

Politecnico di Milano

A.A. 2015-2016

Software Engineering 2 Project

“myTaxiService”

**Requirements Analysis and Specifications Document**

Alessio Martorana - 860584

1. **Introduction**

**1.1 Purpose**

This document represent 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, analyze the need of the customer to modelling the system, show the constraints and the limit of the software and simulate the typical use cases that will occur after the development. This document is intended to all developer and programmer who have to implement the requirements, to system analyst who want to integrate other system with this one, and could be used as a contractual basis between the customer and the developer.

**1.2 Actual System**

I suppose that there is no actual system and, hence, the entire system requested is to be created without using or modifying a previous system.

**1.3 Scope**

The aim of the project is to create a system which will optimize the taxi service of a big city. The main goals of the application are to simplify the access of passenger to the service and to guarantee a fair management of taxi queues.

The system should be able to register new users with their personal information: name, surname, email and password. The system should be also able to keep track of every taxi driver in the city, dividing the city in “taxi zones” (of approximately 2 km²) and computing the position of every taxi in each zone thanks to the GPS position acquired from a web application provided to each taxi driver. Each zone is associated to a queue of taxi codes (one code is associated to one and only one taxi) in order to handle users requests. The system should also allow each taxi driver to confirm or decline a request from a user.

A user should be able to request a taxi and the system should notify the user if a taxi has confirmed the request and, in this case, the system should send to the user taxi code and waiting time.

The system should also manage taxi queues so that when a taxi request is received from a zone, the system forwards it at the first taxi in the queue associated to that zone; at this point, if the taxi confirms the request, a confirmation to the user is sent, otherwise – if the taxi decline – the system should forward the request to the taxi in the second position of the queue and move the taxi which declined in the last position of the queue.

The system should also provide some programmatic interfaces to allow the development of additional services on top of the basic system.

**1.4 Goals**

The system will have to provide the following features:

The user should be able to:

* [G1] Sign up into the system;
* [G2] Log-in into the system;
* [G3] Request a taxi;
* [G4] Be notified by the system about if the request has been confirmed or declined and – in case of request confirmation – be notified about taxi code and waiting time;

The Taxi driver should be able to:

* [G5] Log-in to the application through their web application;
* [G6] Inform the system about his availability;
* [G7] Receive user requests from the system;
* [G8] Confirm or decline users requests;

Moreover, the system should:

* [G9] Guarantee a fair management of taxi queues.

**1.5 Definitions, acronyms, and abbreviations**

In order to make the document more clear, in this paragraph some terms used in all the document are specified better.

* **Guest:** A person who hasn’t either signed up or logged in to the application yet and, therefore, could only register to the application or log-in to the application;
* **User:** A person who has successfully logged to the application and, hence, can use all the features of the application offered to users;
* **Taxi driver:** Driver of one of the taxis handled by the system;
* **Taxi management company (if not ambiguous, indicated as “company”):** Company which handles the taxi service of the city;
* **Taxi request (if not ambiguous, indicated as “request” or “user request”):** Indicates a taxi request performed by the user through the application;
* **Status (referred to taxi driver):** Indicates if the taxi driver is “Available” or “Unavailable”, as specified in the next point;
* **Availability: “available”/”unavailable” (referred to taxi driver):** Indicates if a taxi driver is “at work” and so could be considered a valid taxi to which the system can forward a user request;

A taxi driver can change his status to “Unavailable” when he can’t do his work for some reason (work breaks, damages, issues…)

* **Taxi response (if not ambiguous, indicated as “response or “answer”):** Is the answer of a taxi driver to a request, it can be “confirmed” or “declined” as specified in the next point;
* **Request confirmed (answer of a taxi driver to a user request forwarded by the system):** Indicates that the taxi driver is available and free from other tasks and can satisfy the user request;
* **Request declined (answer of a taxi driver to a user request forwarded by the system):** Indicates that the taxi driver is busy in another request and cannot take the user request in charge;
* **City zone (if not ambiguous, indicated as “zone”):** Slices in which the system divides the city in order to manage taxis as said in the scope paragraph. Every zone measures about 2 km²;
* **Zone queue (if not ambiguous, indicated as “queue”):** The queue in which taxis are placed. The zone queue is automatically created by the system in order to manage taxis, as said in the scope paragraph. The system creates one queue for any city zone;

