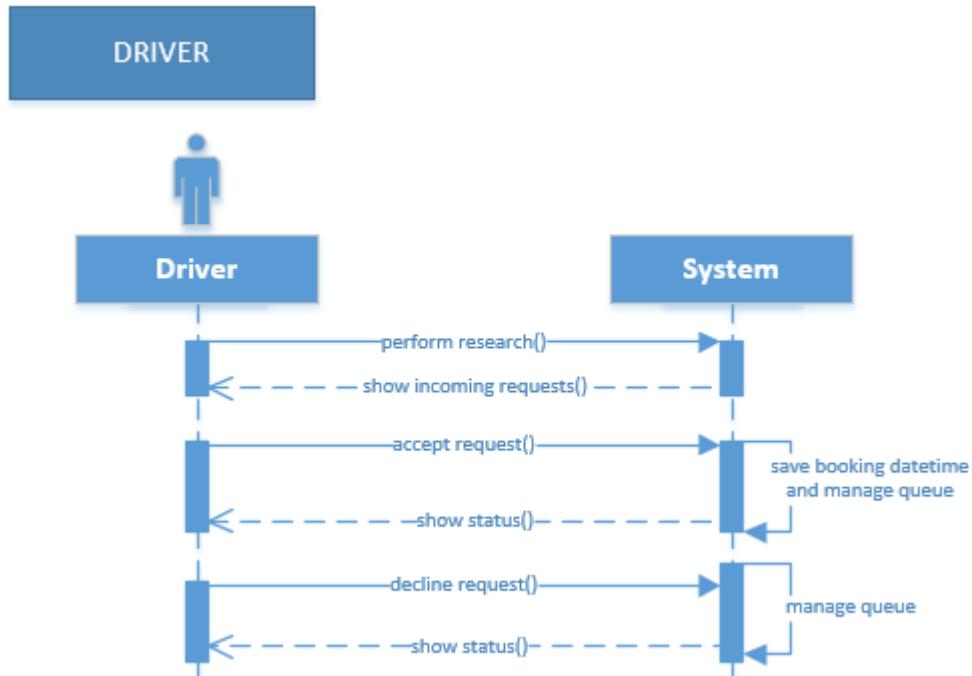
Actor	Registered Driver
Goal	[G4]
Input Condition	Driver is already logged in into the app.
Event Flow	<ol style="list-style-type: none">1. Driver navigate to the private 'Driver Area'.2. Driver scroll the page and find a good ride for him/her.3. Driver select a ride with the tick ✓ button.4. Driver can hide from the view a ride, using the close X button.





