Politecnico Di Milano



A.A. 2015/2016

Software Engineering 2:

”MyTaxiService”

Requirements Analysis and Specifications Document

Paramithiotti Andrea (Matr. 788794) Rompani Andrea (Matr. 854052)

Zoia Lorenzo (Matr. 852392)

Contents

[1 Introduction 3](#_Toc434411297)

[1.1 Purpose 3](#_Toc434411298)

[1.2 Scope 3](#_Toc434411299)

[1.3 Definitions, acronyms, and abbreviations 4](#_Toc434411300)

[1.4 Actors 4](#_Toc434411301)

[1.5 Goals 4](#_Toc434411302)

[1.6 Document Overview 4](#_Toc434411303)

[2 Overall Description 4](#_Toc434411304)

[2.1 Product perspective 5](#_Toc434411305)

[2.1.1 System interfaces 6](#_Toc434411306)

[2.1.2 User interfaces 7](#_Toc434411307)

[2.1.3 Communication interfaces 7](#_Toc434411308)

[2.1.4 Operations 8](#_Toc434411309)

[2.2 Product functions 8](#_Toc434411310)

[2.3 User Characteristics 9](#_Toc434411311)

[2.4 Constraints 9](#_Toc434411312)

[2.5 Assumptions and dependencies 9](#_Toc434411313)

[3 Specific Requirements 10](#_Toc434411314)

[3.1 External Interfaces Requirements 10](#_Toc434411315)

[3.1.1 User Interfaces 10](#_Toc434411316)

# Introduction

## Purpose

This document represents the Requirement Analysis and Specification Document (RASD). The main goal of this document is to completely describe the system in terms of functional and non-functional requirements found by analysing the real needs of the customer.

In addition, this document outlines the constraints and the limits of the software and describes the typical use cases that will occur after the deployment.

This document is intended for all the developers and for the system analysts that need to integrate other systems with ours. It also could be used as a contractual basis between the customer and the developers.

## Scope

The aim of the project is to create a brand new system for the management and organization of a city taxi service. This system offers a mobile application and a web interface in order to give the customer the possibility to benefit from the taxi service. Furthermore, the system provides an additional communication interface for the taxi driver.

The mobile application and the web interface accept requests and reservations for taxis from the users, with the possibility to organise a share ride among different users. The taxi driver is supposed to communicate his availability, acceptances and rejections of requests through the communication interface.

The system is created to simplify the access of passengers to the service, and to guarantee a fair management of taxi queues.

## Definitions, acronyms, and abbreviations

## Actors

1. Person: A generic person who is neither registered nor authenticated.
2. Registered User: A person who is registered to the service.
3. Authenticated User: A person who is authenticated to the service.
4. Passenger: A person who is currently on a taxi ride.
5. Taxi Driver: The person who drives a taxi.
6. System: The software that manages the application interaction between all the actors.

## Goals

1. Allow a user to register and authenticate to the service
2. Allow the user to request a taxi. Inform the taxi driver of the request and the position of the passenger.
3. Manage the taxis in order to fulfil all the requests.
4. Allow the user to reserve a ride at a specific date and time.
5. Allow some users to share a ride on the same route and the related cost.

## Document Overview

# Overall Description

## Product perspective

The system is composed of:

1. A server that handles the requests from the mobile application
2. A data storage server used to store the needed data for the system to work properly
3. A web server that handles the requests from the web interface
4. An Internet interface used to link the system to the internet or dedicated network
5. Taxi drivers server that handles the taxi drivers requests
6. Mobile application interface: used to access the system services by an enabled mobile device
7. Web interface: used to access the system services by an enabled web browser
8. Taxi terminal: a device that sends to the system the taxi location and also acts at the driver’s system interface

Diagram of the system showing its different parts and how they are connected.

