



Politecnico di Milano  
Academic Year 2015/2016  
Software Engineering 2: “myTaxiService”  
Design Document

Massimo Schiavo, Marco Edoardo Cittar

December 1, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Scope . . . . .	4
1.3	Definitions, acronyms, abbreviations . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Acronyms . . . . .	4
1.3.3	Abbreviations . . . . .	4
1.4	Reference documents . . . . .	4
1.5	Document structure . . . . .	4
<b>2</b>	<b>Architectural design</b>	<b>4</b>
2.1	Overview . . . . .	4
2.2	High level components and their interaction . . . . .	4
2.3	Component view . . . . .	4
2.4	Deployment view . . . . .	4
2.5	Runtime view . . . . .	4
2.5.1	Registration . . . . .	5
2.5.2	Log in . . . . .	6
2.5.3	Queue management . . . . .	7
2.5.4	Simple request . . . . .	8
2.5.5	Detailed request . . . . .	9
2.5.6	View free taxis . . . . .	10
2.6	Component interfaces . . . . .	10
2.7	Selected architectural styles and patterns . . . . .	10
2.8	Other design decisions . . . . .	10
<b>3</b>	<b>Algorithm design</b>	<b>10</b>
3.1	Registration . . . . .	10
3.2	Log in . . . . .	11
3.3	Simple request . . . . .	11
3.4	Detailed request . . . . .	11
3.5	View free taxis . . . . .	11
<b>4</b>	<b>User interface design</b>	<b>12</b>
4.1	Log in page . . . . .	12
4.2	Registration page . . . . .	13
4.3	Confirmation page . . . . .	14
4.4	User page . . . . .	15
4.5	Simple Request . . . . .	16
4.6	Detailed Request . . . . .	17
4.7	Request confirmation page . . . . .	18
4.8	Available taxis page . . . . .	19
4.9	User notifications page . . . . .	20
4.10	Driver page . . . . .	21

4.11 Requests page . . . . .	22
<b>5 Requirements traceability</b>	<b>23</b>
<b>6 References</b>	<b>23</b>

# **1 Introduction**

## **1.1 Purpose**

## **1.2 Scope**

## **1.3 Definitions, acronyms, abbreviations**

### **1.3.1 Definitions**

### **1.3.2 Acronyms**

- DD: Design Document

### **1.3.3 Abbreviations**

## **1.4 Reference documents**

## **1.5 Document structure**

# **2 Architectural design**

## **2.1 Overview**

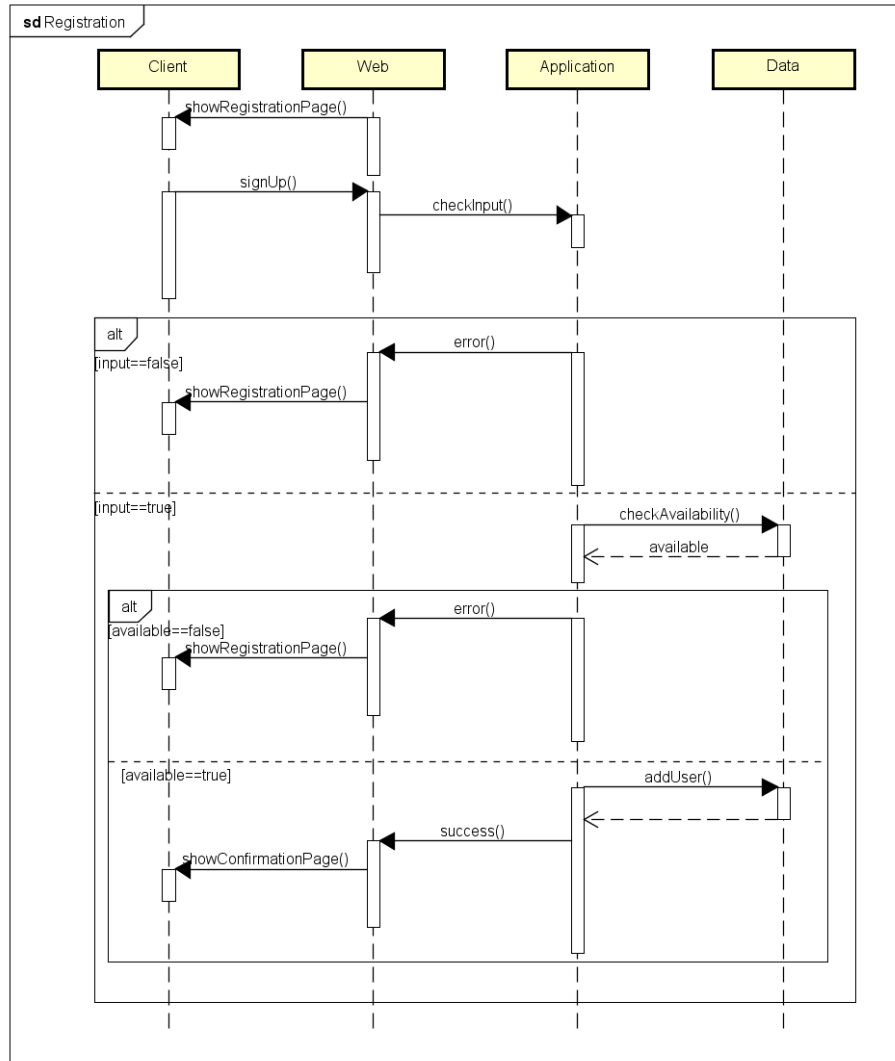
## **2.2 High level components and their interaction**