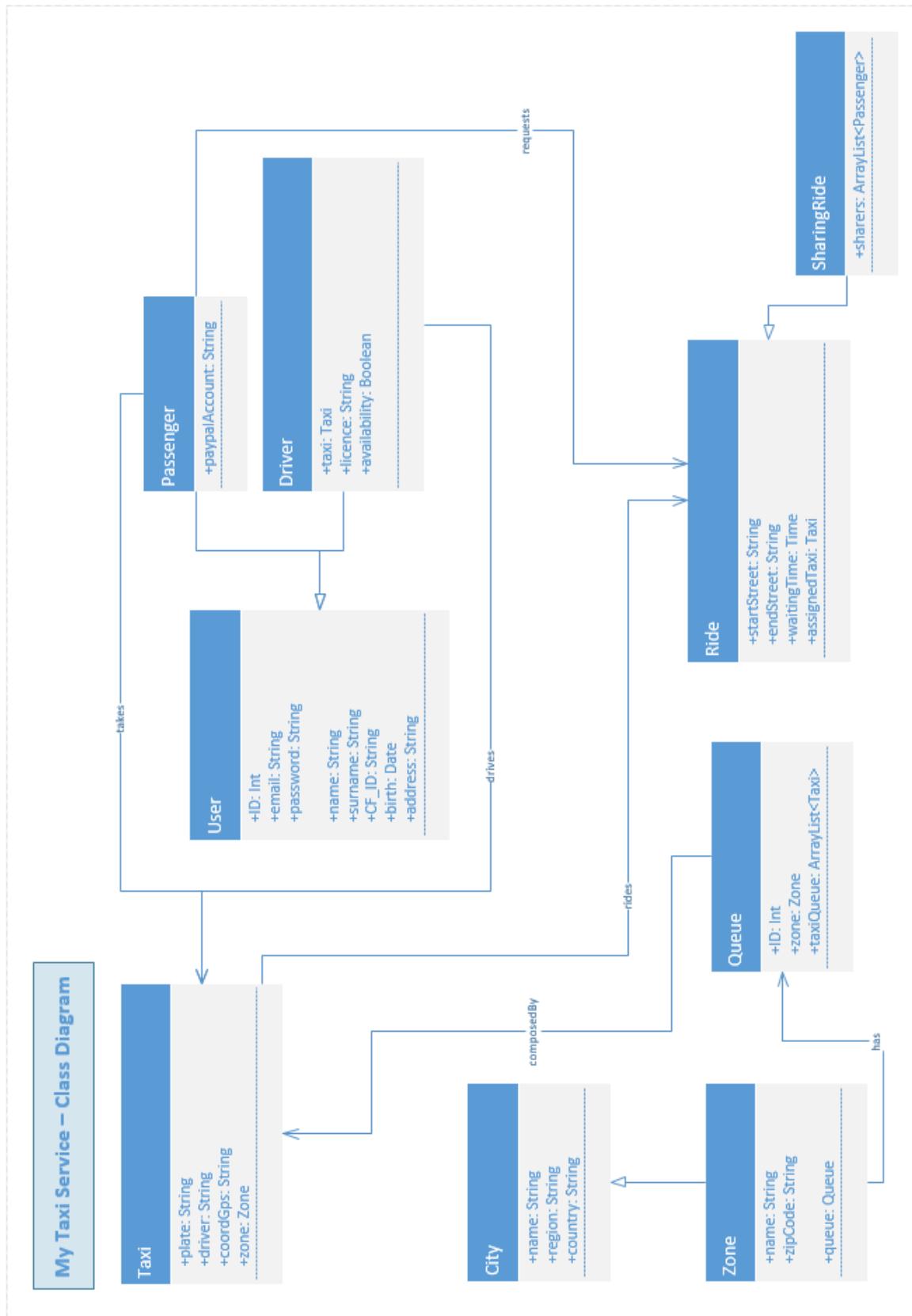
Output Condition	The site/app shows a popup with the confirmation for the taking charge of the ride.
Exception	A Driver cannot refuse a confirmed ride. If a Driver wants to cancel the charge for a ride: -has to call the callcenter that will contact the user at his registered email address