### System interfaces

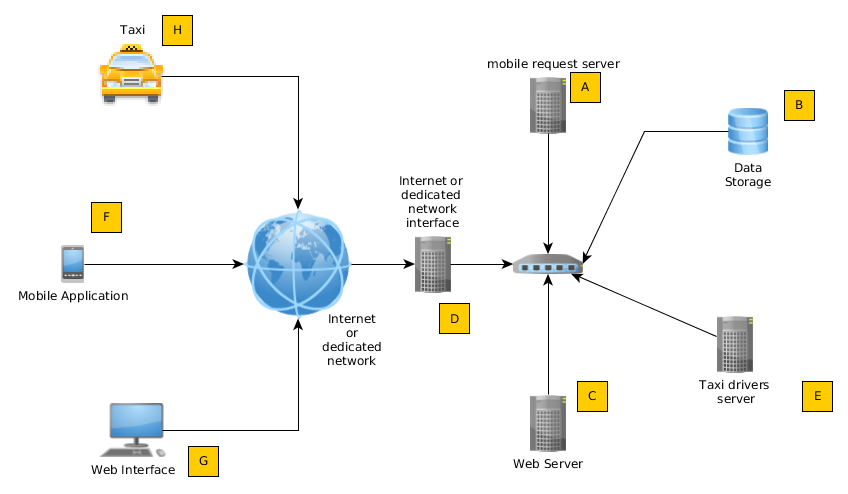
The system provides these interfaces:

1. Taxi driver interface that:
   1. Sends the current location of the taxi to the system
   2. Accepts the commands concerning the taxi driver normal operations
2. Mobile interface that:
   1. Enables the user registration to the service
   2. Enables the user authentication for the service access
   3. Allows the user to request a taxi for a ride
   4. Allows the user to reserve a taxi ride on a future date
   5. Allows the user to engage in a shared taxi ride
3. Web interface that:
   1. Enables the user registration to the service
   2. Enables the user authentication for the service access
   3. Allows the user to request a taxi for a ride
   4. Allows the user to reserve a taxi ride on a future date
   5. Allows the user to engage in a shared taxi ride
4. Internet or dedicated connection interface that:
   1. Establishes the link from the mobile and web interfaces to the respective servers
   2. Establishes the link from the taxi driver interfaces to the taxi drivers server

The system also provides a set of API to enable the expansion of the functionalities in the future

### User interfaces

The mobile interface:

1. Must enable the user to insert string values and also, by using a map, to choose a location
2. Must notify any error in an informative way, also showing the user a way to solve the error
3. Any person with any school education level must be able use the mobile application services without any training

The web interface

1. Must be compliant with html 5 specifications and be readable and usable in any standardized resolution
2. Must notify any error in an informative way, also showing the user a way to resolve the error
3. Any person with any school education level must be able use the web interface services without any training

The Taxi driver interface:

1. To be able to use the driver’s specific services, the taxi driver’s interfaces must be easy to use also while driving (where highway code permits it) and while the taxi is still.
2. It must be able to send the position of the taxi to the system, manually or automatically
3. Any taxi driver must learn every function in at most 1 hour of training

### Communication interfaces

In order to support the integration between the various parts of the system every protocol must use only standardized protocols to enable future modernization without compromising current operations

### Operations

1. The application is always interactive, except for the operation of localization of the taxi
2. The backup of the data must be done daily with the risk of losing at most 24 hours

## Product functions

User Registration

User Authentication

Request

Reservation

Initiate a Shared Ride

Add a shared ride passenger

Communication of the arrival and payment

## User Characteristics

* Any technical expertise must not be necessary in order to access and use the mobile application and the web interface.
* Every person with a primary school education level must be able to execute every action concerning the use of the operations in the taxi service.
* The taxi drivers need to be instructed in order to be able to use their specific functionalities.

## Constraints

* Any request, reservation, and shared ride cannot be performed by a non authenticated user.
* Any unauthorized person cannot use the taxi driver specific functions.
* Every taxi driver must be authenticated and authorized by the system administrator.
* The system must always respond to a user request (positively or negatively).
* No downtime due to an internal system failure that is greater than 1% of the total time (of 1 year) is accepted.
* Only a downtime (due to an internal error) smaller than the 1% of the total amount of working hours is accepted.
* Two different users cannot simultaneously authenticate to the system with the same account.