## **2.3 Component view**

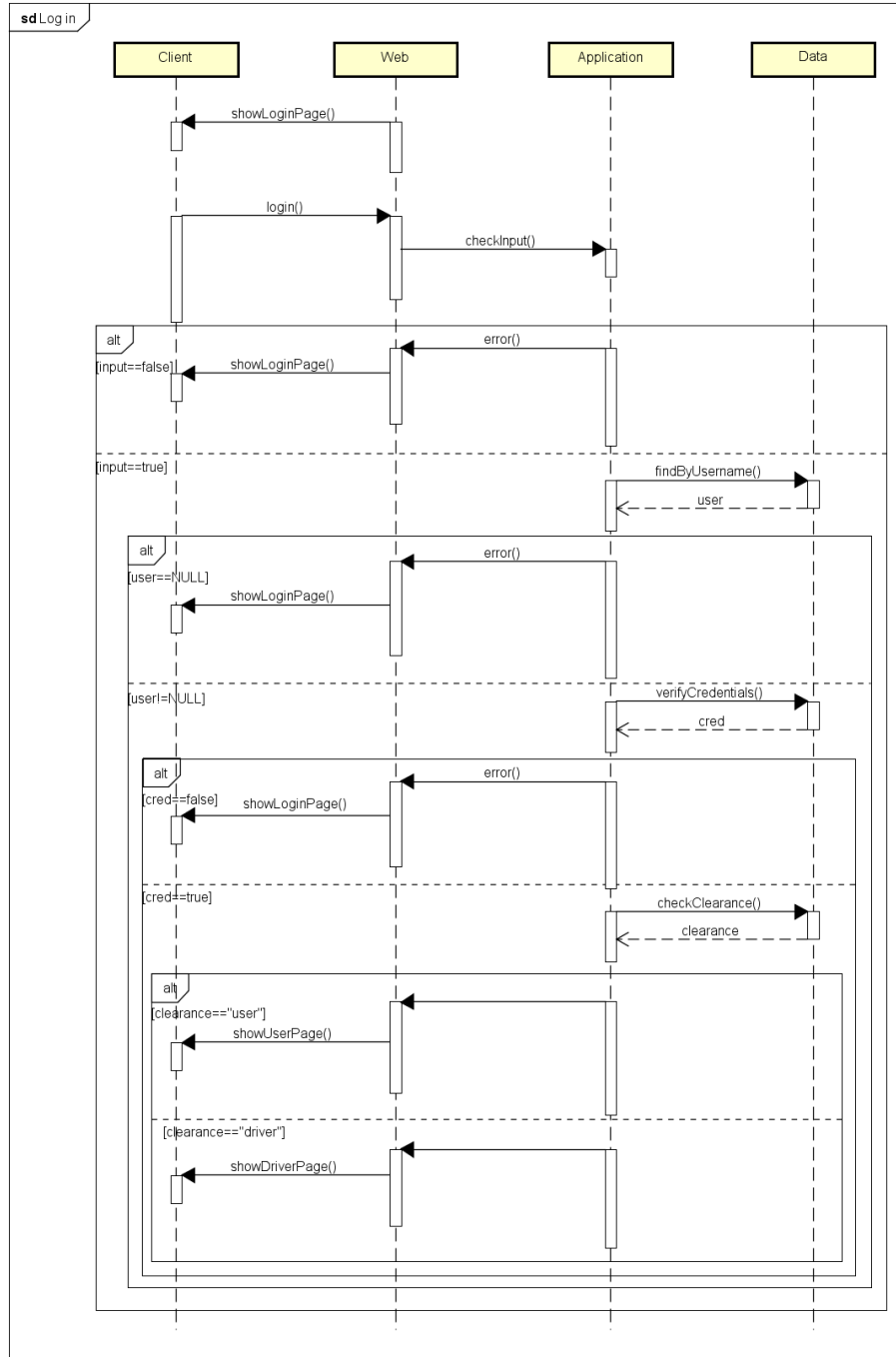
## **2.4 Deployment view**

## **2.5 Runtime view**

### 2.5.1 Registration

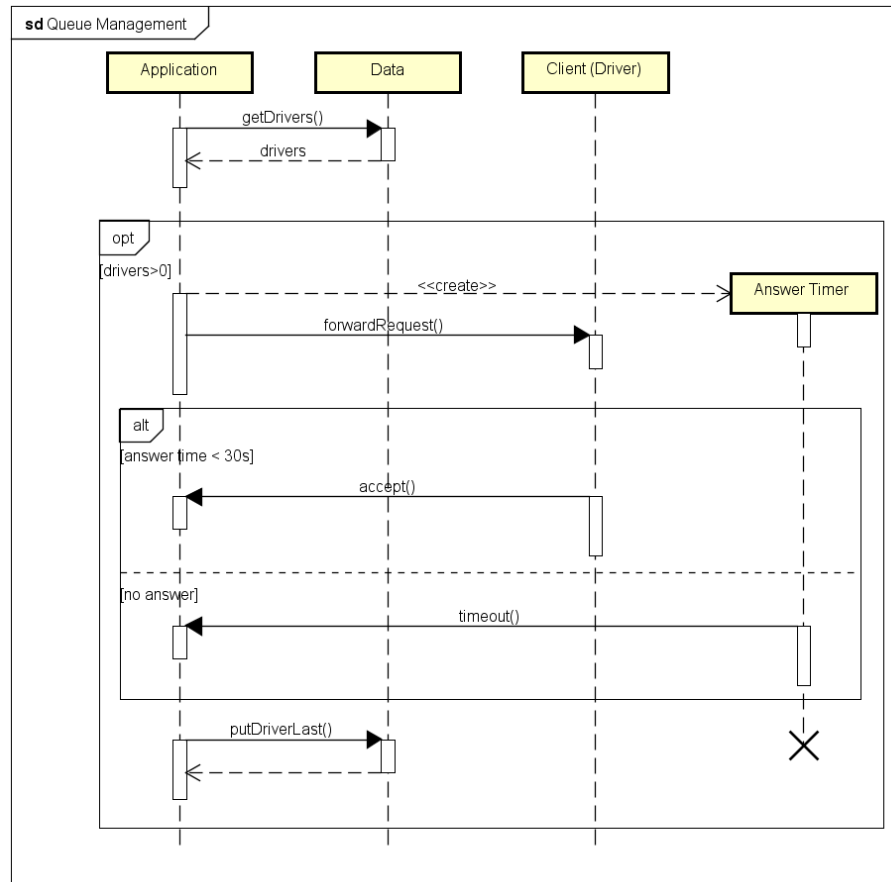


## 2.5.2 Log in

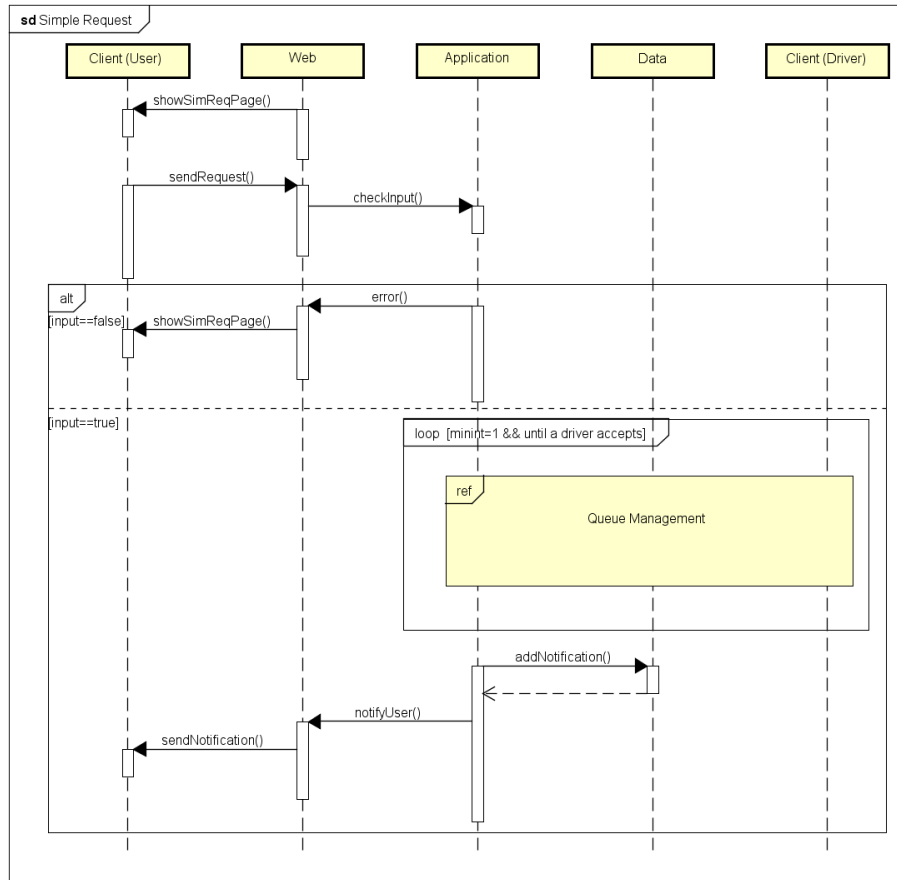


### 2.5.3 Queue management

This sequence diagram represents how the application manage to forward a request to a driver.