**1.6 Identifying Actors**

* **Guest:** could only register to the application through a registration form or log-in to the application submitting his email address and password.
* **User**: A “Guest” turns into a registered user after a successful log-in to the application; since then he can request a taxi from his position using the application.
* **Taxi driver**: Drivers who drive taxis handled by the system. A ”Guest” turns into a taxi driver after a successful log-in from the mobile app.

**2 Overall description**

**2.1 Identifying Stakeholders**

My stakeholder is the professor who assigned me the project, asking me to focus on the entire development process of a complex enterprise application, which includes requirement analysis, design, testing and project reporting phases. The professor also said that the implementation phase doesn’t have to be performed and so I didn’t analyze it. I assume that the main objective of the project is to demonstrate the ability to follow all the phases of the whole software development process.

I can imagine that some hypothetical stakeholders for the system could be the members of a company which handles the taxi service of the city, people living in the city and taxi drivers who work in the city.

I can also suppose that discussions with stakeholders have been done and surveys have been submitted to the stakeholders have been submitted.

**2.2 Domain properties and assumptions**

Domain properties and assumption that I suppose hold in the domain:

* + A user gives correct information about his position when doing a request;
  + If a request is confirmed by a taxi driver, a user remains settled in his position until the arriving of the taxi;
    - At the arriving of the taxi which will serve a user request, the user must take the ride;
      * When a taxi driver confirms a request he must go serve the request;
      * At the receiving of a request, a taxi driver must confirm or decline the request at most in 1 second;
      * When a taxi driver confirms a request he begin to move immediately to serve the request, there is no time lapse between the confirmation and the departure of the taxi driver;
      * From the moment a taxi driver receives a request, he reaches the user location at most in 15 minutes;
        + All taxi drivers and corresponding taxi codes are registered into a database - or, more in general, in some data collection – from which the application can read, and the credentials are sent to every taxi driver by e-mail and post mail.

I also suppose that the list of taxi drivers and taxi codes is always correct and up to date, for example because some employees of the taxi management company take care about filling the database with taxi drivers and correspondent taxi codes and constantly update the database as new taxi drivers begin their work career;

Each taxi driver is associated with one and only one taxi;

Each taxi is associated with one and only one taxi code, which is unique.

* + - * + Position data about taxis sent by the GPS are always correct and available at any time to the system
        + Each taxi driver can run the mobile application (for example through a mobile support he owns) of the system, in order to send his position to the system and receive, confirm or deny user requests;
        + Each taxi driver, when is in the “available” status, can immediately read the request, independently from the time in which the requests arrive;

**2.3 Proposed System**

The system will be composed by a server which will take care of dividing the city in city zones, management of the taxis through zone queues, receiving and forwarding of user requests and, more in general, of all the business logic; it will also guarantee the communication with users and taxi drivers, through either the web application or mobile app interfaces.

The other part of the system will be the clients, which can be of two types:

* User client
* Taxi driver client

User clients are clients used by user (either the web application or the mobile app) and propose to satisfy all the user goals described before.

Similarly, taxi clients are clients used by taxi drivers - in the form of a mobile app - and propose to satisfy all the taxi drivers goals described before. Two different mobile app are supplied to user and taxi drivers.

The web application will run into a web browser and – so – the interface will be included into the dynamically created web pages, the mobile app – instead - will provide graphical interface.

**2.4 User interfaces**

The user interfaces of either the web application and the mobile app are shown in this paragraph. All possible interactions actors can do with the system are described.