## Assumptions and dependencies

* The mobile application must be executed and tested on Android, IOS and Windows Phone platforms.
* The web interface must function correctly and be tested on any HTML 5 compliant browser.
* The number of taxies must be minus or equal of the number of the zone multiplied for the maximum length of a taxies queue.
* The number of taxies must be at least equal at the number of the zones.

# Specific Requirements

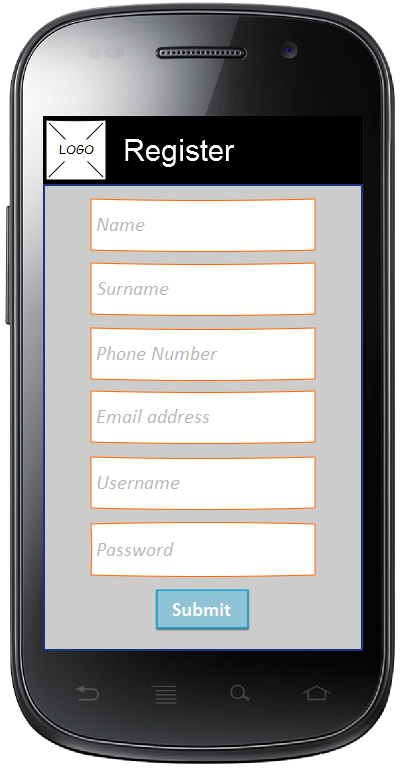