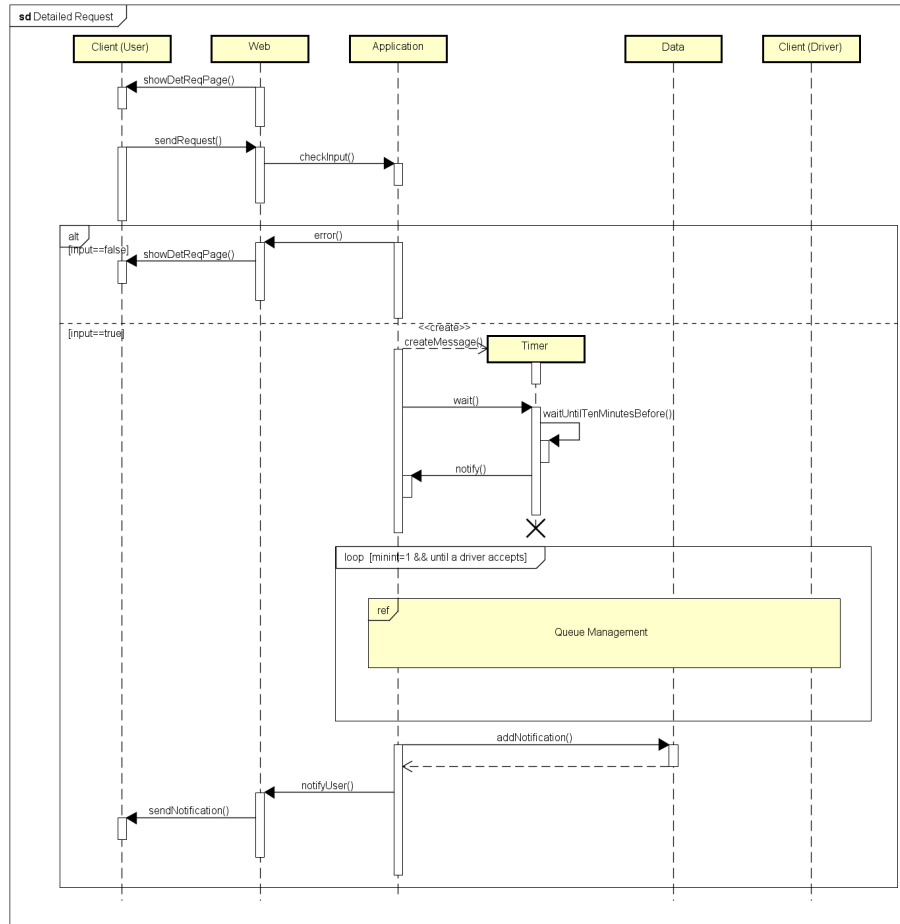


### 2.5.4 Simple request

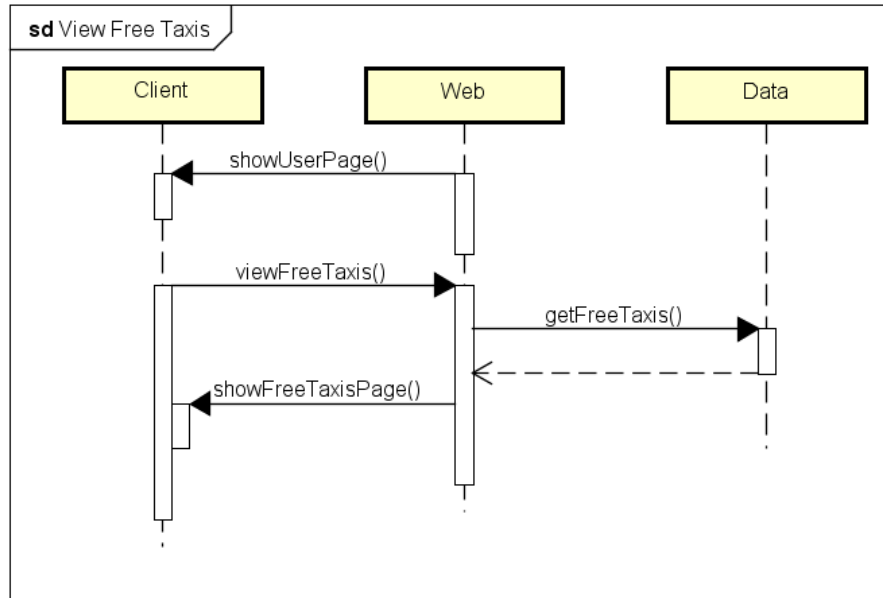




## 2.5.5 Detailed request



### 2.5.6 View free taxis



## 2.6 Component interfaces

## 2.7 Selected architectural styles and patterns

## 2.8 Other design decisions

# 3 Algorithm design

## 3.1 Registration

The algorithm takes in account the completeness, the correctness and the availability of the information inserted.

After the guest submits the information the application checks if all the mandatory fields are filled and if the “confirmation” fields, such as password and email, correspond to the respective previous ones.

If one of the above doesn’t apply, the application shows again the registration page to the guest. If they do, then it looks up into the database to verify that the username and email inserted are not already in use.

If they don’t, then the user is added to the database and is shown a confirmation page, inviting him to log in., otherwise the registration page is shown again.

### 3.2 Log in

After the guest inputs his credentials, the application first checks if both fields are filled, then looks for the username in the database to see if it exists. If it does, it checks if the password matches too.

If all of these steps are verified, the guest is logged in and is shown his personal page, otherwise the application shows the login page again.

After the password verification, the application will check the privileges of the account that is about to log in and will consequently show the respective page, be it user's or driver's.

### 3.3 Simple request

After the user clicks on the "Simple request" button, he is shown the simple request page.