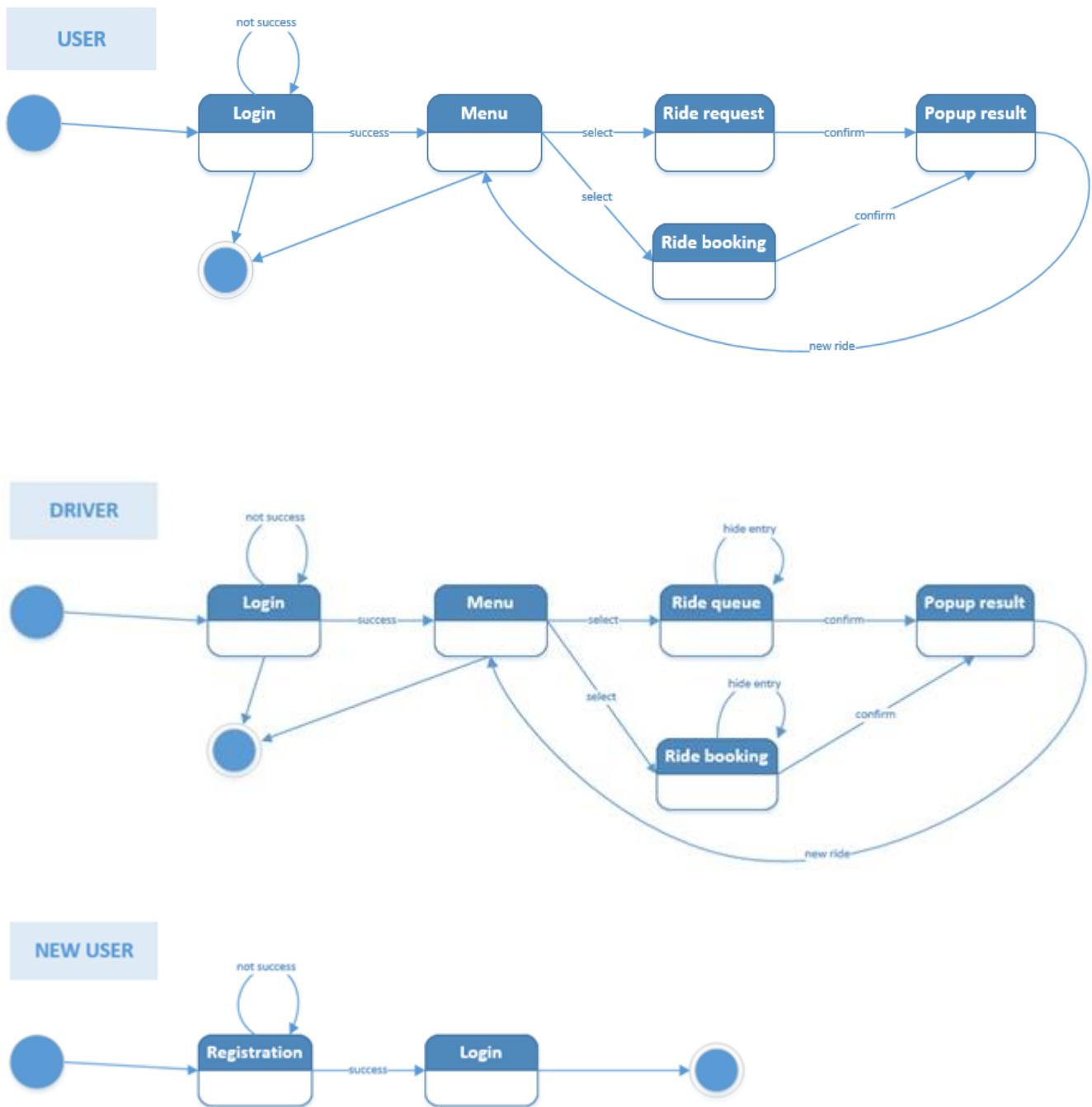
3.4.2 Class Diagram





3.4.3 Finite State Machine Diagrams

The following state-machine diagram would give a simplified vision of entire application.







3.5 Non Functional Requirements

3.5.1 Performance Requirements

Performance must be acceptable to guarantee a good grade of usability. We assume the response time of the system is close to zero, so the performance are essentially bounded by users internet connection.

3.5.2 Design Constraints

The application will be developed using Java EE so it will inherit all language's constraints.