## External Interfaces Requirements

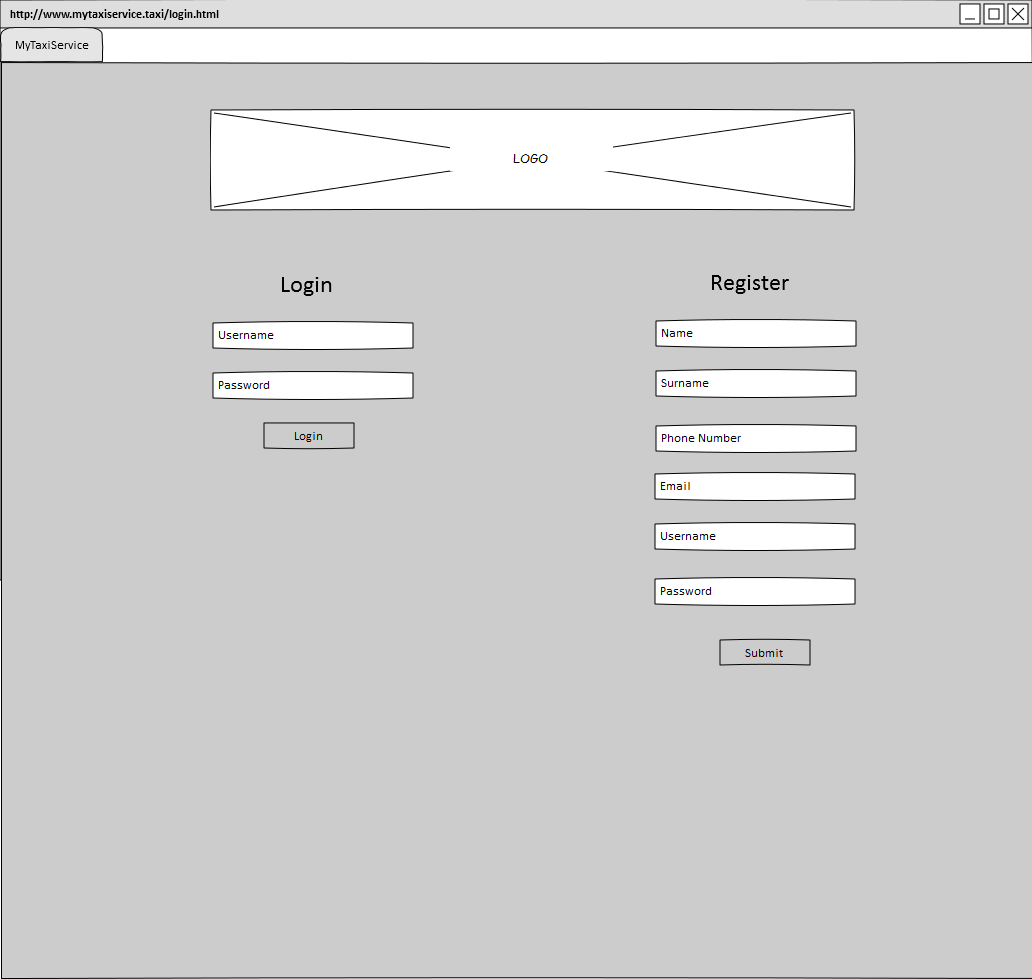
### User Interfaces

The mockups show the pages for the interaction with the users, through the web application or the mobile application.

#### Registration and Authentication

These mockups show the home page of the web application and the pages of the mobile application that allow a new user to register to the service or a registered user to authenticate.

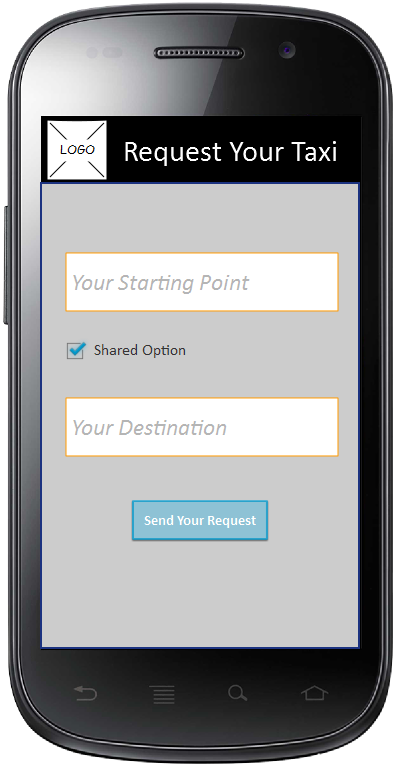
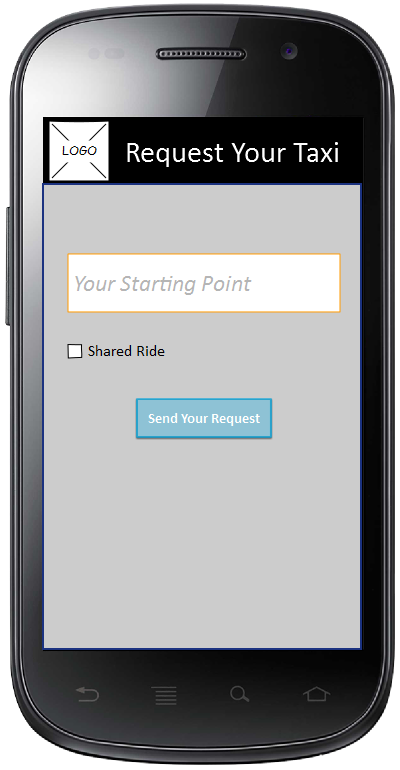


