

myTaxiService

-

**Requirements Analysis and Specification
Document**

Davide Cremona, Simone Deola

November 3, 2015

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Actual System	4
1.3	Scope	4
1.4	Actors	5
1.5	Goals	5
1.6	Definitions, Acronyms, Abbreviations	6
1.6.1	Definitions	6
1.6.2	Acronyms	6
1.6.3	Abbreviations	6
1.7	Reference documents	6
1.8	Document overview.	6
2	Overall Description	7
2.1	Product perspective	7
2.2	User Characteristics	7
2.3	Constrains	7
2.3.1	Regulatory Policies	7
2.3.2	Hardware Limitations	7
2.3.3	Software Limitations	8
2.3.4	Parallel Operations	8
2.3.5	Documents Related	8
2.4	Assumptions	8
2.5	Future possible Implementations	9
3	Specific Requirements	10
3.1	External Interface Requirements	10
3.1.1	User Interfaces	10
3.1.1.1	Login	10
3.1.1.2	Customer registration	12
3.1.1.3	Taxi Drivers registration	14
3.1.1.4	Customer home page	16
3.1.1.5	Reservation page	18
3.1.1.6	My reservation Page	20

3.1.1.7	Customer notification pop-up	22
3.1.1.8	User page	24
3.1.1.9	Taxi Driver home page	26
3.1.1.10	Taxi Driver notification	28
3.1.1.11	Registered user end ride page	30
3.1.2	Hardware Interfaces	32
3.1.3	Software Interfaces	32
3.1.4	Interfaces to Others Application	32
3.1.5	Communication Interfaces	33
3.2	Functional Requirements	33
3.2.1	Functional Requirements for Guest Users	34
3.2.1.1	[G.1 - Allow guest users to become a customer creating a myTaxiService Account]	34
3.2.1.2	[G.2 - Allow a Guests User to become a Customer using his Facebook Account]	34
3.2.1.3	[G.3 Allow guest users to become a taxi driver]	34
3.2.1.4	[G.4 Allow Guest Users to log in with myTaxiService account.]	35
3.2.1.5	[G.5 Allow Guest Users to log in with Facebook account.]	35
3.2.2	Functional Requirements for Registered Users	35
3.2.2.1	[G.6 Allow a Registered User to view or modify his username and email.]	35
3.2.2.2	[G.7 Allow a Registered User to retrieve his password if he doesn't remember it.]	36
3.2.2.3	[G.8 Allow a Registered User to signal another one if he has made a bad use of the system.]	36
3.2.3	Functional Requirements for Customers	37
3.2.3.1	[G.9 Allow customers to require a taxi.]	37
3.2.3.2	[G.10 Allow customers to reserve a ride.]	37
3.2.3.3	[G.11 Allow customers to delete a previous reservations.]	38
3.2.4	Functional Requirements for Taxi Drivers	38
3.2.4.1	[G.12 Allow Taxi Drivers to accept or decline a ride request.]	38
3.2.4.2	[G.13 Allow Taxi Drivers to notify their availability.]	39
3.3	Scenarios	39
3.3.1	S.1 Registration as a Customer creating a myTaxiService account	39
3.3.2	S.2 Registration as a Customer using a Facebook Account	39
3.3.3	S.3 Registration as a Taxi Driver creating a new Taxi Driver myTaxiService account.	40
3.3.4	S.4 Login with a myTaxiService account.	40
3.3.5	S.5 Customer Login with Facebook	40
3.3.6	S.6 Profile Modification.	41
3.3.7	S.7 Password Retrieval	41

3.3.8	S.8 Reporting Abuses	41
3.4	UML Models	42
3.4.1	Use-Case Diagrams	42
3.4.2	Class Diagrams	43
3.4.2.1	Registration through the creation of a new my-TaxiService account.	43
3.4.2.2	Registration using a Facebook account.	44
3.4.2.3	Taxi Driver Registration	46
3.4.2.4	Registered User Login	47
3.4.2.5	Customer Facebook Login	49
3.4.2.6	Profile Modification	49
3.4.2.7	Password Retrieval	51
3.4.2.8	Taxi Driver Reporting	52
3.4.2.9	Customer Reporting	53
3.4.3	State Machine Diagrams	54
3.5	Non Functional Requirements	54
3.5.1	Performance Requirements	54
3.5.2	Design Constraints	54
3.5.3	Software System Attributes	54
3.5.3.1	Availability	54
3.5.3.2	Maintainability	54
3.5.3.3	Portability	54
3.5.4	Security	54
3.5.4.1	External Interface Side	54
3.5.4.2	Application Side	54
3.5.4.3	Server Side	54
4	Appendix	55

Chapter 1

Introduction

1.1 Purpose

This document is the R.A.S.D. (Requirement Analysis and Specification Document). The purpose of this document is the description of the "myTaxiService" system. At first, it will provide functional and non-functional requirements, a complete overview of the constraints of the system and its limits. Then it will explain in detail the dynamics of the system using real-life use cases. Finally this document will provide a base for the developers that concretely have to implement the system.

1.2 Actual System

The functionality that the new system will provide is now not supported. So the entire system must be developed without using or modifying existing system.

1.3 Scope

The objective of myTaxiService is to provide an interface between customers and taxi drivers to optimize their interaction and provide a fair management of taxi queues. The users, once registered through the mobile application or the web application, can request a taxi for their travel or reserve one, specifying the origin and the destination. The reservation can be done at least two hour before the ride; if the reservation can take place, the system will allocate a taxi 10 minutes before the meeting time. On the other side, taxi drivers can inform the system that they are waiting for a client and accept or decline a ride request. If the request has been accepted, a notification will be sent to the requesting customer with the identification number of the incoming taxi and the time he has to wait. Otherwise, if the request has been rejected it will be forwarded to the next taxi in the queue. The system has to optimize the management of customers requests giving the rides

to the taxi with the highest priority that has to be evaluated in function of availability and the nearness of the taxi driver.

1.4 Actors

- **Guest User:** guest users are unlogged or unregistered users. They can visit the login page or the registration forms.
- **Registered User:** this kind of user identify either a Guest User or a Taxi Driver.
- **Customer:** this kind of user is the end-user of the service. He can perform request for taxis or reserve a ride. In his personal page he can view his requests and the system responses.
- **Taxi Driver:** this kind of user is composed by the actual taxi drivers that can only see customers requests that has been forwarded by the system. He can accept or decline these requests. Also, he's considered a special kind of user because one can register as a "Taxi Driver" only if he provide a valid Taxi licence.

1.5 Goals

- [G.1] Allow guest user to become a customer creating a myTaxiService Account.
- [G.2] Allow guest user to become a customer using his Facebook Account.
- [G.3] Allow guest user to become a taxi driver.
- [G.4] Allow registered user to log in with myTaxiService account.
- [G.5] Allow customer to log in with Facebook account.
- [G.6] Allow a Registered User to view or modify his username and email.
- [G.7] Allow a Registered User to retrieve his password if he doesn't remember it.
- [G.8] Allow a RegisteredUser to signal another one if he has made a bad use of the system.
- [G.9] Allow customers to require a taxi.
- [G.10] Allow customers to reserve a ride.
- [G.11] Allow customers to delete a previous reservations.
- [G.12] Allow taxi drivers to accept or decline a ride request.
- [G.13] Allow taxi drivers to notify their availability.

1.6 Definitions, Acronyms, Abbreviations

1.6.1 Definitions

1.6.2 Acronyms

- RASD: Requirement Analysis and Specification Documents.
- DD: Design Document.
- UML: Unified Modeling Language.
- OS: Operative System.
- API: Application Program Interface.
- GPS: Global Positioning System.
- HTTP: Hypertext Transfer Protocol.
- HTTPS: Secure Hypertext Transfer Protocol.

1.6.3 Abbreviations

- G.x is the x-Goal
- Req.x is the x-Functional Requirement
- Dom.x is the x-Domain Assumption

1.7 Reference documents

- (IEE830) IEEE Recommended Practice for Software Requirements Specifications

1.8 Document overview.

Until now, we have given a general explanation about the software functionalities and a brief description of this document. Now we will describe what the rest of this RASD contains.

In Section 2 we will focus more on system constraints and assumptions.

In Section 3 we will describe requirements, typical scenarios and use-cases. In this section there is also a collection of UML diagrams that describes in particular the functionalities of the system.

//TODO SECTION 4

Chapter 2

Overall Description

2.1 Product perspective

The system will be composed of a web application and a mobile application developed for the three major OS (Apple iOS, Android, Windows 10). The system will provide some API with the purpose of a future connection with another travel planning systems.

2.2 User Characteristics

The users that we suppose will use our system are of two types. The ones who want to find a taxi for a travel in the simplest way (customers). The others are taxi drivers that want to increment their productivity. The first ones must be able to access to a web browser or download and using a mobile application, the second ones also must have a taxi license.

2.3 Constrains

2.3.1 Regulatory Policies

myTaxiService has to meet regulatory policies about taxis in the countries where it will be used.

2.3.2 Hardware Limitations

The only hardware limitation that the myTaxiService mobile application has to meet is the mobile phones characteristics. the rest of the system will not be affected by particular hardware limitations.

2.3.3 Software Limitations

myTaxiService mobile application has to be compatible with all major mobile operating systems (Android, Apple iOS, Windows 10). Moreover, myTaxiService web application has to be compatible with all major browsers (Chrome, Safari, Firefox, Microsoft Edge).

2.3.4 Parallel Operations

Our system must be able to perform parallel operations on the database to satisfy all the requests from multiple users.