3.5.3 Software System Attributes

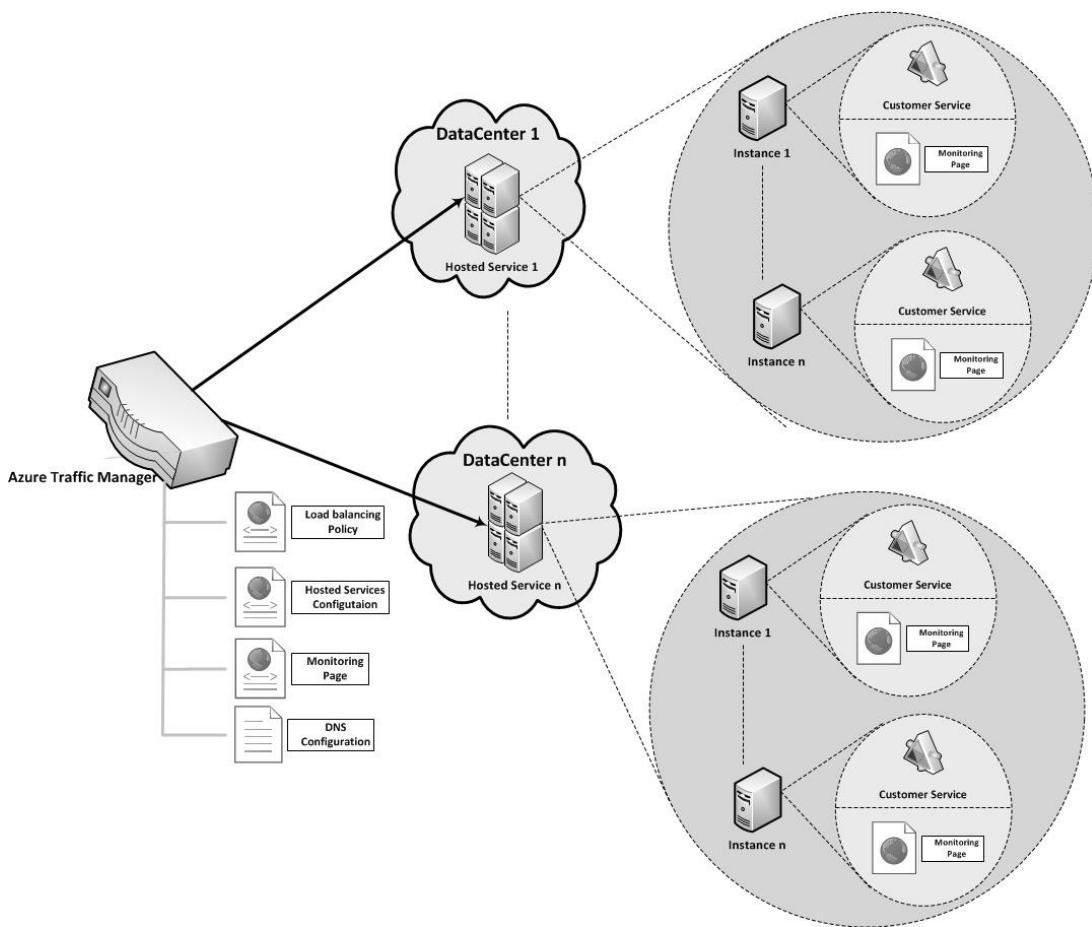
3.5.3.1 Availability

As all distributed sites/apps, MyTaxiService will be accessible online anytime.

The backend uses a variety of services:

- hosting machine
- webserver
- database server
- load balancer
- IP failover
- CDN

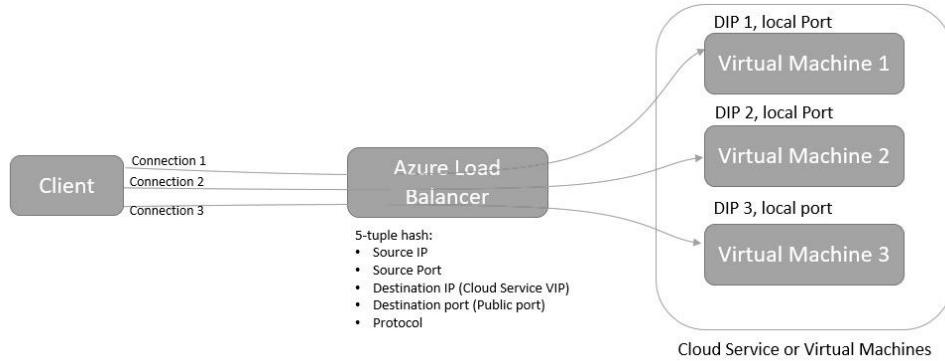




To guarantee a future scalability of the service, all system could be hosted into cloud platform like Microsoft Azure.

This solution gives more scalability to performance required by the system and could reduce the cost for dedicated server, maintaining an high level of performance especially in case of full load with a lot of requests per second.





3.5.3.2 Maintainability

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

3.5.3.3 Portability

The application could be used on any SO which supports JVM and DBMS.

3.5.4 Security

3.5.4.1 External Interface Side

- **Login**

MyTaxiService application embeds a login authentication to protect the information of users. The connection with the server is encrypted.

- **Passwords**

Password of user is saved using hashing mechanism like SHA-128.





This system does not actually check the password length, so could be developed a system that require an 8-character password with number, letter and special character.

Password also is static, user is not involved in password changing so system will ask to user to change frequently the password for example every 3 months.

To have a more secure system, it should be implemented a login system with a captcha code, to prevent botnet attack.