After he inserts the origin of his ride and press the "Request" button, the application will then begin to allocate the first available taxi, otherwise if there's none it will just wait until one becomes available.

After forwarding the request to the taxi, the system will wait 30 seconds, if the driver answers the call he will be in charge for that request, or else the system forwards the request to the next available taxi in queue. In both cases the driver will be put at the end of the queue.

Once the taxi is allocated, the user is informed with a notification containing the code of the taxi and the waiting time.

### 3.4 Detailed request

After the user click on the "Detailed request" button, he is shown the detailed request page.

The user must fill the form and click on the "Request" button. The application will then check if all the fields are filled and if time and date are formally correct. If that's so, the system will wait until 10 minutes before the ride and then begin to allocate a taxi for the request.

The following taxi allocation, queue management and notification works as the one in the simple request.

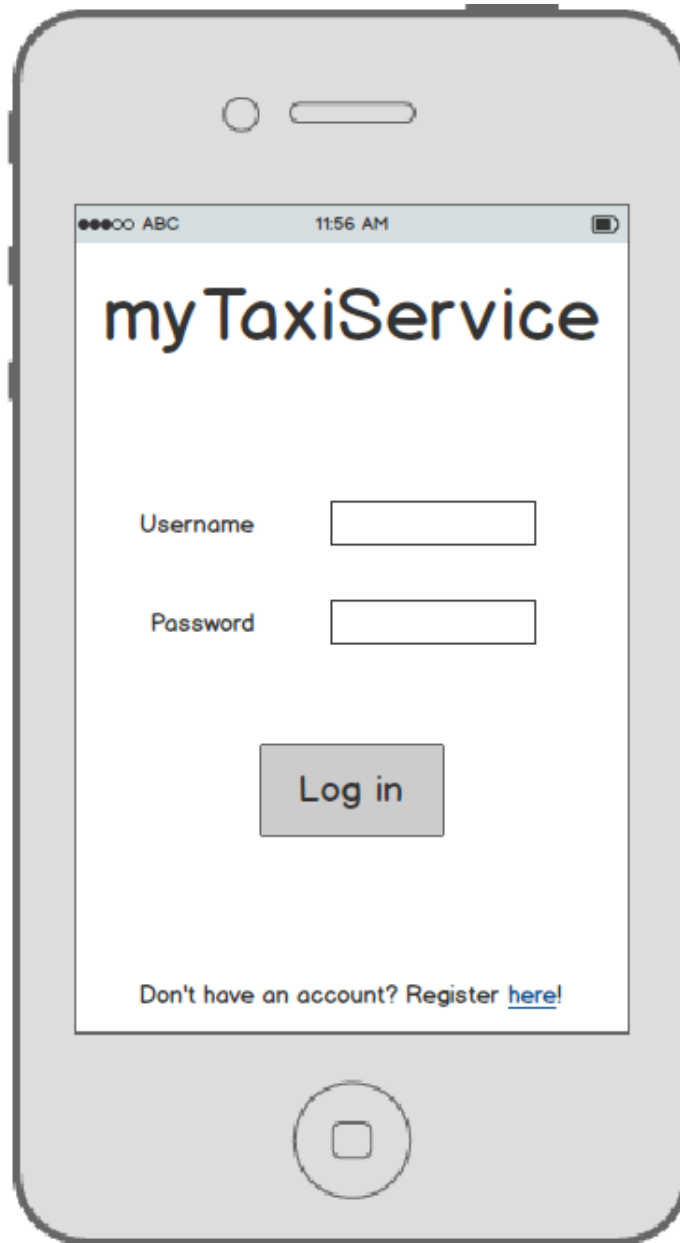
### 3.5 View free taxis

The users can look at the number of available taxis in their zone. By clicking on the "View available taxis" button, the application will query the database about it, and then return the result to a new page.

## 4 User interface design

### 4.1 Log in page

This is the page that shows up when the app is opened. From here guests can log in or register themselves if they don't have an account yet.



## 4.2 Registration page

Here guests can register themselves by filling all the mandatory fields (the ones shown).