2.3.5 Documents Related

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

2.4 Assumptions

- Every taxi driver has equipped a smartphone during working hours.
- Every taxi driver has a unique taxi license.
- Every taxi has a GPS locator to send GPS information to the central server.
- Android, Apple iOS or Windows 10 are available on the registered users smartphones.
- Every registered users can connect to the Internet with a mobile device when outside.
- When a customer require a taxi, the GPS information about his location are automatically sent to the central server.
- The reservation of a ride is made at least two hours before the ride.
- The deletion of a reservation is made at least two hours before the ride.
- The requests of the customers are automatically notified to the first taxi driver in the zone queue.
- If a taxi driver declines a request he will be placed in the bottom of the zone queue.
- If a request is declined it will be forwarded to the next taxi driver in the zone queue.
- If a customer makes a bad use of the taxi request system, he can be reported as a bad customer.

- If a taxi driver notifies his availability, it is because he is actually available
- If a taxi driver notifies his availability, it is because he wants to be notified of customers that needs a ride.
- If a taxi driver accepts a request, the requesting customer will be notified

2.5 Future possible Implementations

A possible future implementation can be a complex feedback system that allows the customers to leave a comment about the taxi driver and vice versa. For example, taxi drivers can be interested in knowing the punctuality or how is the behave of the customer that requests the ride.

Chapter 3

Specific Requirements

This chapter contains a detailed description of how the applications works and its features. It also gives a specification of the functional and quality requirements.

3.1 External Interface Requirements

This section gives a description of the various inputs and relative outputs of the system. It also gives a description of the hardware, software and communication interfaces that are necessary to make the system work. It will also provide a generic visualization of the user interface in the various user platforms.

3.1.1 User Interfaces

Here we describe in particular how the application should look like either for mobile and web application. To make an easier explanation of the aspect of the various screen of the application we are going to use Mockup.

3.1.1.1 Login

On this page the users can log in with username and password or with Facebook (only if as registered user is a Customer). If the user is not already registered can access to the registration pages (for Customer or for Taxi Drivers). Also if the users forgot the password from this page can go to the forgot password page.

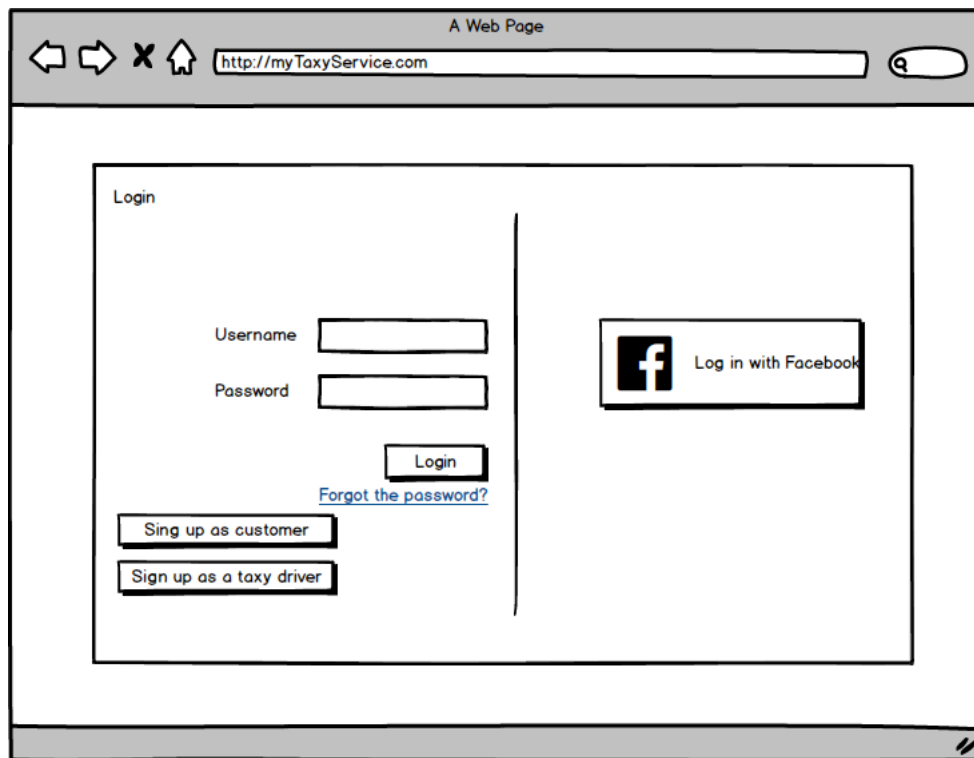


Figure 3.1: Login Page, web version

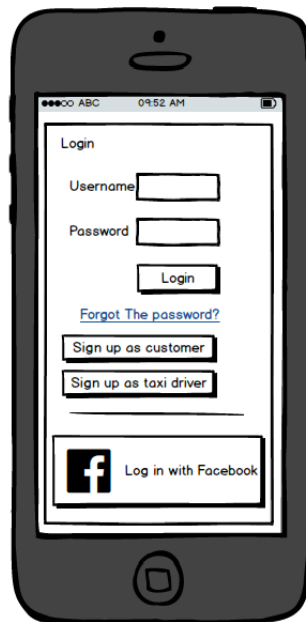


Figure 3.2: Login Page, mobile version

3.1.1.2 Customer registration

On this page the users can register itself. This page must provide two way of registration, registration with the standard form (e-mail, password and username) of with Facebook API.

A Web Page

http://myTaxiService.com

Customer registration

Username

Password

Repeat password

e-mail


 Log in with Facebook

Figure 3.3: Customer registration Page, web version

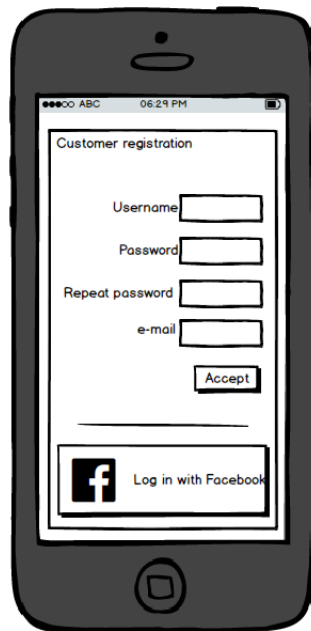


Figure 3.4: Customer registration Page, mobile version

3.1.1.3 Taxi Drivers registration

On this page the users can register itself as a taxi driver. Taxi driver have a special form to been registered because of the additional information the user must provide (taxi license number). Just because the additional information the registration can be done just with the standard form.

A Web Page

http://myTaxiService.com

Taxi drivers registration

Username

Password

Repeat password

e-mail

license number

taxi number

Accept

Figure 3.5: Taxi Driver registration Page, web version

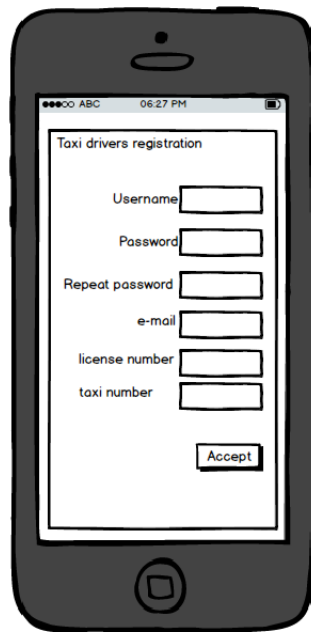


Figure 3.6: Taxi Driver registration Page, mobile version

3.1.1.4 Customer home page

On this page the Customer can see his position (information from his GPS) and perform all the main operation that he can perform. He can Request a taxi on the position showed, can go to the reservation page (in which can reserve a ride), can visit his personal page (in which can see all the information of his profile), can go to the information page (information about myTaxiService) or go to the 'my reservation page' (in which can manage all the previous reserved ride).

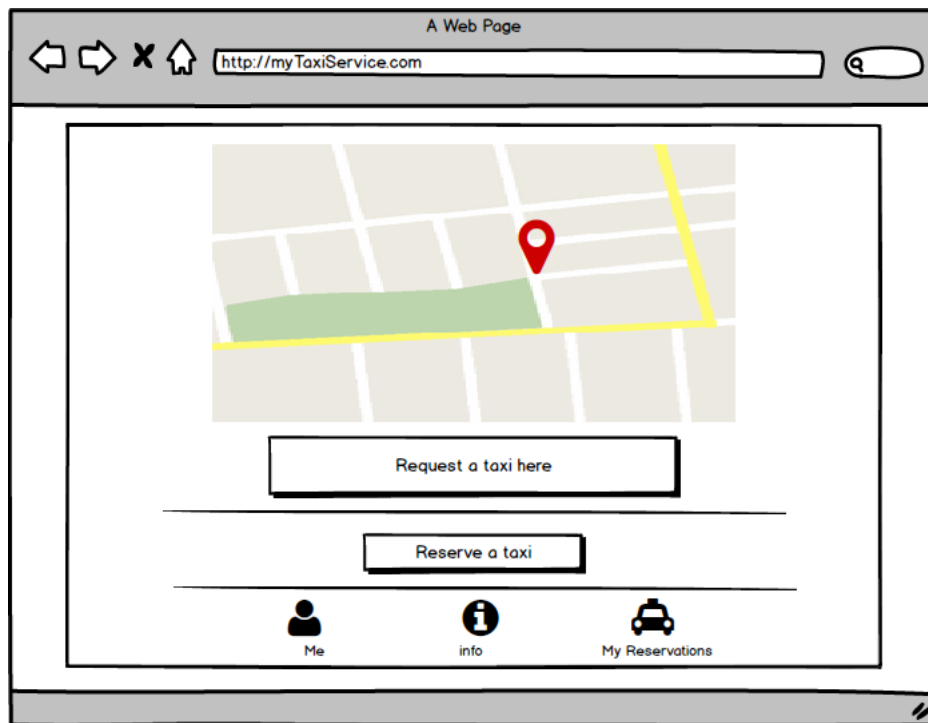


Figure 3.7: Customer home page, web version

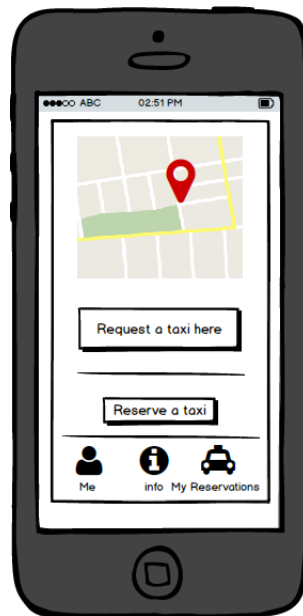


Figure 3.8: Customer home page, mobile version

3.1.1.5 Reservation page

On this page the Customer can reserve a taxi ride. To do this he must insert a Origin position, a Destination position, a Date and a Time. The reservation must be at least two hours after the current Time, so the system must avoid to reserve a previous ride.