As images show, the general layout of the web application provides the name of the system in the top left of the window and the information about who is using the application in the top right of the window. Moreover in the left part of the window the name and/or the functionality of the actual page is shown.

In the mobile app the general layout provides again the name of the system in in the top left of the screen and the information about who is using the application in the top right of the screen; in the screen area just below this last area, the name and/or the functionality of the actual page is shown.

* **User**
* **Guest web application home page:** is the page of the web application (which only users could use) in which a guest can choose if sign-up or log-in to take user privileges. The user can choose from the two links in the center.

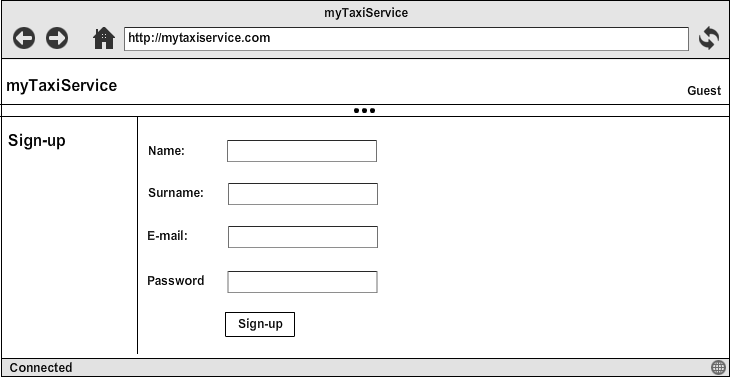
****

* **Guest web application sign-up page:** is the page of the web application (which only users could use) in which a guest signs up into the system to and get the possibility to log in.

The user signs up submitting his personal data:

* *Name*
* *Surname*
* *E-mail*
* *Password*

and then click on the “Sign-up” button.

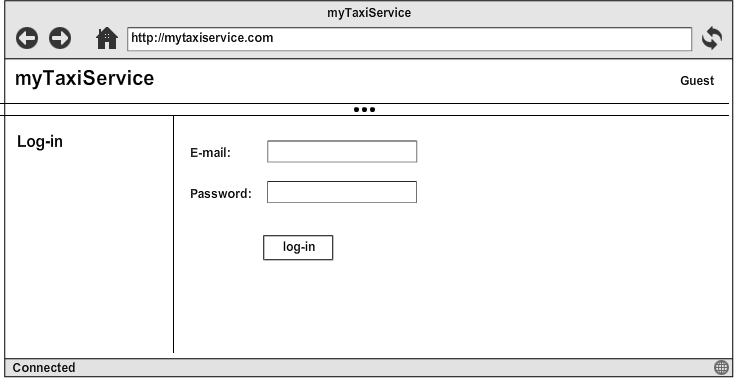


* **Guest web application log-in page:** is the page of the web application (which only users could use) in which a guest logs in into the system to take user privileges.

The user logs in submitting:

* *E-mail*
* *Password*

and then click on the “Log-in” button.

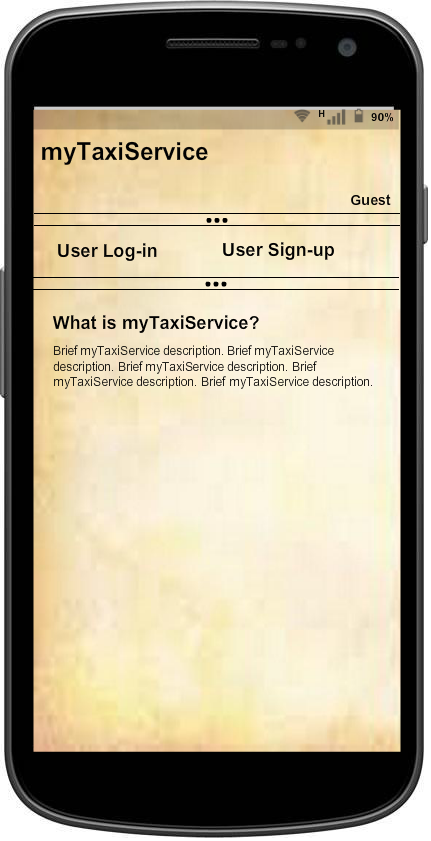
**

* **User web application request a ride page:** is the page of the web application (which only users could use) in which a user can request a taxi ride. The user must specify
* *The street where he’s located*
* *The nearest house number from the his position*

and then click on the “Request ride” button.