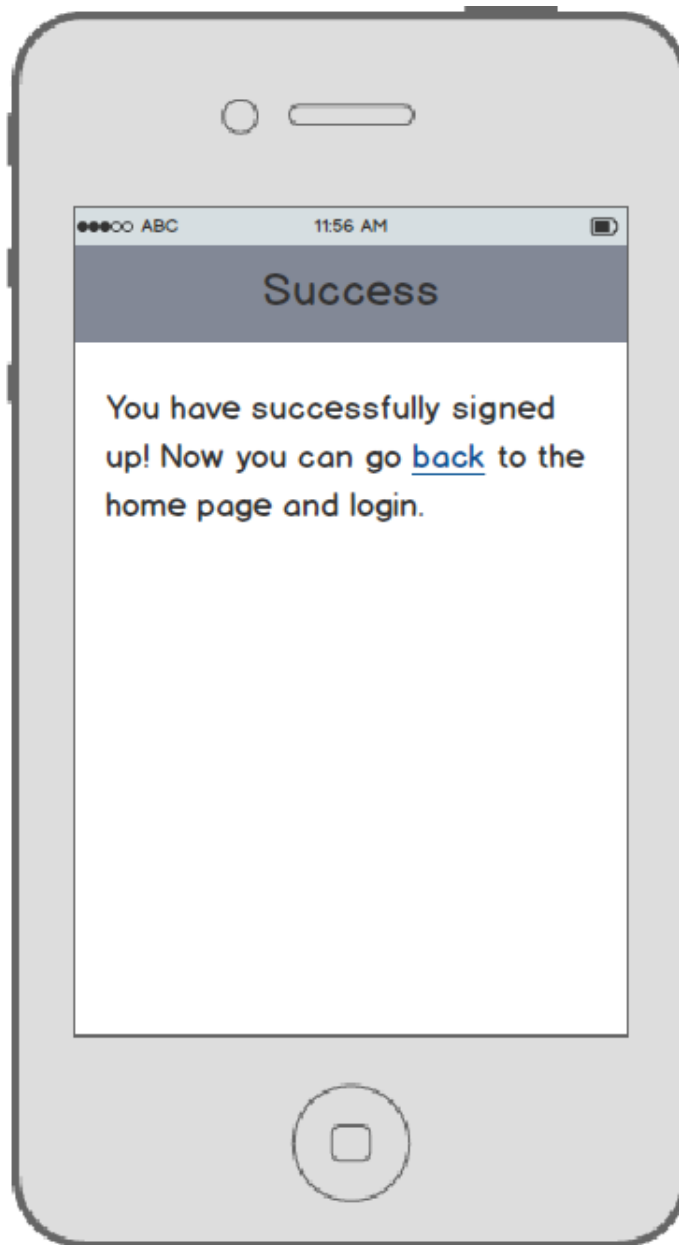
The image shows a mobile phone screen with a registration form. The phone has a grey bezel and a circular home button at the bottom. The screen displays a status bar at the top with signal strength, 'ABC', and the time '11:56 AM'. Below the status bar is a dark grey header with a 'Login' button and the title 'Registration'. The main content area is white and contains a form with the following fields and labels:

- Name
- Surname
- Username
- Password
- Confirm password
- Email
- Confirm email

At the bottom of the form is a large grey button labeled 'Sign up'.

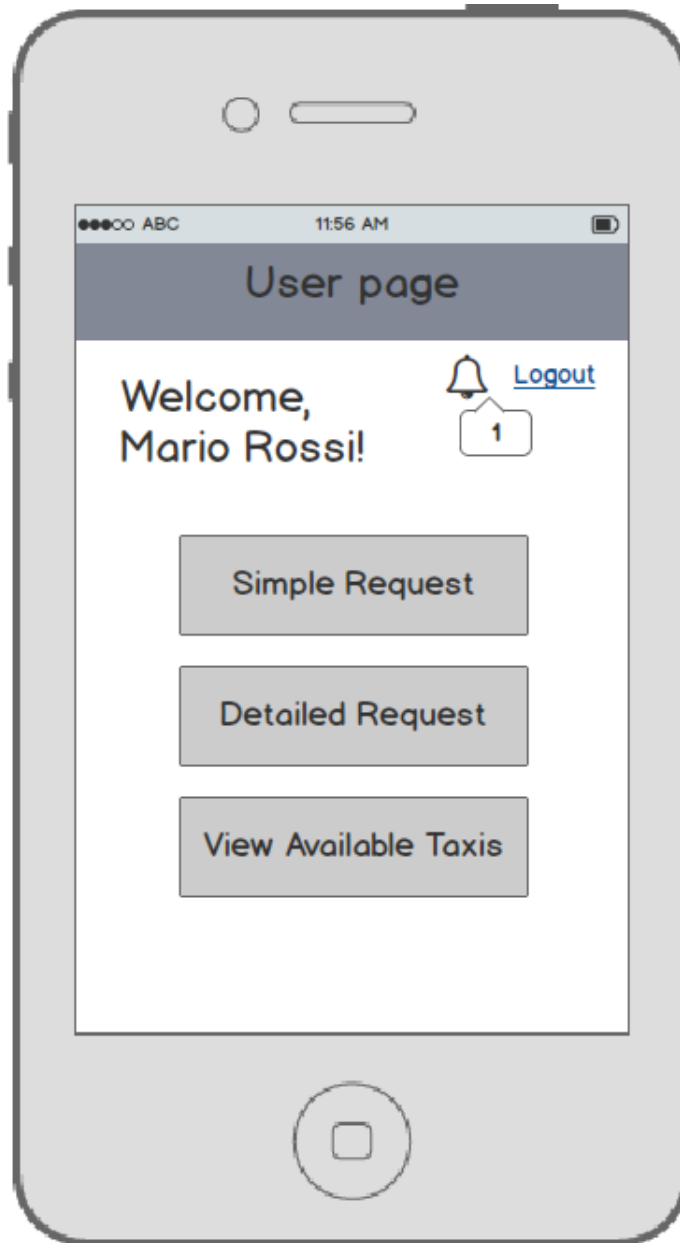
### 4.3 Confirmation page

If the registration was successful this page shows up.