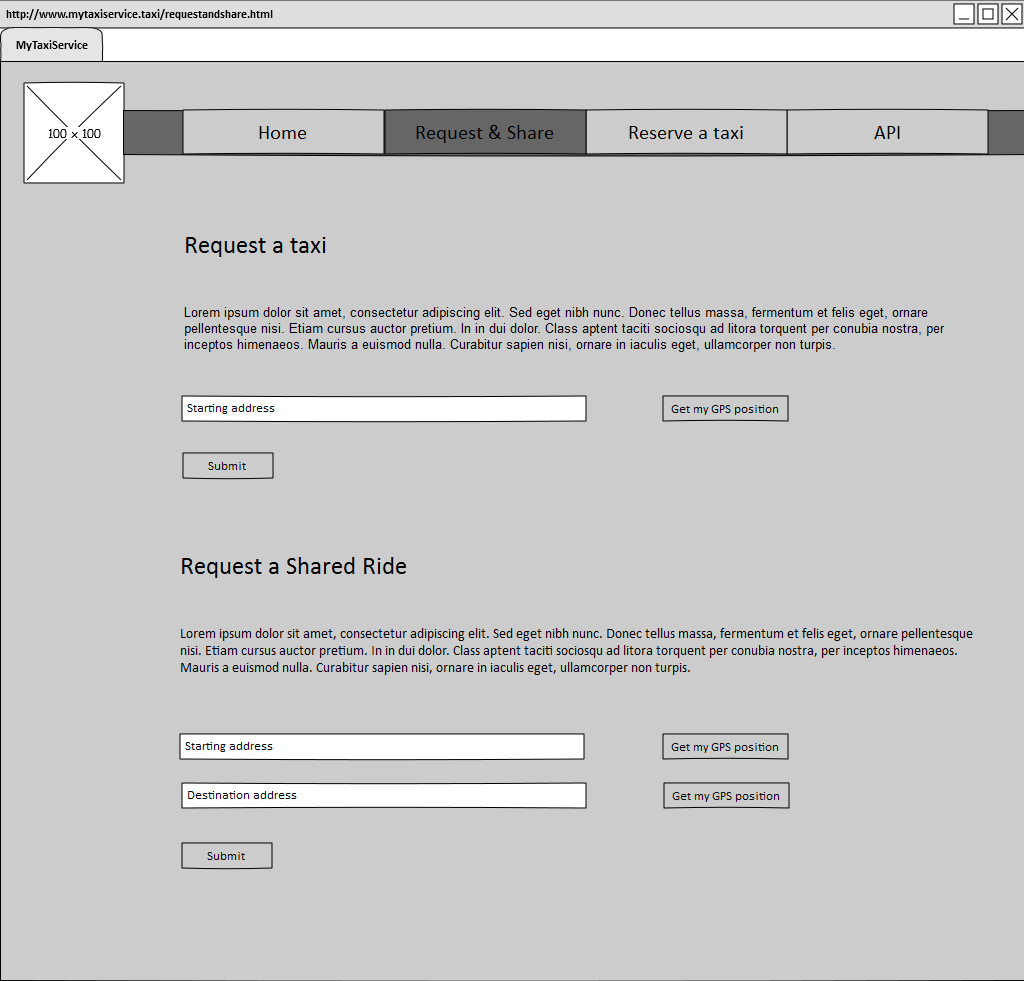


#### Request a taxi or a Shared ride

The mockups show the interfaces that allow the users to request a normal ride or a shared ride.

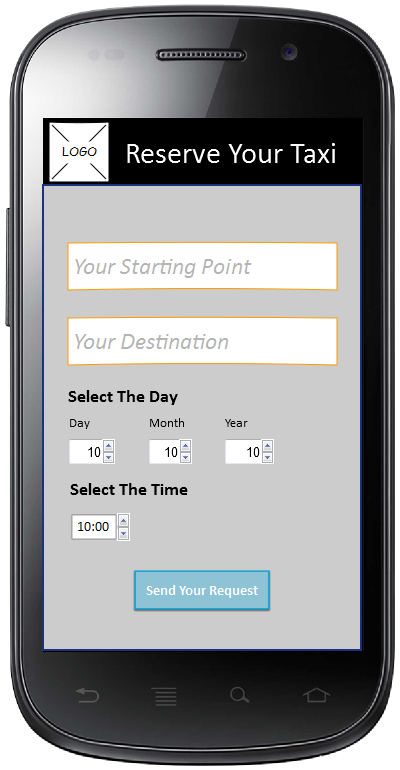
The two options are on the same page in the web application, while in the mobile application the user must select a flag in order to make the shared ride options appear.