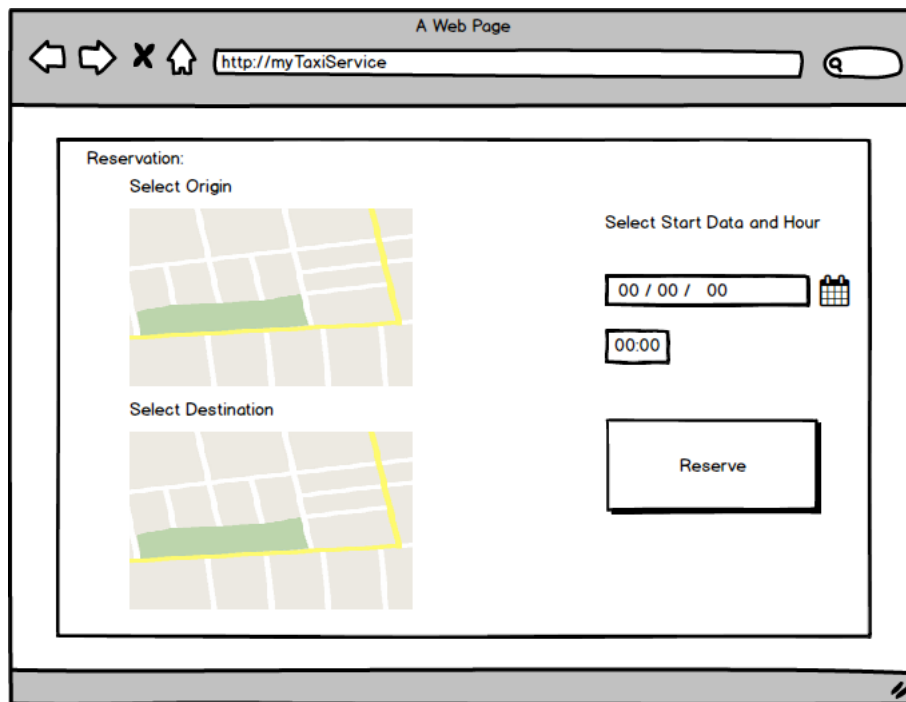


Figure 3.9: Reservation page, web version

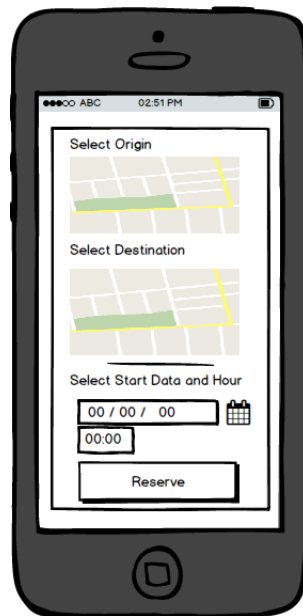


Figure 3.10: Reservation page, mobile version

3.1.1.6 My reservation Page

On this page Customer can see all the reservation he have already done and adding some new. For each previous reservation can see all the information and, if the Date is before the next two hours, can delete it.

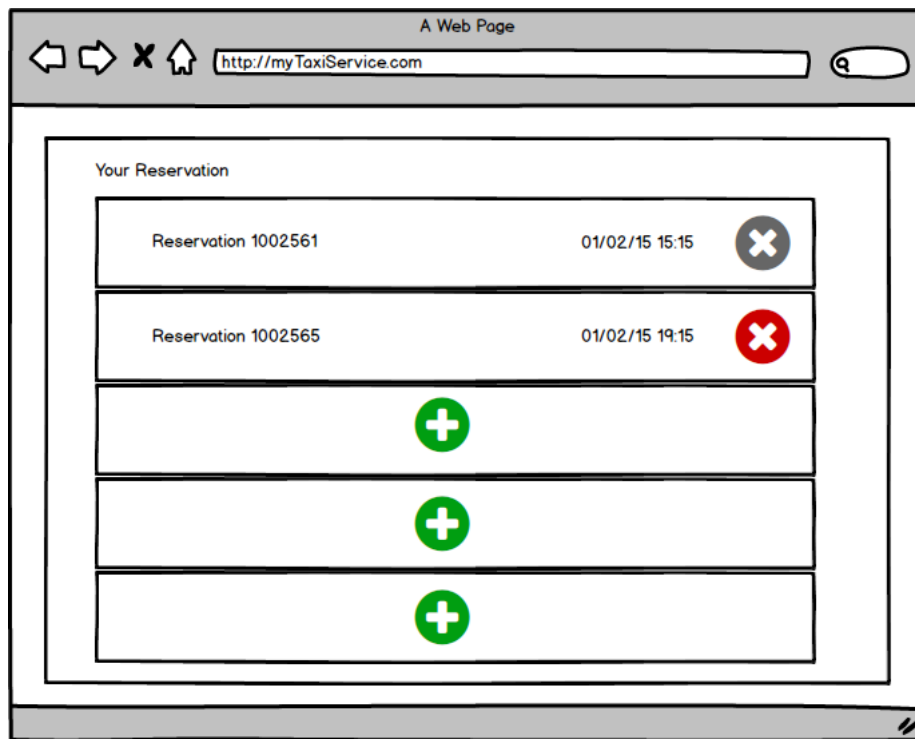


Figure 3.11: My reservation, web version

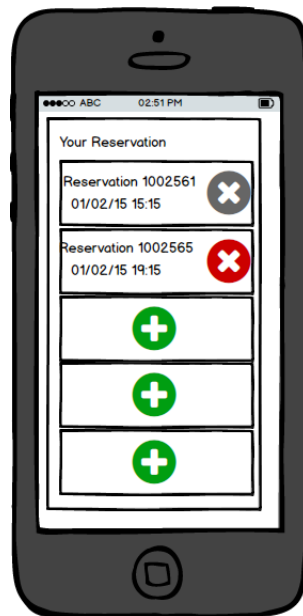


Figure 3.12: My reservation, mobile version

3.1.1.7 Customer notification pop-up

This pop-up is showed to the requesting Customer when his request has been handled. On this pop-up the system show also the number of the incoming taxi and an approssimative waiting time.



Figure 3.13: Customer notification, web version

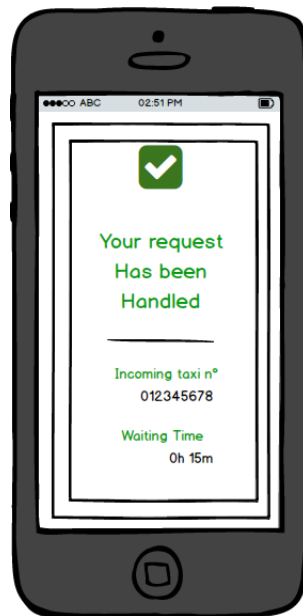


Figure 3.14: Customer notification, mobile version

3.1.1.8 User page

This page show all the information about your user and show how many time you've been notified as bad user.

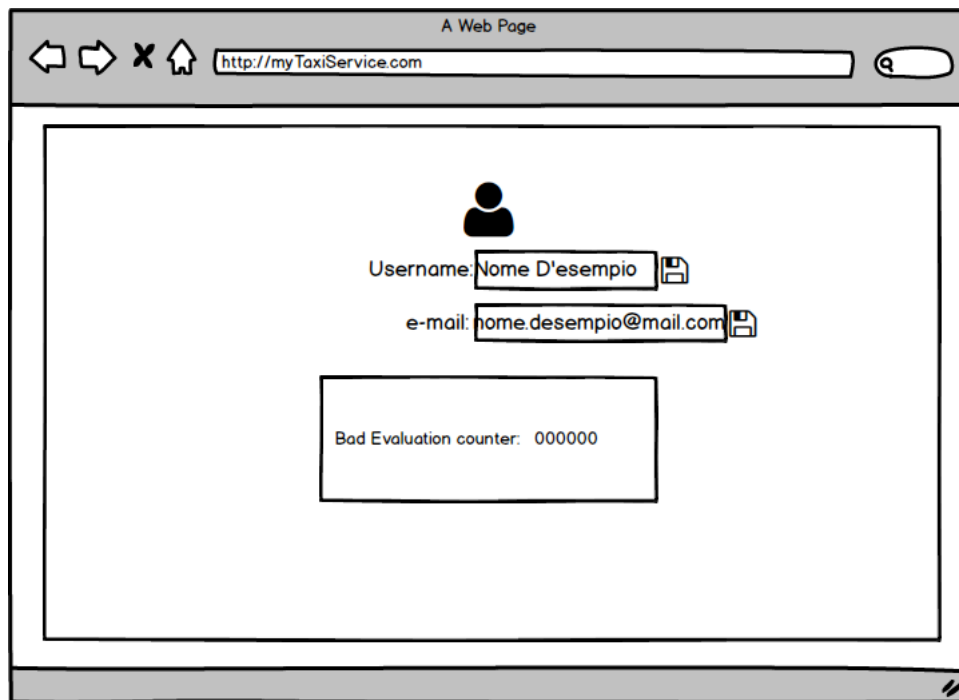


Figure 3.15: User Page, web version

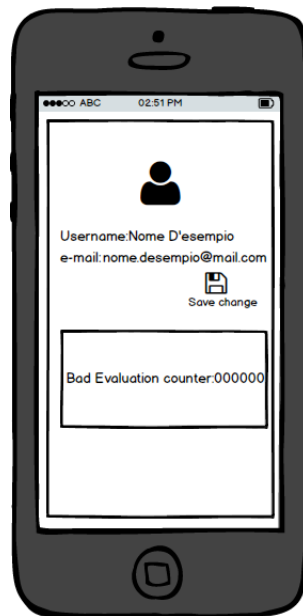


Figure 3.16: User Page, mobile version

3.1.1.9 Taxi Driver home page

On this page the Taxi driver can see his position (information from his GPS) and inform the system about his availability.