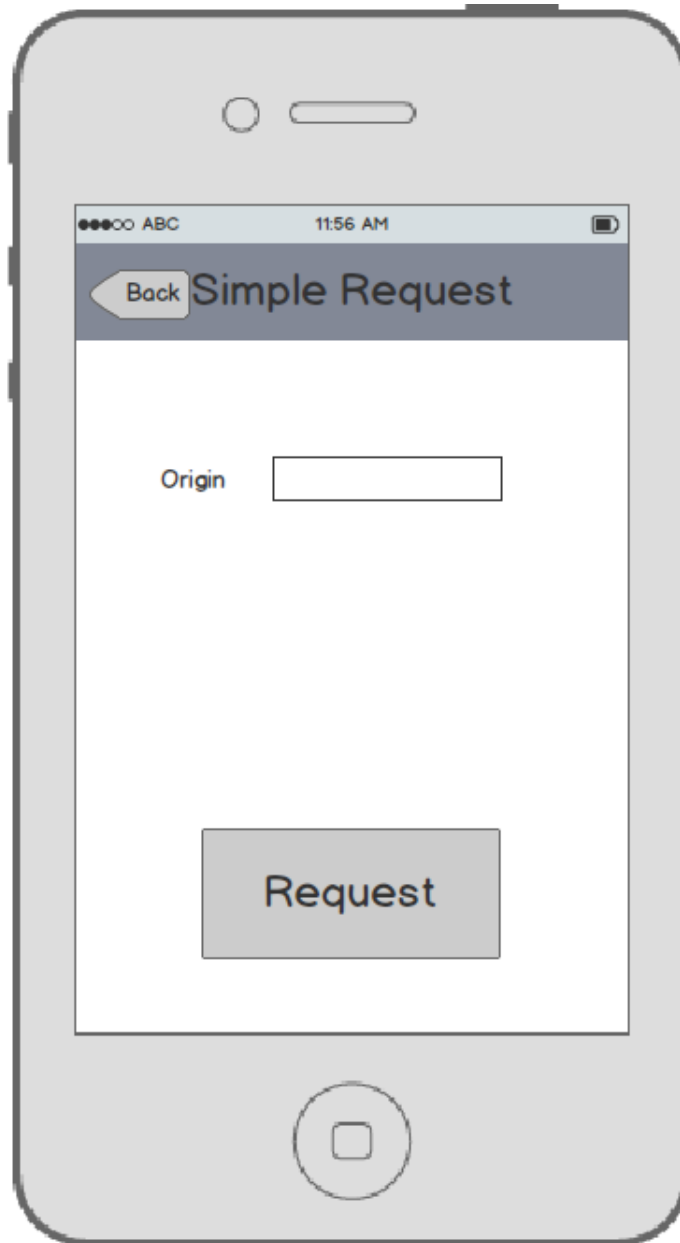
#### 4.4 User page

After logging in, users see this page from which they can use all of their functionalities: request a ride, see available taxis in their area and read notifications.



## 4.5 Simple Request

Users can request a ride by inserting its origin, that is where the driver should pick them up.







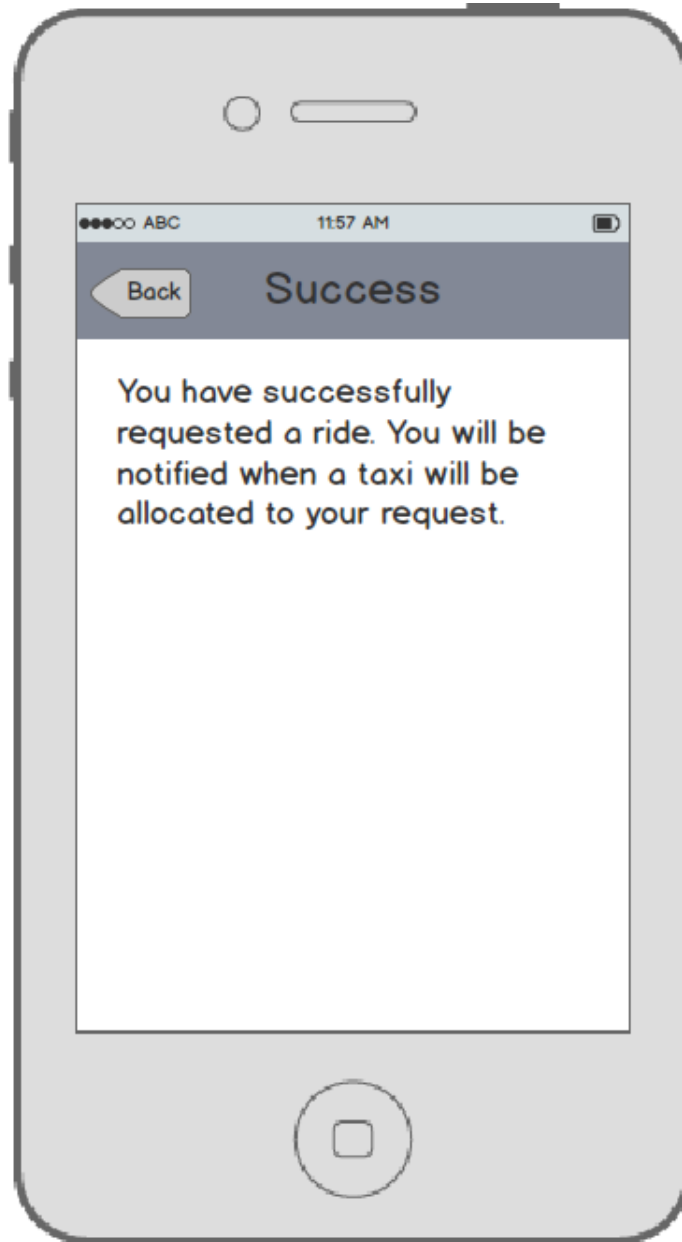
## 4.6 Detailed Request

Users can make a detailed request by filling this form.