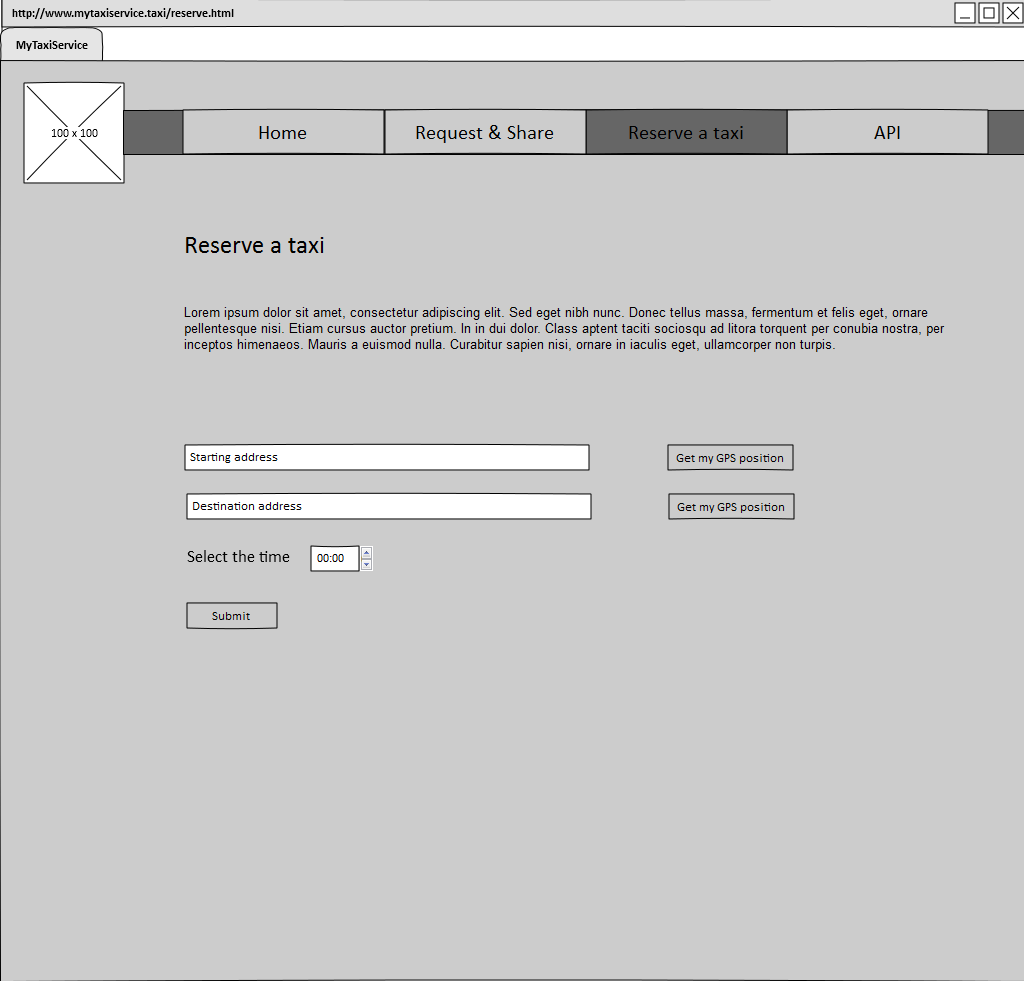




#### Reserve a taxi

The mockups show the interfaces that allow the users to make a reservation.