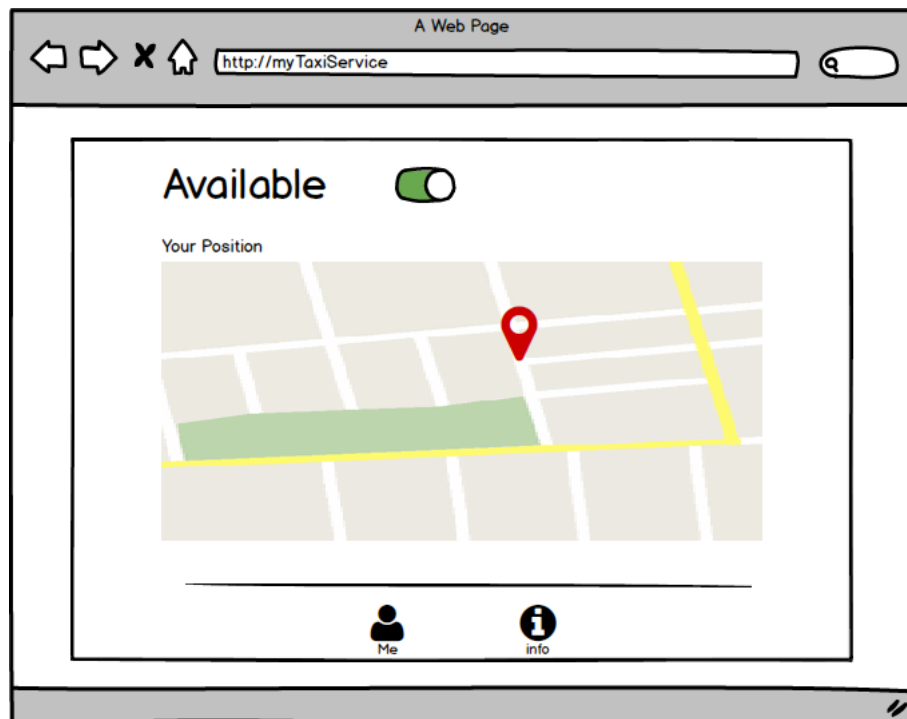


Figure 3.17: Taxi driver home page, web version

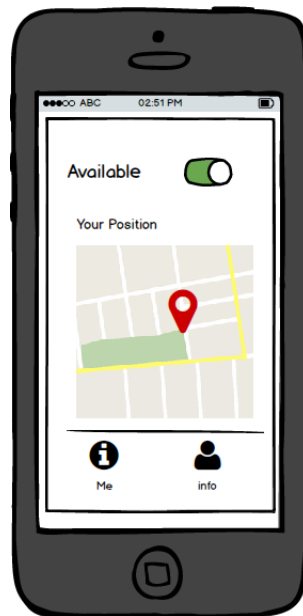


Figure 3.18: Taxi driver home page, mobile version

3.1.1.10 Taxi Driver notification

On this pop-up the taxi driver will be notified of a request that he can handle. On this pop-up, there are also two buttons, with which the taxi driver can accept or decline the request.



Figure 3.19: Taxi driver notification pop-up, web version

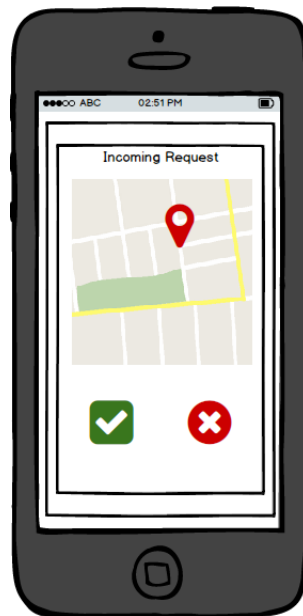


Figure 3.20: Taxi driver notification pop-up, mobile version

3.1.1.11 Registered user end ride page

When a taxi driver accept a ride this page is showed up to each registered user involved in this ride. On this page the taxi driver can check the name of the customer and the Origin position of his journey. When the ride is ended each of the registered user can give a bad evaluation to the other or simply end the ride and return to the main page.

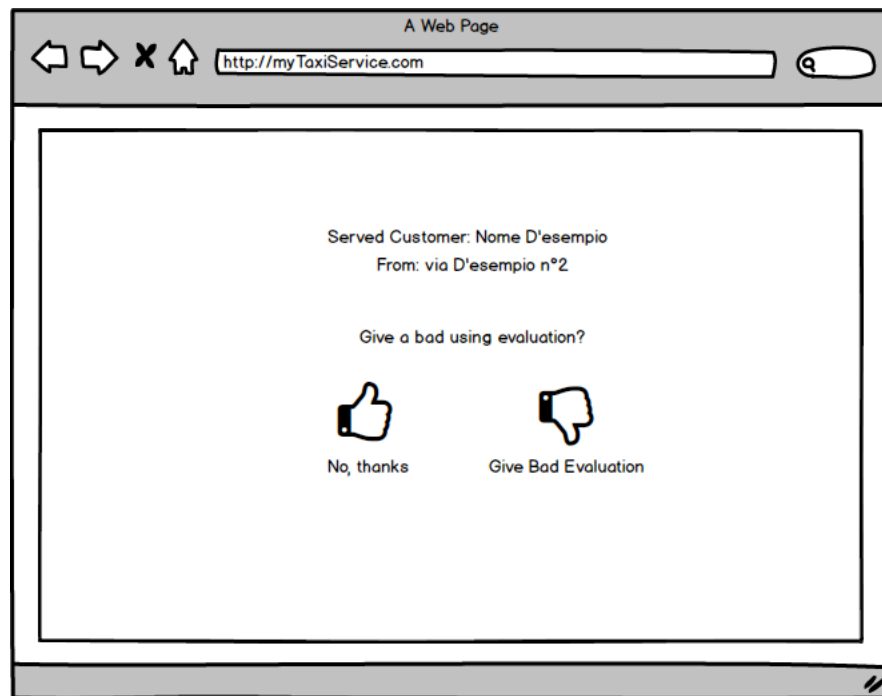


Figure 3.21: On Ride Page, web version

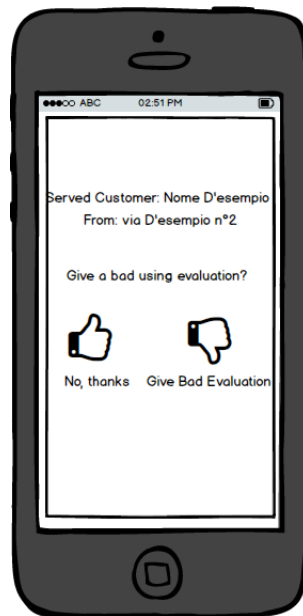


Figure 3.22: On Ride page, mobile version

3.1.2 Hardware Interfaces

Since mobile and web applications don't have any dedicated hardware we have not designed any hardware interfaces for our system. The interaction with the central database is performed by connections handled by the already installed operating system on the mobile devices or the users computers.

3.1.3 Software Interfaces

- The mobile application communicates with the GPS application in order to get geographical informations about the user.
- The web application communicates with the browser in order to get geographical informations about the user.
- Mobile and web applications communicates with the database through HTTP requests to the server.

3.1.4 Interfaces to Others Application

- myTaxiService web application require that at least one of these browsers is installed on the user Personal Computer:

Table 3.1: Browsers

Name	Version	Company	Source
Safari	9.0.1	Apple Inc.	Get Safari
Firefox	41.0	Mozilla	Get Firefox
Chrome	46.0.2490	Google	Get Chrome
Microsofr Edge	20.10240.16384.0	Microsoft	Get Edge

- myTaxiService mobile application require that at least one of these operating systems is installed on the user Smartphone:

Table 3.2: Mobile Operative Systems

Name	Version	Company	Source
Android	KitKat 4.4W.2 or later	Google	Android Info
iOS	9.1 or later	Apple Inc.	iOS Info
Windows 10	10.0.10572.0 or later	Microsoft	Windows 10 Info

- To give an additional LogIn method, we use also the "LogIn with Facebook" API relased by Facebook. Facebook Login for Apps is a fast and convenient way for people to create accounts and log into our system across multiple platforms. It is well described at Facebook Login API Page.

3.1.5 Communication Interfaces

The communication between system pieces is not specified because it is handled by the underlying operating systems for both the mobile application and the web portal.

In particular, the web and mobile applictaions will communicate with the server through HTTP/HTTPS requests.

- HTTP communicate through the port number 80 and is handled by the operating system.
- HTTPS communicate through the port number 443 and is handled by the operating system.

3.2 Functional Requirements

In this section are described, for every Actor, the Functional Requirements needed to reach the linked Goal.

3.2.1 Functional Requirements for Guest Users

Here are listed all the Functional Requirements referring to the Goals that affects Guest Users.

3.2.1.1 [G.1 - Allow guest users to become a customer creating a myTaxiService Account]

To allow the guest user to perform a successful registration the system has to:

- [Req.1] provide a registration page containing:
 1. A text box where the user can insert his username.
 2. Two text box where the user can insert his password (the second one is for security check).
 3. A text box where the user can insert his email.
 4. A button to submit informations to the system.
- [Req.2] The registration page shall be the only page accessible by a Guest User.
- [Req.3] All the informations submitted by the user must be validated by the system.
- [Dom.1] The email used by the Guest User is a valid one.