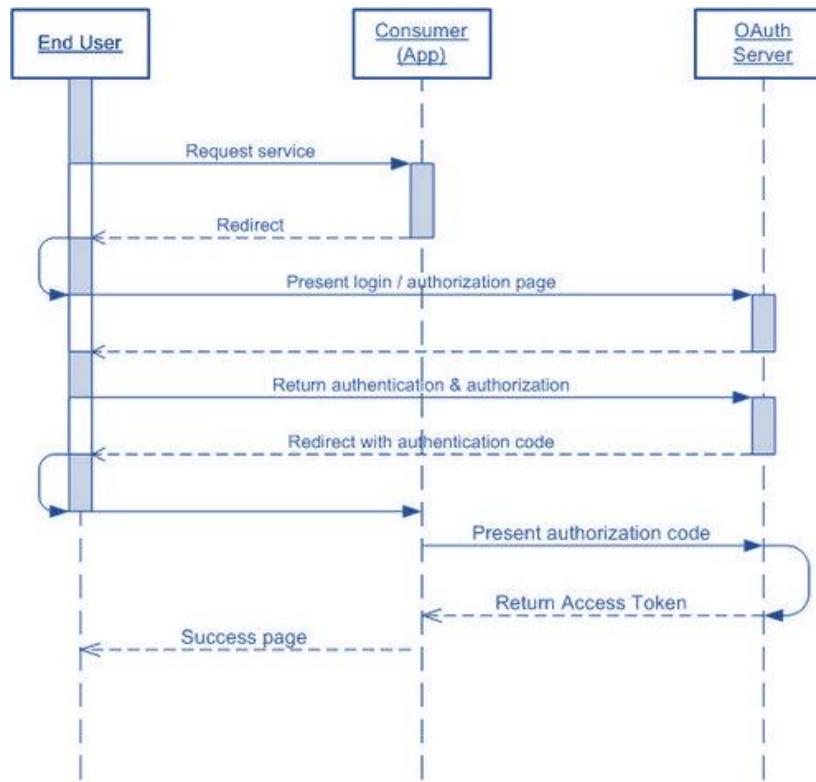
- **Multifactor authentication**

Could also be implemented an authentication system with a mix of these technologies:

- ✓ Two-factor authentication with a code sent by email or SMS to the user
- ✓ OTP one-time password, composed by a card static password list owned by user or a dynamic embedded password generator (ex. using a code-generator app). The system asks for random code on that card or an entire code generated.
- ✓ OAUTH login system: to authenticate a user by other public authentication protocols like Facebook, Twitter, Windows Live etc.. (ex. 'Login with Facebook' Button)

Follows the Active Diagram as example of OAUTH events flow:





3.5.4.2 Application Side

On the application side could be implemented a security backend to protect the application against multiple attacks

- **SQL Injection**

Could be implemented a filtering system to all form.

Malicious user can fill the form with an ad-hoc SQL code to have access to information who normally cannot have access.

- **Buffer overflow**

Buffer overflows can be triggered by inputs that are designed to execute code, or alter the way the program operates. This may result in erratic program behavior, including memory access errors, incorrect results, a crash, or a breach of system security.



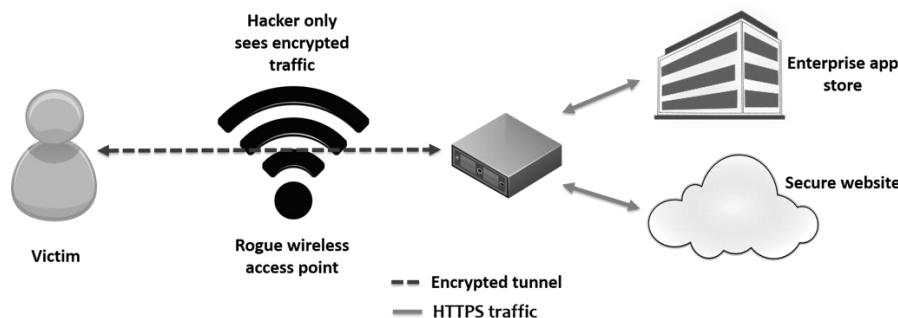


A simple way to avoid the BOF attack is to constantly check the integrity of the requests, and in case of insanely big memory request, drop it.

- **MITM Attack**

Implement the https connection instead of http is a best effort method to guarantee communication confidentiality and integrity and also mutual authentication.

Moreover the SSL protocol (in common use with a Digital Certificate signed by a Certification Authority(CA)) is resistant to the Man In the Middle attack.

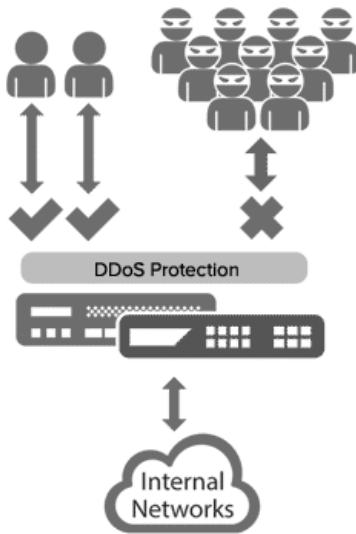


3.5.4.3 Server Side

The server side architecture could be implemented dividing strongly the data from application. In particular, a real threat is the DoS attack.