The image shows a mobile application interface for a 'Detailed Request' form. The screen is framed by a grey border representing the phone's bezel. At the top, a status bar displays signal strength, the network name 'ABC', and the time '11:56 AM'. Below the status bar is a dark grey header with a 'Back' button and the title 'Detailed Request'. The main content area is white and contains four input fields: 'Origin', 'Destination', 'Date', and 'Time'. The 'Date' field includes a calendar icon, and the 'Time' field includes a clock icon. A large grey button labeled 'Request' is positioned at the bottom of the form.

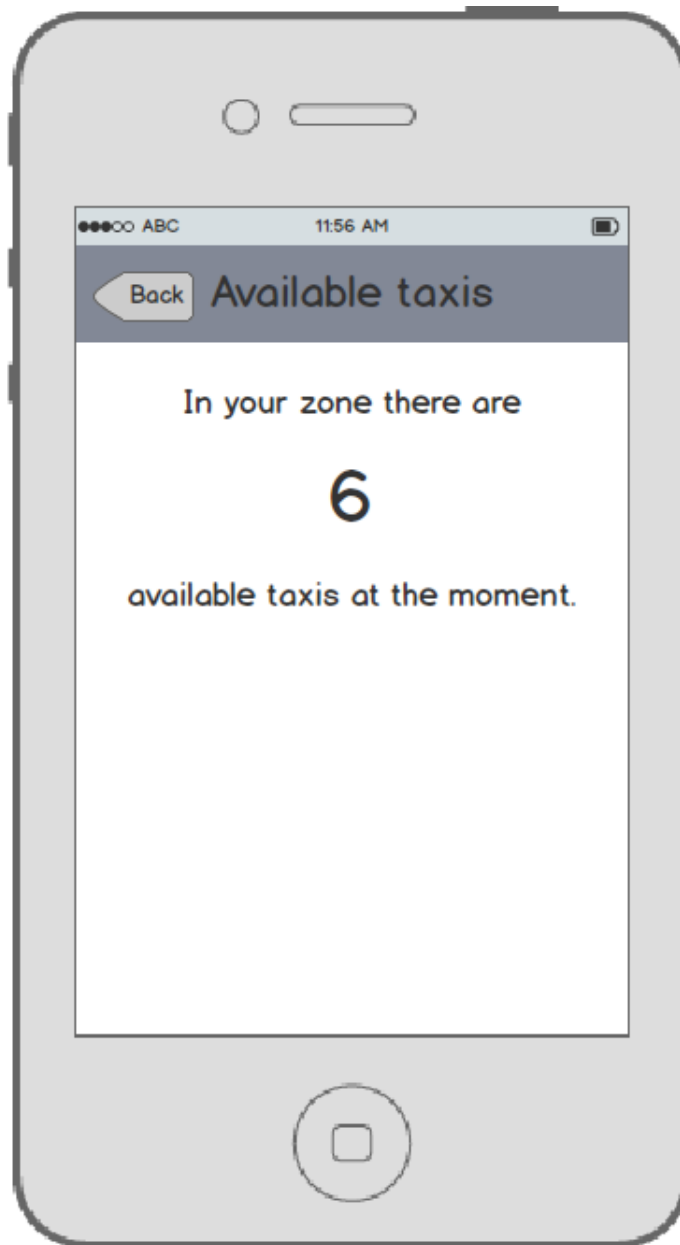
Origin	<input type="text"/>
Destination	<input type="text"/>
Date	<input type="text" value="DD/MM/YYYY"/> 
Time	<input type="text" value="HH:MM"/> 

#### 4.7 Request confirmation page



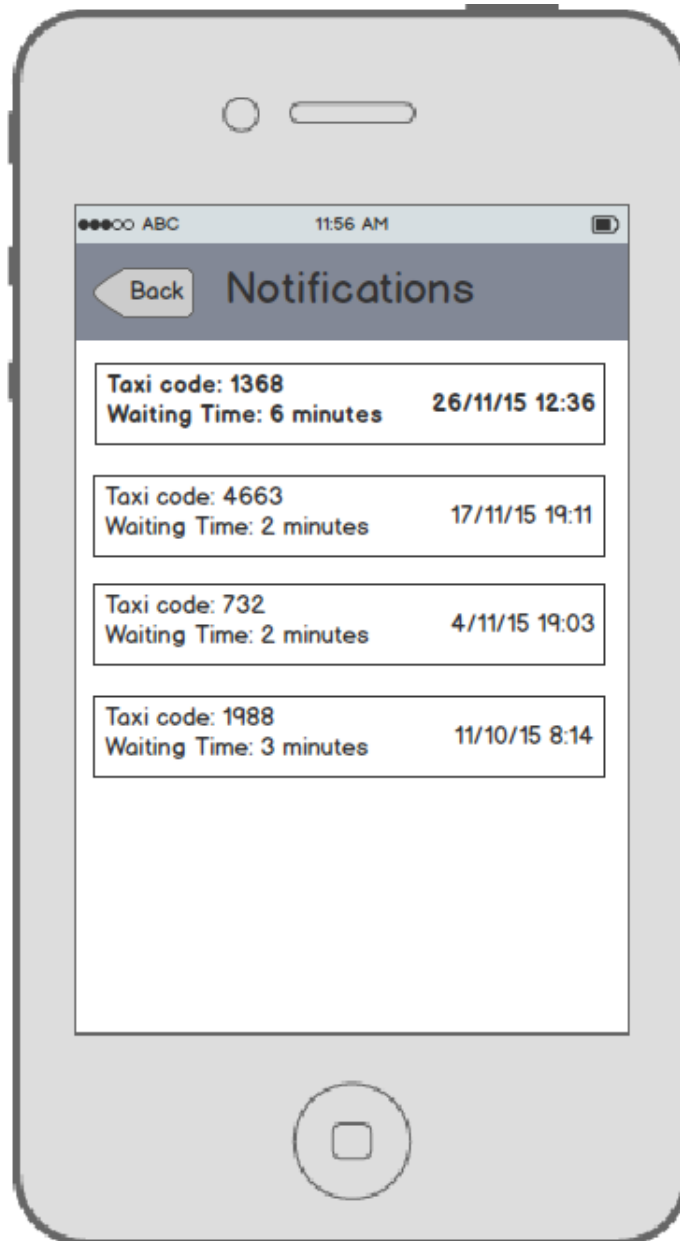
#### 4.8 Available taxis page

This page shows the number of available taxis in the area that the user is in.