3.2.1.2 [G.2 - Allow a Guests User to become a Customer using his Facebook Account]

To allow the guest user to perform a successful registration using Facebook API, the system has to:

- [Req.1] provide a registration page containing:
 1. A button that calls the Facebook Registration API.
- [Req.2] The registration page shall be the only page accessible by a Guest User.
- [Req.3] All the informations must be evaluated by the Facebook system.

3.2.1.3 [G.3 Allow guest users to become a taxi driver]

To allow the guest user to become a Taxi Driver, the system must:

- [Req.1] provide a registration page containing:
 1. A text box where the user can insert his username.

2. Two text box where the user can insert his password (the second one is for security check).
 3. A text box where the user can insert his email.
 4. A text box where the user can insert his valid taxi license.
 5. A text box where the user can insert his valid taxi number.
 6. A button to submit informations to the system.
- [Req.2] The registration page shall be the only page accessible by a Guest User.
 - [Req.3] All the informations submitted by the Guest User must be validated by the system.
 - [Dom.1] The email used by the Guest User is a valid one.

3.2.1.4 [G.4 Allow Guest Users to log in with myTaxiService account.]

To allow a registered user to log in with his myTaxiService account, the system must:

- [Req.1] provide a login page containing:
 1. A text box where the user can insert his username.
 2. A text box there the user can insert his password.
 3. A button to submit informations to the system.

3.2.1.5 [G.5 Allow Guest Users to log in with Facebook account.]

To allow a customer to log in with Facebook account, the system must:

- [Req.1] provide a button that calls the Facebook Login API.
- [Req.2] delegate to the Facebook system all the checks about the existence of the user.

3.2.2 Functional Requirements for Registered Users

Here are listed all the Functional Requirements referring to the Goals that affects Registered Users.

3.2.2.1 [G.6 Allow a Registered User to view or modify his username and email.]

To allow a Registered User to view or modify his username and email, the system must:

- [Req.1] provide an homepage that contains:

1. A button that, if clicked, shows the page that contains the Registered User's informations.
- [Req.2] provide a page used to show Registered User's informations that contains:
 1. A clickable label showing the username of the Registered User.
 2. A clickable label showing the email of the Registered User.
 3. A box with a label inside that shows how many times the Registered User has been reported as a bad user.
 - [Req.3] The personal Registered User's page shall be only accessible after the login.

3.2.2.2 [G.7 Allow a Registered User to retrieve his password if he doesn't remember it.]

To allow a Registered User to retrieve his password, the system must:

- [Req.1] on the log in page, provide a link that will show to the Registered User a field where he can write his email to allow the system to send him an email containing the password.

3.2.2.3 [G.8 Allow a Registered User to signal another one if he has made a bad use of the system.]

To allow a Registered User to signal another one, the system must:

- [Req.1] check if the Registered User is correctly logged in.
- [Req.2] for Taxi Drivers, provide an end-ride page that contains:
 1. A label showing the username of the Customer that he has just served.
 2. A label showing the starting location of the current ride.
 3. A label that asks to the Taxi Driver if he want to report the Customer.
 4. A button that ends the current ride without reporting the Customer.
 5. A button that ends the current ride increments the Customer's Bad Evaluation Counter.
- [Req.3] When one of the Taxi Driver end-ride page button is pressed, the system must ensure that the page is not reachable anymore and the Taxi Driver must access only to his homepage.
- [Req.4] for Customers, provide an end-ride page that contains:
 1. A label showing the username of the Taxi Driver that has just served him.

2. A label showing the starting location of the current ride.
 3. A label that asks to the Customer if he want to report the Taxi Driver.
 4. A button that ends the current ride without reporting the Taxi Driver.
 5. A button that ends the current ride and increments the Taxi Driver's Bad Evaluation Counter.
- [Req.5] When one of the Customer end-ride page button is pressed, the system must ensure that the page is not reachable anymore and the Customer must access only to his homepage.
 - [Req.6] The end-ride page shall be only accessible after the Registered User login.
 - [Dom.1] The ride is ended after a Taxi Driver or a Customer action.

3.2.3 Functional Requirements for Customers

Here are listed all the Functional Requirements referring to the Goals that affects Customers.

3.2.3.1 [G.9 Allow customers to require a taxi.]

To allow the customers to require a taxi, the system must:

- [Req.1] provide an homepage that contains:
 1. A map showing the current location of the customer.
 2. A button that calls the function of the system to require a taxi (called "require button").
- [Req.2] The require-taxi page shall be only accessible by Customers and after they have performed the login as a Customer.

3.2.3.2 [G.10 Allow customers to reserve a ride.]

To allow the customers to reserve a ride, the system must:

- [Req.1] provide an homepage that contains:
 1. A button that, if clicked, shows the page used to reserve a ride.
- [Req.2] provide a page used to reserve a ride that contains:
 1. A map where the customer can select the starting location for his ride.
 2. A map where the customer can select the ending location for his ride.
 3. A field where the customer can insert the starting date for the ride.
 4. A field where the customer can insert the starting time for the ride.

- 5. A button that calls the function of the system to reserve a ride (called "reserve button").
- [Req.3] The page used to reserve a ride shall be only accessible after the Customer login.
- [Req.4] "reserve button" must be clickable if and only if the reserve date and time is at least two hours after the current time. (for instance: if the current date is 10/10/2015 and current time is 10.00, the reservation time must be at least 12.00 of the 10/10/2015).
- [Req.5] The system must notify the Customer that a Taxi Driver has accepted his reservation at least 10 minutes before the ride.

3.2.3.3 [G.11 Allow customers to delete a previous reservations.]

To allow a customer to delete a previous reservation, the system must:

- [Req.1] provide an homepage that contains:
 1. A button that, if clicked, shows the page that contains all the reservations made by the customer.
- [Req.2] provide a page that contains all the reservations made by the customer:
 1. All reservations that have a difference between the start time and the current time of less than two hours are not erasable
 2. Other reservations are erasable.
 3. The page also contains a button to add a new reservation. If clicked that button leads the customer to the page used to reserve a ride (described in the Functional Requirements Req.3 of G.4)
- [Req.3] The page that contains all the reservations shall be only accessible after the Customer login.
- [Dom.1] The reservation is deleted after the Customer action.

3.2.4 Functional Requirements for Taxi Drivers

Here are listed all the Functional Requirements referring to the Goals that affects Taxi Drivers.

3.2.4.1 [G.12 Allow Taxi Drivers to accept or decline a ride request.]

To allow Taxi Drivers to accept or decline ride requests, the system must:

- [Req.1] notify the Taxi Driver (if he's available) that is at the top of the zone-queue that a new taxi-request is made (Req.4 of G.9).
- [Req.2] on the notification screen, show two buttons: one for the acceptance and one for the declination of the taxi-request.

3.2.4.2 [G.13 Allow Taxi Drivers to notify their availability.]

To allow Taxi Drivers to notify their availability, the system must:

- [Req.1] provide an home page that contains:
 1. An ON/OFF button that signals to the system that the Taxi Driver is Available/Not Available.
 2. A map showing his current zone and position.
- [Req.1] The home page should be accessible only if the Taxi Driver is logged in.

3.3 Scenarios

Here are described in natural language some useful scenarios that shows how the system should work in different cases:

3.3.1 S.1 Registration as a Customer creating a myTaxiService account

To read the Functional Requirements for this scenario, refer to G.1 Functional Requirements Specification

Aristotle is new in the system and he wants to simplify the way he calls for a taxi. So he download the "myTaxiService" application from the Play Store (since he has a smartphone with the fancy Android 5.0 Lollipop Operative System) and he open it. The application shows to Aristotle the login screen and he taps on the "Sign Up as a Customer" button (see Login Screen) going to the registration page. In that page, he has to fill the form (see Customer Registration Screen (Mobile)) and after that he clicks on the "Accept" button. If the data entered by Aristotle are ok, he is automatically logged in and he can start using myTaxiService. Otherwise, he is redirected to an error page, that describes what is the problem.

3.3.2 S.2 Registration as a Customer using a Facebook Account

To read the Functional Requirements for this scenario, refer to G.2 Functional Requirements Specifications

Pythagoras is new in the system and he wants to simplify the way he calls for a taxi, but he doesn't want to loose time inserting his credential into a boring registration form, so he decide to register with his Facebook Account. He download the application from the Apple App Store (since he has a glorious iPhone 6 Plus) and he open it. The application shows to Pythagoras the login screen and he taps on the "Log in With Facebook" button calling the Facebook Login API. If the registration

is successful, the Facebook System will reply with the informations about the user and the myTaxiService system can create a record in the database with Pythagoras informations. Otherwise, he is redirected to an error page, that describes what is the problem.

3.3.3 S.3 Registration as a Taxi Driver creating a new Taxi Driver myTaxiService account.

To read the Functional Requirements for this scenario, refer to G.3 Functional Requirements Specifications

Thales is a great taxi worker and he wants to improve his service to the customers so he decide to join myTaxiService. First of all he visit the myTaxiService web page from his pc. That page shows the login screen (see Login Screen) and he clicks on the "Sign up as a Taxi Driver" button. The web application redirect him to the Taxi Drivers registration page. Thales fills up the form (see Taxi Driver Registration Screen (Web)) and clicks on the "Accept" button. The system perform a special check for the validity of the given taxi licence and taxi number. If the data entered by Thales are ok, he is automatically logged in and he can start using myTaxiService as a Taxi Driver. Otherwise, he is redirected to an error page, that describes what is the problem.

3.3.4 S.4 Login with a myTaxiService account.

To read the Functional Requirements for this scenario, refer to G.4 Functional Requirements Specifications

Epicurus is a Registered User and he wants to login to use the myTaxiServer Application with his smartphone. So he taps on the application icon and start using it. myTaxiService Application shows to him the login screen (see Login Screen) and Epicurus fills up the form with his credentials. After that he submits the data by pressing the "Login" button. The Application calls the server function to check the credentials of Epicurus and if them are ok, the server reply to the application with a success message and the homepage is shown to Epicurus. Otherwise the server reply with an error message that is shown to Epicurus.