### Taxi Driver Interfaces

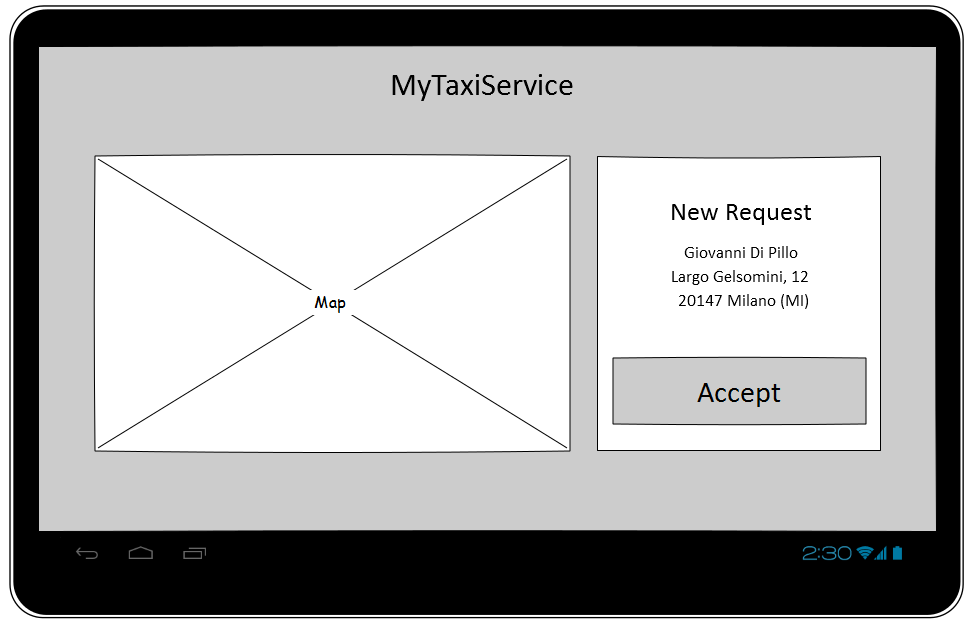
The mockups show the pages on the mobile in the taxi for the interaction with the taxi driver.

#### Accept a Request

The mockup shows the interface that allow the taxi driver to accept a request forwarded by the system.

The taxi driver visualizes a map with the position of the user and the information concerning the requested ride.

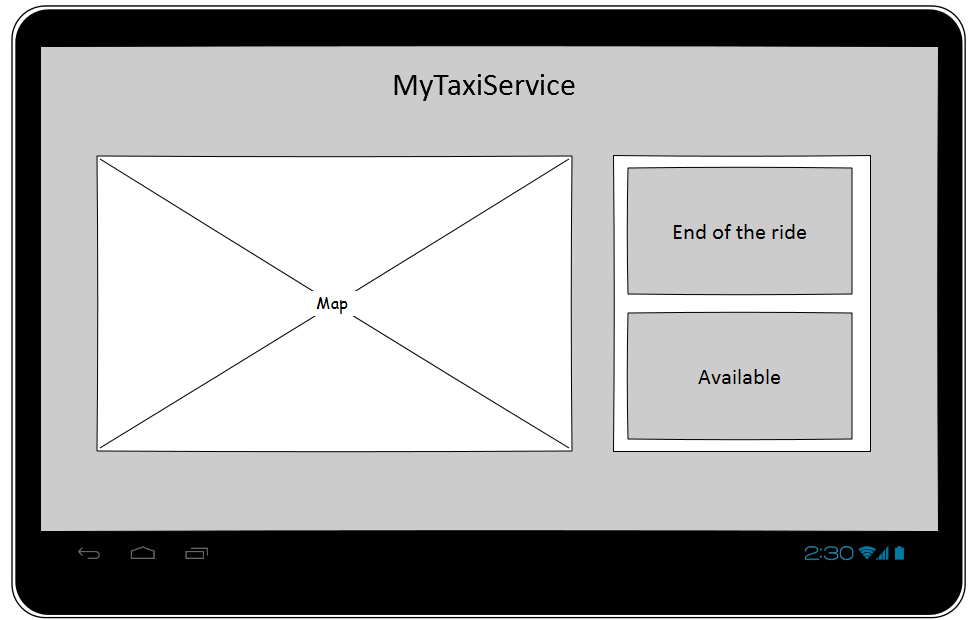
In order to accept the request the taxi driver must click on the button “Accept”.