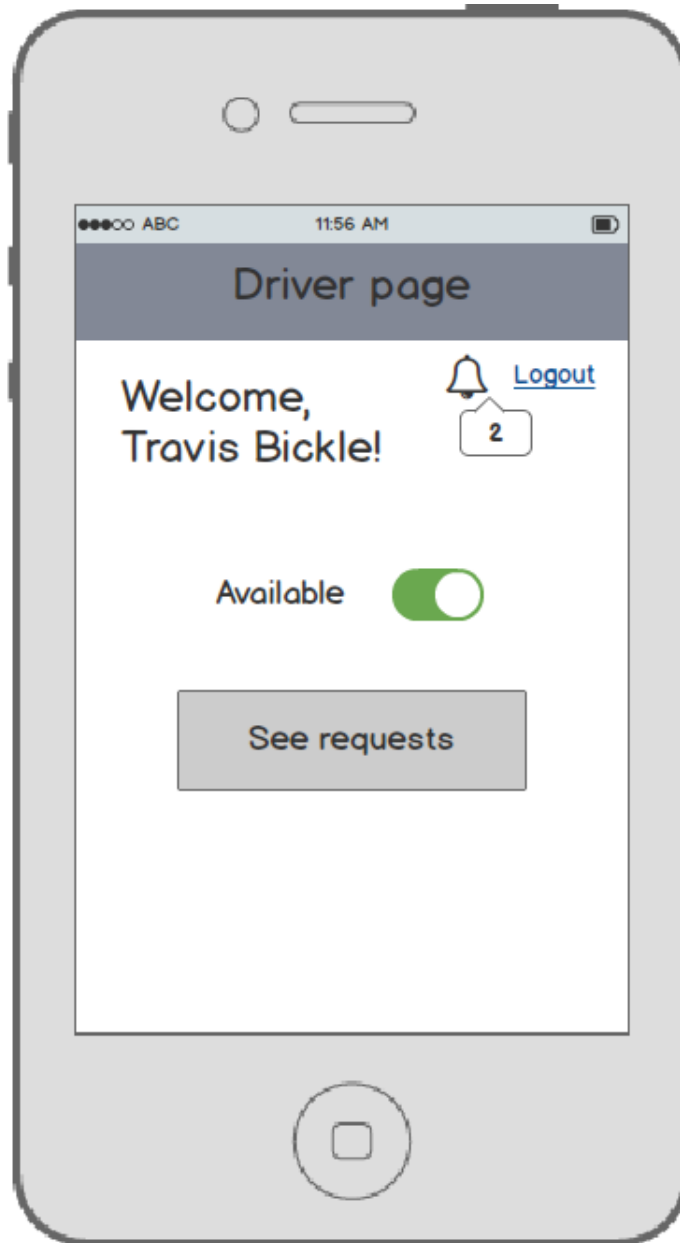
## 4.9 User notifications page

Here users can see notifications about the requests they made. The new ones that hasn't been read yet are written in bold.



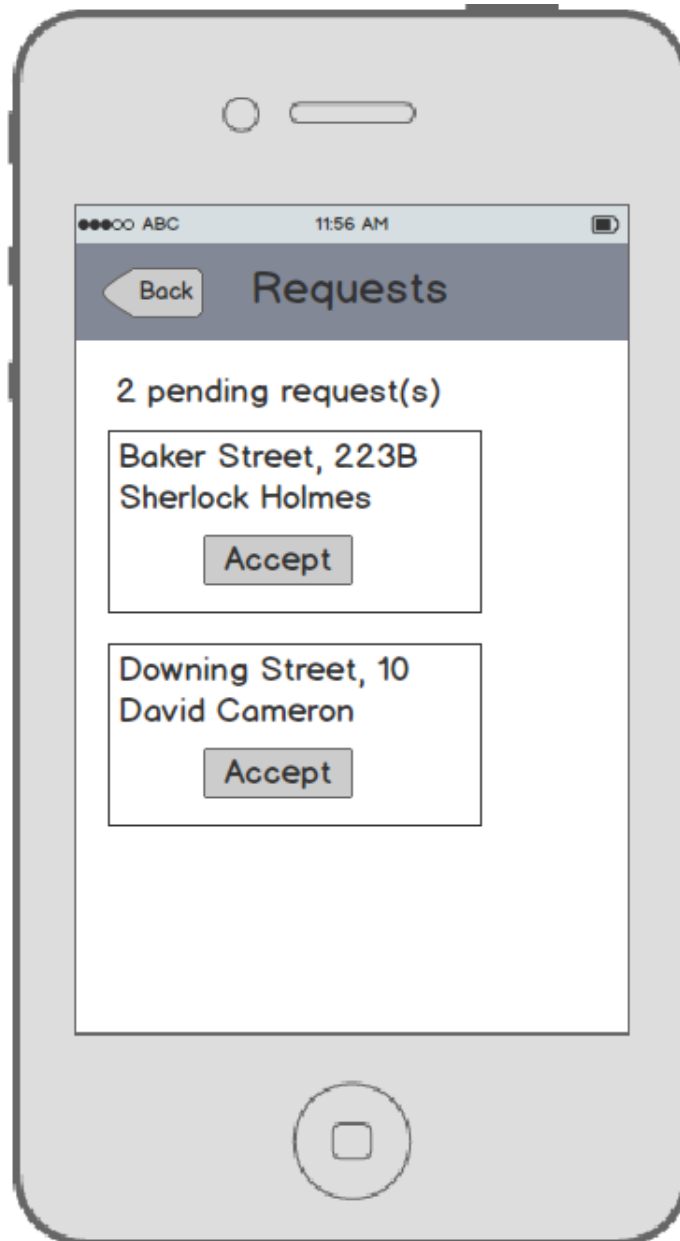
#### 4.10 Driver page

This is the page that driver see after they log in. They can set themselves as available and see incoming requests.



#### 4.11 Requests page

In this page drivers can see pending requests. If a driver is first in line, he can receive multiple requests but obviously can only accept one.



**5 Requirements traceability**

**6 References**