3.3.5 S.5 Customer Login with Facebook

To read the Functional Requirements for this scenario, refer to G.5 Functional Requirements Specifications

Solon is a Customer of myTaxiService and he's registered with his Facebook account as described in the scenario S.2. Now he wants to log in and use myTaxiService from his iPhone. First of all he opens the application that shows the login screen (see Login Screen). In a second moment, he clicks on the "Log in with Facebook" button and he logs in.

3.3.6 S.6 Profile Modification.

To read the Functional Requirements for this scenario, refer to G.6 Functional Requirements Specifications

Zeno is a Registered User but he wants to change his username. He opens his browser and navigates to the myTaxiService Web Application. He logs in to the system and the application shows his homepage (see Homepage). Zeno clicks on the "Me" button to view the profile page (see Profile Page). He clicks on his username to modify it. Once he is satisfied of his new username, he clicks on the "Save" button to save his modifications. Now Zeno is happy.

3.3.7 S.7 Password Retrieval

To read the Functional Requirements for this scenario, refer to G.7 Functional Requirements Specifications

Archimedes is a Registered User and he wants to log in to the myTaxiService Application. Unfortunately, he doesn't remember his password, so he taps on the link in the login page (see Login Screen) and the myTaxiService system sends him an email with the password if the provided email is registered. Otherwise an error message is shown to Archimedes.

3.3.8 S.8 Reporting Abuses

To read the Functional Requirements for this scenario, refer to G.8 Functional Requirements Specifications

Crates is a Customer and he desperately wants to go to the cinema with his friends. He logs in to the myTaxiService Mobile Application and navigates to the homepage. Then he taps on the "Request a Taxi here" button and waits for a taxi (since he received the acceptance notification). But this taxi does not arrive and he can't reach the cinema in time. Crates is very frustrated so he reports the Taxi Driver that has accepted his taxi request.

Gorgias is a Taxi Driver and he has already logged in to the myTaxiService Mobile Application and signaled his availability. He receives a taxi request from a Customer and he accepts it. But when he goes to the meeting location, the Customer is not there. Gorgias has lost precious time and money so he is very hungry and reports the Customer through the application.

3.4 UML Models

3.4.1 Use-Case Diagrams

In the follow figure is explained how the Actor interact with the system and how the various use case interact with each other.

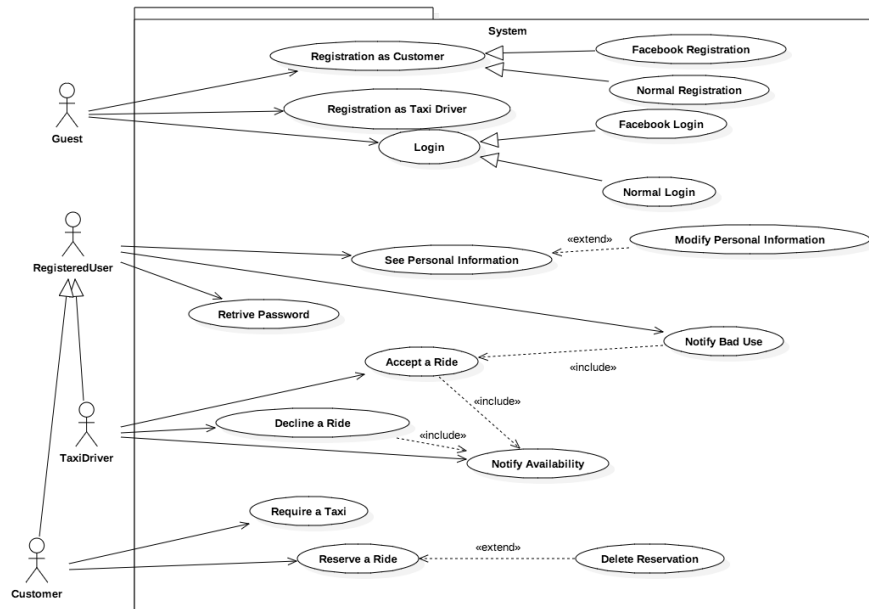


Figure 3.23: Use Case Diagram

3.4.2 Class Diagrams

This diagram explains a first proposal of the entities involved in this system and their relations. //TODO descrizione dopo alloy

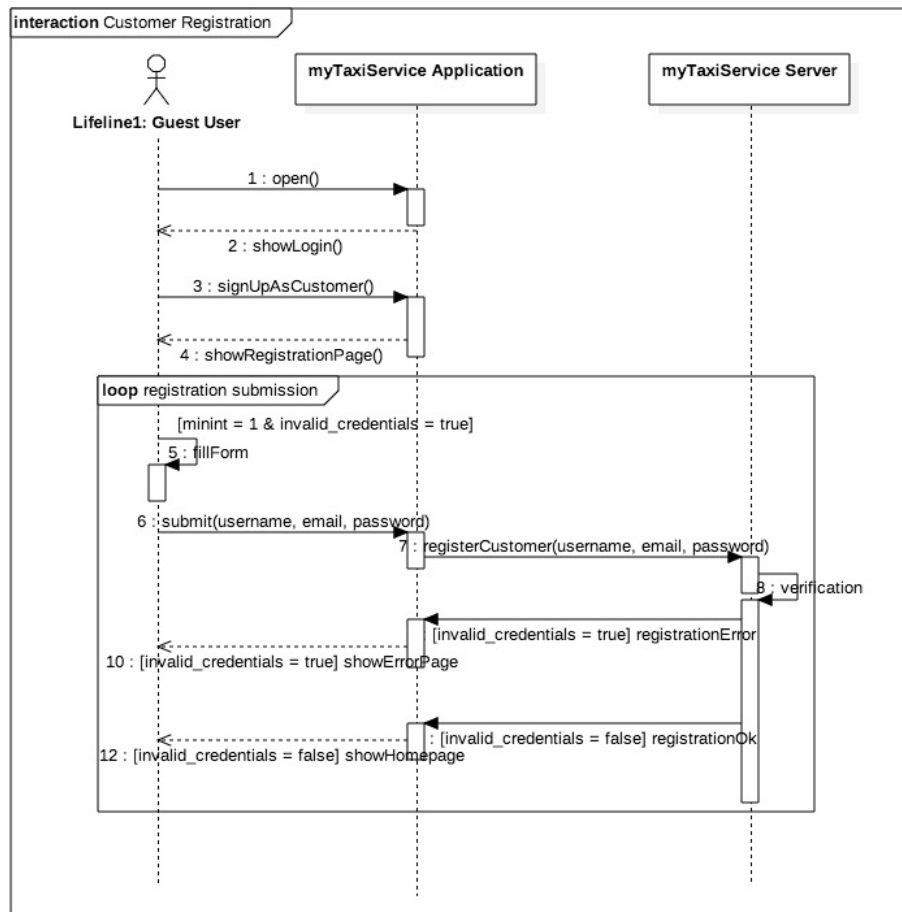


Figure 3.25: Customer Registration Interaction Diagram

3.4.2.2 Registration using a Facebook account.

To accompany this diagram, read the Scenario S.2.

Subject	Description
Actors	Guest User, myTaxiService Application, myTaxiService Server, Facebook Login System
Preconditions	Guest User must not be already registered
Execution	<ol style="list-style-type: none"> 1. Guest User open the myTaxiService Application. 2. Guest User taps on the "Log in with Facebook" button. 3. myTaxiService Application calls Facebook Login API 4. Facebook Login System makes his stuff 5. Facebook Login System reply 6. myTaxiService Application shows the result to the Guest User
Postconditions	The Guest User is now a Customer, he is registered in the database and he's now logged in.
Exceptions	1. The connection is lost or the Guest User doesn't complete the registration clicking the button.

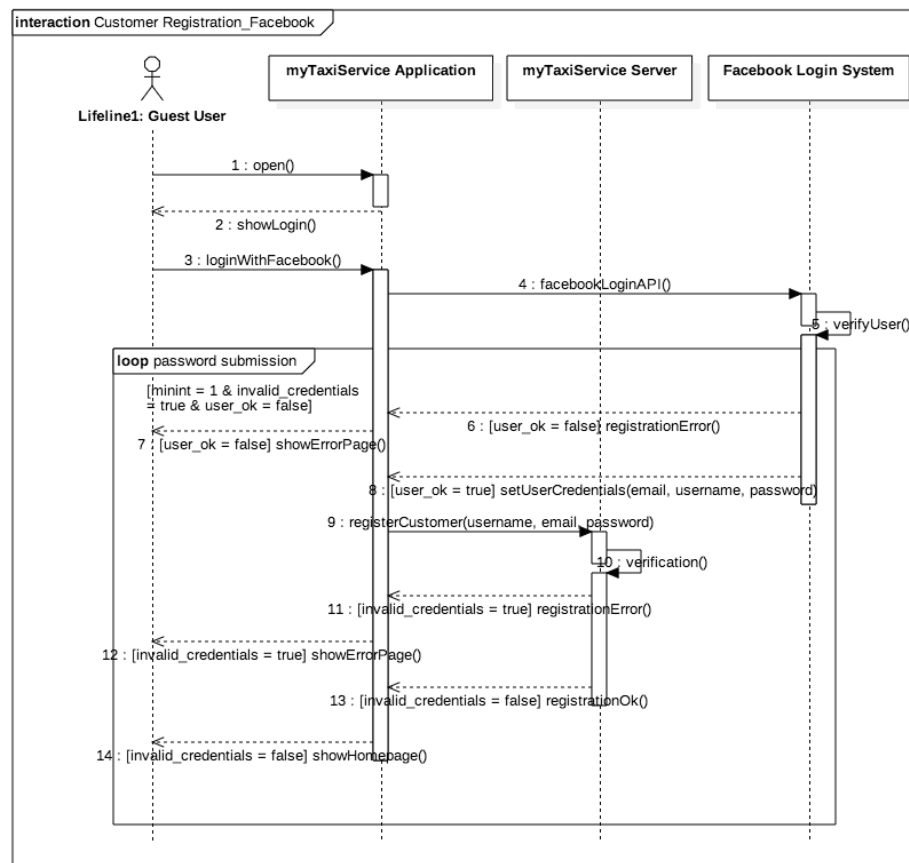


Figure 3.26: Facebook Customer Registration Interaction Diagram

3.4.2.3 Taxi Driver Registration

To accompany this diagram, read the Scenario S.3.