#### End of the Ride and Availability

The mockup shows the interface that is shown to the taxi driver while he is driving towards the destination with the passenger/s aboard.

The map has a navigation function that guides the taxi driver and the buttons on the right are used to communicate that the ride is finished and to confirm the availability of the taxi after the ride.



### API Interfaces

The web application offers a development area, indicated with “API” in the Index menu of the mockup.

In that area the developers can develop additional services on top of the basic ones.

### Hardware Interfaces

Every taxi is prearranged with a special tablet that executes the software for the interaction with the taxi driver.

A generic personal computer or smartphone is sufficient for the interaction with the users.

## Functional Requirements

### Goal 1: Allow a user to register and authenticate to the service

* The user must not be already registered
* The username chosen by the user in the registration phase must be unique
* The password must contain at least a capital letter, a number and a symbol and can’t be shorter than 8 characters (non functional)
* The user must confirm his email address in order to become a registered user
* The personal data inserted in the registration phase must be correct and valid
* The authentication data inserted must be correct in order to authenticate
* If the user is not authenticated, he can only see the authentication/registration page

### Goal 2: Allow the user to request a taxi. Inform the taxi driver of the request and the position of the passenger

* The user must be authenticated
* The address inserted in the request must be correct and valid
* The taxi driver must accept the forwarded request using his mobile app
* The system must calculate the estimated time for the arrival of the taxi and inform the user
* The user must get in the taxi by 10 minutes from its arrival (non functional)
* The system must forward the request to an available taxi
* The taxi must go to the requesting user as soon as he accepts the request

### Goal 3: Manage the taxies in order to fulfil all the requests

* The system puts the taxis at the bottom of the queue of the area where they finish the rides
* If an area has a full queue, the system must send the taxi in the area with the least available taxies
* When a user makes a request, the system must send the nearest taxi available
* If there isn’t any available taxi when a user makes a request, the system must warn the user
* If a taxi doesn’t accept a request in a maximum of 5 minutes, the system forwards the request to another taxi and puts the first one at the bottom of the queue (non functional plus functional to be divided)

### Goal 4: Allow the user to reserve a ride at a specific date and time

* The user must be authenticated
* The data inserted by the user must be valid and correct
* The system must dispatch a taxi for the reservation 10 minutes before the requested time
* The user must get in the taxi by 10 minutes from the arrival of the taxi
* If something happens to the taxi while driving towards the user, the system must warn the user (and send another taxi???)
* The taxi must be at the user’s position at the requested time

### Goal 5: Allow some users to share a ride on the same route and the related cost

* The users must be authenticated
* The data inserted by the users must be valid and correct
* In order to share a ride, the different users must be on the same route (in a range of 500 meters, non functional)
* The system must calculate the fee for every user, dividing the total fee and weighting it on the amount of kilometres done by each of them
* There can’t be more than 4 passengers in a taxi at the same time
* The total number of passengers for a shared ride can be greater than four if the constraint above is respected
* If a shared ride is made for a single user, the system calculates the fee as it does for a simple request

## The World and the Machine

For a first domain analysis of this application, we use “The World & Machine” model by M. Jackson & P. Zave. This approach let us to divide the domain into three zone. “The World” that contains all the phenomena that are observed only by the Environment. “The Machine” that, otherwise, contains all the phenomena located entirely in the machine. At the end, the “Shared Phenomena” that, as the name suggests, contains all the phenomena that are controlled by the world and observed by the machine or controlled by the machine and observed by the world.

**Shared**

**Phenomena**

**The Machine**

**The World**

### The World

Contains all the Goal described upper

### Shared Phenomena

* For every Request, Call and Reservation the System must find the nearest Taxi and sends it to the User Location.
* The System must save the information about all the Reservations in order to allocate a Taxi in time.
* The System must storage the Credentials of all Registered Users in order to allow them to perform the authentication.

### The Machine

* The System must have a queue of Taxies for every zone.
* The System must have a GPS technology in order to perform the fee calculation.