- **DDoS Attack**





A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service.

Defensive responses to denial-of-service attacks typically involve the use of a combination of attack detection, traffic classification and response tools, aiming to block traffic that they identify as illegitimate and allow traffic that they identify as legitimate.

This can be done with a use of the cloud based services like traffic managers and virtual routers/virtual switches like the MS Azure ones.

- **DNS Spoofing**

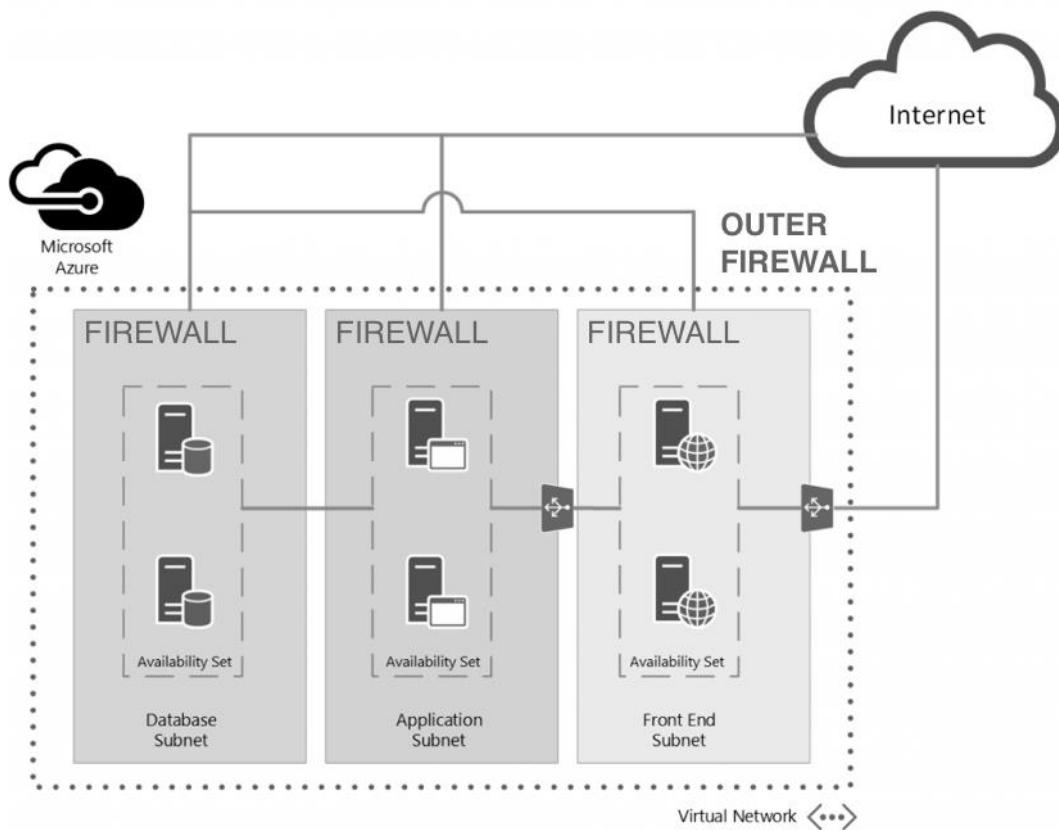
DNS spoofing (or DNS cache poisoning) is a computer hacking attack, whereby data is introduced into a Domain Name System (DNS) resolver's cache, causing the name server to return an incorrect IP address, diverting traffic to the attacker's computer (or any other computer).

A protection from this attack could come by services like 'Secure DNS' (*DNSSEC*) that uses cryptographic digital signatures signed with a trusted public key certificate to determine the authenticity of data.

- **Firewall protection**

An idea of possible secure infrastructure is well represented by the following picture. Application Server is separated from database and from the web server. All zones are divided by firewall. Access to this zone is restricted and forbidden to not authorized user.







TOTALE ORE USATE (AGGIUNGERE VIA VIA)

NUMERO DI ORE USATE (somma Leo+Sara o coworking)

+4h (19/10)

+3h (20/10)

+3h (21/10)

+4h (22/10)

+5h (25/10)

+8h (26/10)

+4h (27/10)

+8h (28/10)

+10h (29/10)

+8h (30/10)

+h (/10)

+h (/10)

+h (/10)

+h (/11)

+h (/11)

+h (/11)

+h (/12)

+h (/12)

+