Subject	Description
Actors	Guest User, myTaxiService Web Application, myTaxiService Server
Preconditions	Guest User must not be already registered as a Taxi Driver
Execution	<ol style="list-style-type: none">1. Guest User open the myTaxiService Web Application.2. Guest User taps on the "Sign Up as a Taxi Driver" button.3. myTaxiService Web Application shows the registration page.4. Guest User fills up the registration form.5. Guest User taps on the "Submit" button.6. myTaxiService Web Application calls the server registration service.7. myTaxiService Server verify the credentials8. myTaxiService Server reply with an ok or an error9. myTaxiService Web Application shows to the Guest User the result of his registration.
Postconditions	The Guest User is now a Taxi Driver, he is registered in the database and he's now logged in.
Exceptions	<ol style="list-style-type: none">1. The email regex is not optimal (no regex for email is optimal)2. The connection is lost or the Guest User doesn't complete the registration clicking the button.

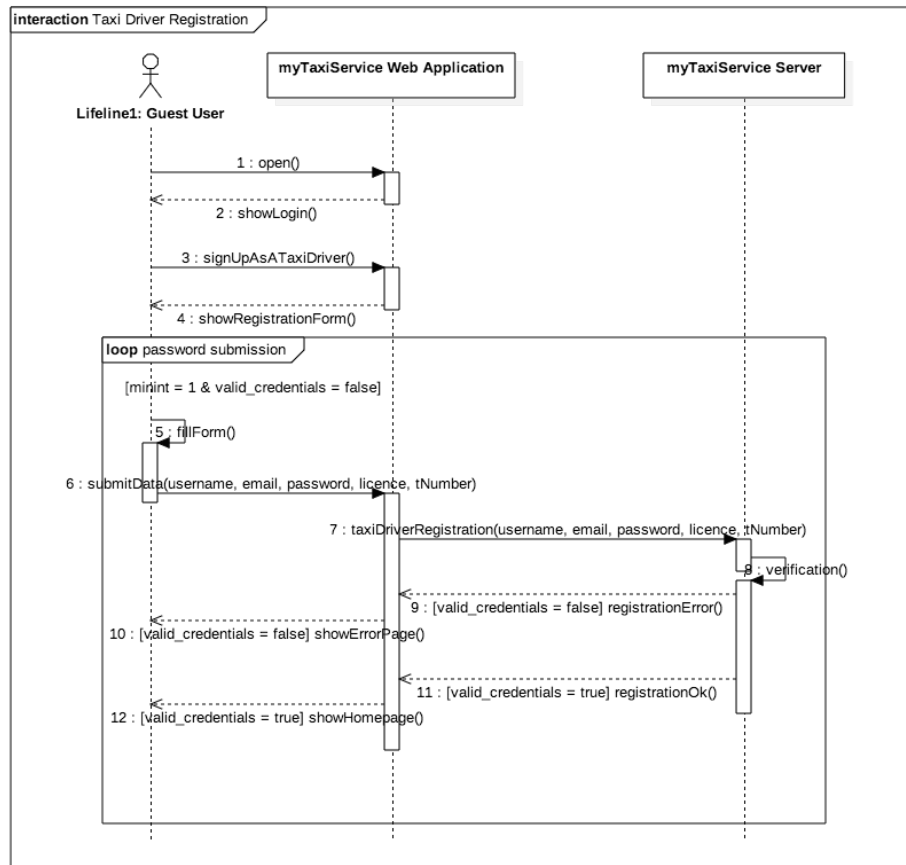


Figure 3.27: Taxi Driver Registration Interaction Diagram

3.4.2.4 Registered User Login

To accompany this diagram, read the Scenario S.4.

Subject	Description
Actors	Guest User, myTaxiService Web Application, myTaxiService Server
Preconditions	1. Registered User must be registered on the system. 2. Registered User must not be already logged in.
Execution	1. Registered User open the myTaxiService Web Application. 2. Registered User fills the login form. 3. Registered User press on the "Login" button. 4. myTaxiService Application submits the data to the server. 5. myTaxiService Server checks the data. 6. myTaxiService Server reply with an error or with success. 7. myTaxiService Application shows an error notification or a success one.
Postconditions	The Registered User is now logged in and is able to use the application.
Exceptions	1. The connection is lost or the Guest User doesn't complete the login clicking the button.

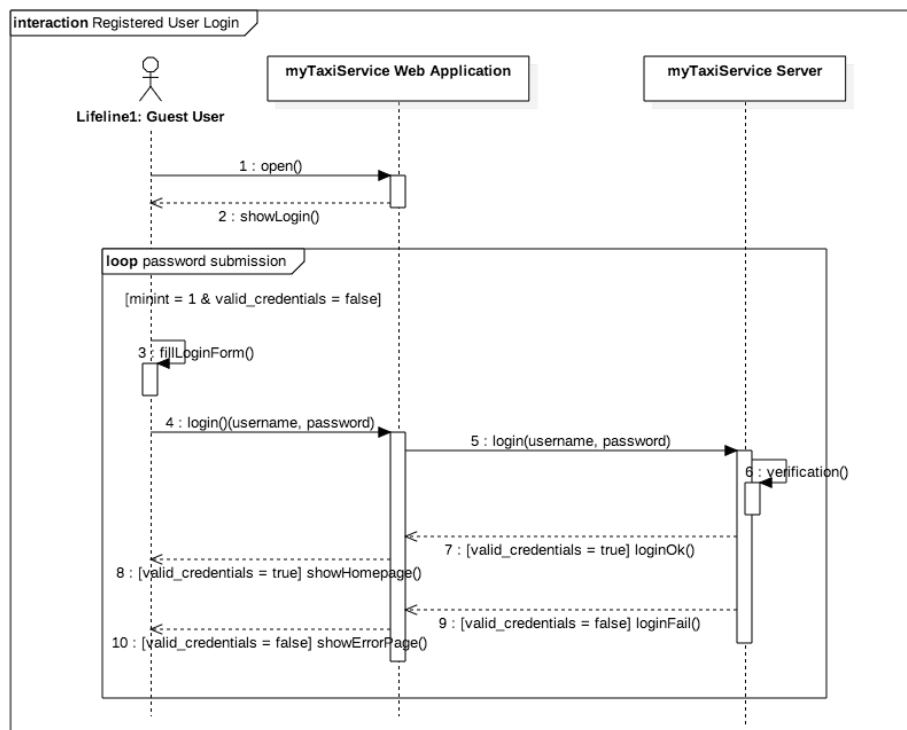


Figure 3.28: Login Interaction Diagram

3.4.2.5 Customer Facebook Login

To accompany this diagram, read the Scenario S.5.

Subject	Description
Actors	Customer, myTaxiService Mobile Application, myTaxiService-Server, Facebook System.
Preconditions	Registered User must have a Facebook account registered in the myTaxiService System.
Execution	1. Customer opens the myTaxiService Mobile Application. 2. myTaxiService Mobile Application shows the Login screen. 3. Customer clicks on the "Login with Facebook" button. 4. myTaxiService facebook login function is called and the homepage is showed to the Customer if the Facebook login is successful.
Postconditions	Customer is logged in.
Exceptions	1. The Customer disconnects before the homepage is showed.

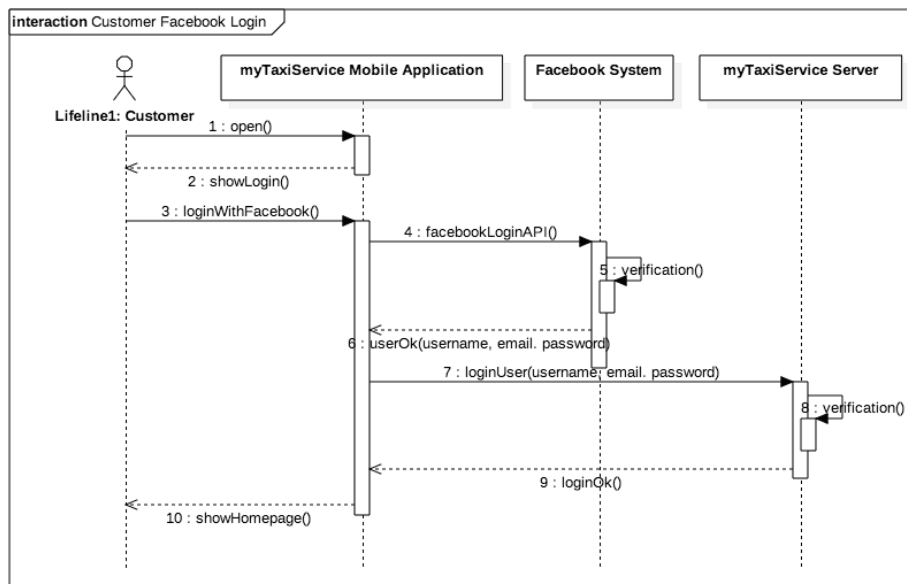


Figure 3.29: FacebookLogin Interaction Diagram

3.4.2.6 Profile Modification

To accompany this diagram, read the Scenario S.6.

Subject	Description
Actors	Registered User, myTaxiService Web Application, myTaxiService Server
Preconditions	Registered User must be logged in.
Execution	<ol style="list-style-type: none"> 1. Registered User open the myTaxiService Web Application. 2. Registered User taps on the "Me" button. 3. myTaxiService Web Application shows the profile page. 4. Registered User changes his credentials. 5. Registered User taps on the "Save" button. 6. myTaxiService Web Application calls the server modification function. 7. myTaxiService Server verify the credentials 8. myTaxiService Server reply with an ok or an error 9. myTaxiService Web Application shows to the Registered User the result.
Postconditions	Registered User's credentials are now changed.
Exceptions	1. The Registered User disconnects before saving.

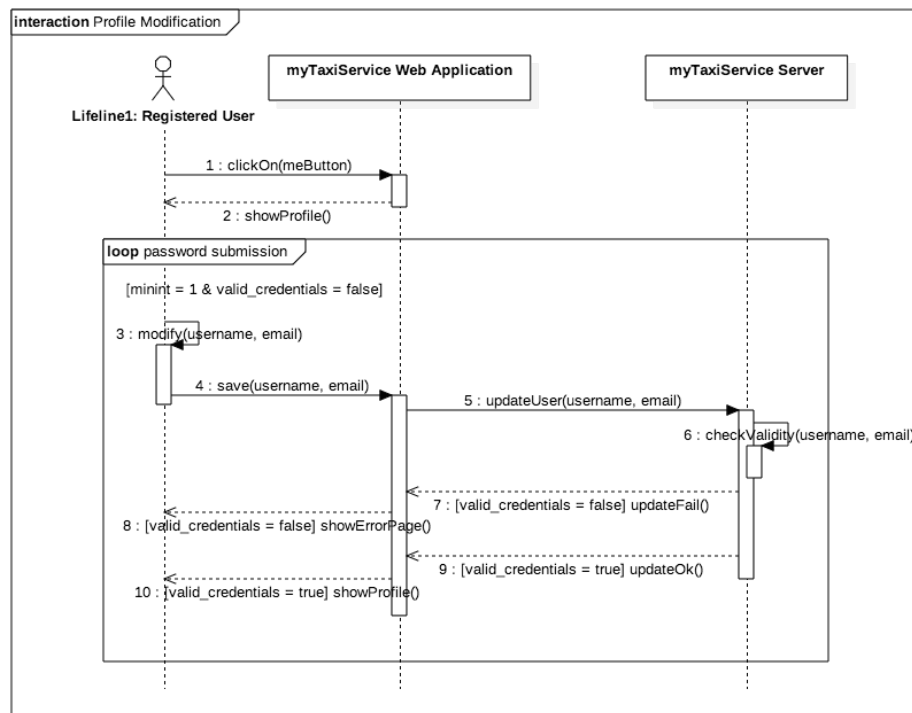


Figure 3.30: Profile Modification Interaction Diagram

3.4.2.7 Password Retrieval

To accompany this diagram, read the Scenario S.7.

Subject	Description
Actors	Registered User, myTaxiService Web Application, myTaxiService Server
Preconditions	Registered User must be actually registered.
Execution	<ol style="list-style-type: none">1. Registered User open the myTaxiService Web Application.2. Registered User taps on the "Retrieve Password" link.3. myTaxiService Web Application shows the email field.4. Registered User inserts his email.5. Registered User taps on the "Retrieve" button.6. myTaxiService Web Application calls the password retrieval function.7. myTaxiService Server verify the email.8. myTaxiService Server reply with an ok or an error9. myTaxiService Web Application shows to the Registered User the result.
Postconditions	The system has sent an email to the Registered User containing his password.
Exceptions	<ol style="list-style-type: none">1. The Registered User disconnects before clicking on the "Retrieve" button.

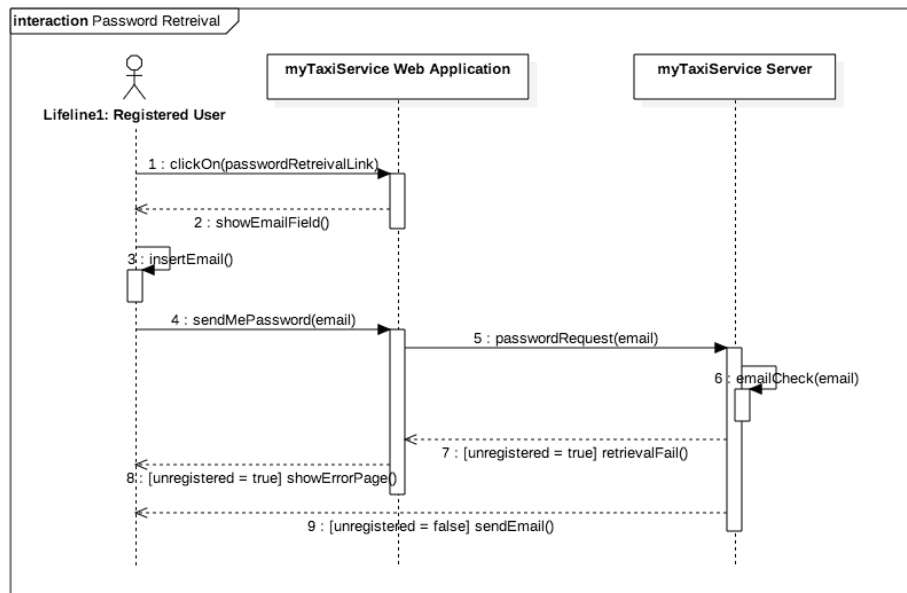


Figure 3.31: Password Retrieval Interaction Diagram

3.4.2.8 Taxi Driver Reporting

To accompany this diagram, read the Scenario S.8.

Subject	Description
Actors	Customer, myTaxiService Mobile Application, myTaxiService Server
Preconditions	Customer must be logged in.
Execution	1. Customer open the myTaxiService Mobile Application. 2. Customer taps on the "Request Taxi" button. 3. myTaxiService Mobile Application send the requests to the Server. 4. Customer waits for the acceptance. 5. myTaxiService Mobile Application shows the acceptance notification. 6. Customer waits for the Taxi that does not arrive. 7. Customer ends the ride reporting the Taxi Driver.
Postconditions	The ride is set to ended and the Taxi Driver is reported.
Exceptions	1. The Custoemr disconnects before reporting.

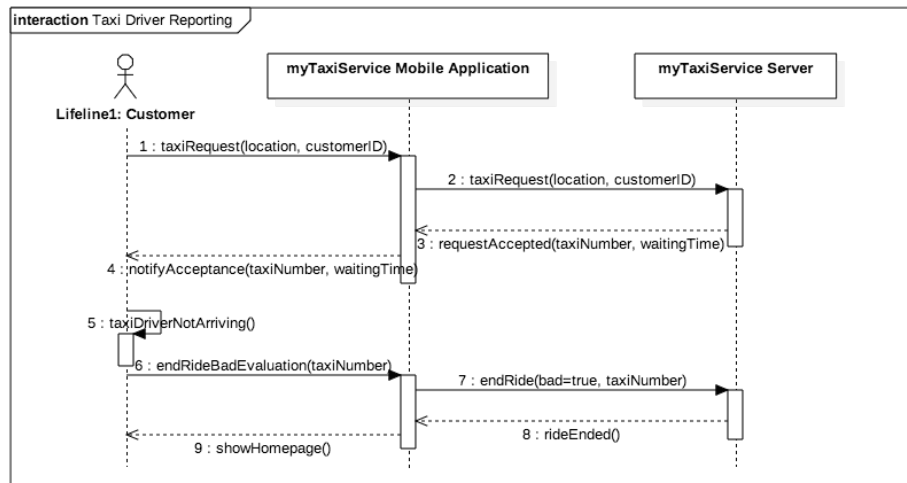


Figure 3.32: Taxi Driver Reporting Interaction Diagram

3.4.2.9 Customer Reporting

To accompany this diagram, read the Scenario S.8.

Subject	Description
Actors	Taxi Driver, myTaxiService Mobile Application, myTaxiService Server
Preconditions	Taxi Driver must be logged in.
Execution	1. Taxi Driver open the myTaxiService Mobile Application. 2. Taxi Driver receives a ride request notification from the Application. 3. Taxi Driver accepts the ride request. 4. Taxi Driver goes to the meeting location and the Customer is not there. 5. Taxi Driver ends the ride reporting an abuse.
Postconditions	The ride is set to ended and the Customer is reported.
Exceptions	1. The Taxi Driver disconnects before reporting.

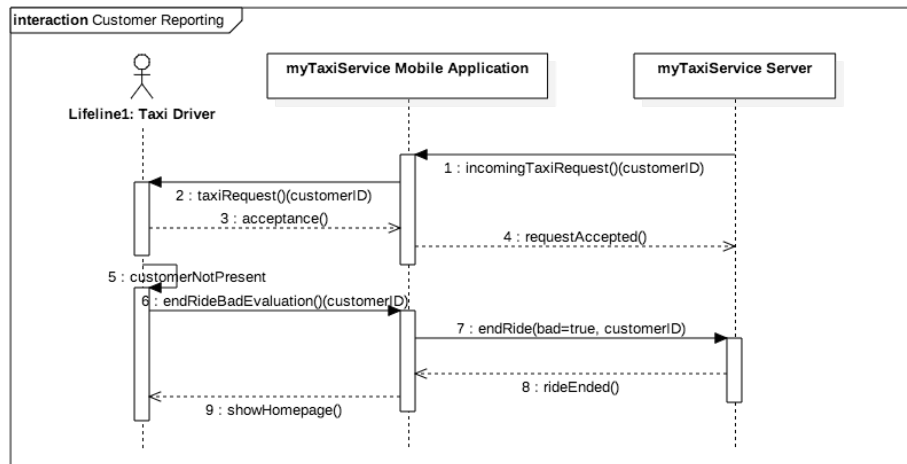


Figure 3.33: Customer Reporting Interaction Diagram

3.4.3 State Machine Diagrams

3.5 Non Functional Requirements

3.5.1 Performance Requirements

3.5.2 Design Constraints

3.5.3 Software System Attributes

3.5.3.1 Availability

3.5.3.2 Maintainability

3.5.3.3 Portability

3.5.4 Security

3.5.4.1 External Interface Side

3.5.4.2 Application Side

3.5.4.3 Server Side

Chapter 4

Appendix

//TODO