**

* **Guest user mobile app home page:** is the page of the mobile app provided to users, in which a guest can choose if sign-up or log-in to take user privileges. The user can choose from the two links on the top.

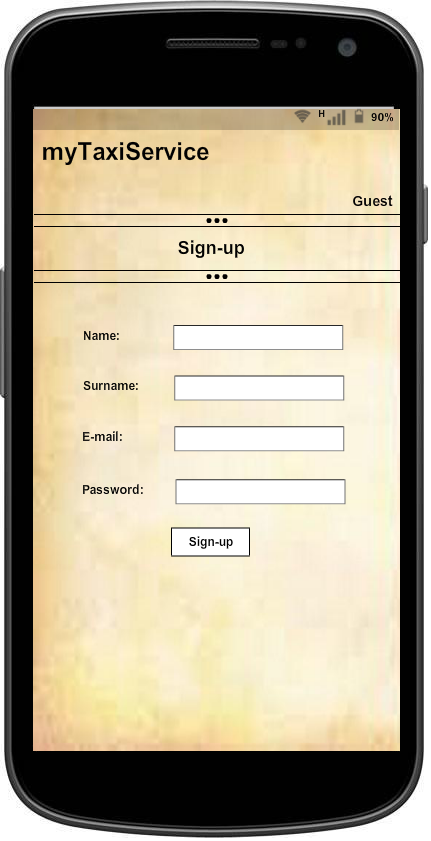
****

* **Guest user mobile app sign-up page:** is the page of the mobile app provided to users in which a guest signs up into the system to be registered into the system and have the possibility to log in.

The user signs up submitting his personal data:

* *Name*
* *Surname*
* *E-mail*

and then click on the “Sign-up” button.

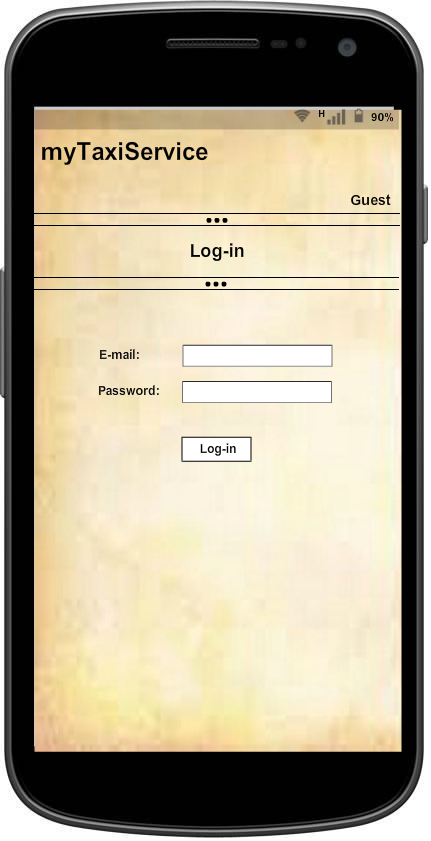


* **Guest user mobile app log-in page:** is the page of the mobile app provided to users in which guest logs in into the system to take user privileges.

The user logs in submitting:

* *E-mail*
* *Password*

and then click on the “Log-in” button.



* **User mobile app request a ride page:** is the page of the mobile app provided to users in which a user can request a taxi ride. The user must specify
* *The street where he’s located*
* *The nearest house number from the his position*

and then click on the “Request ride” button.

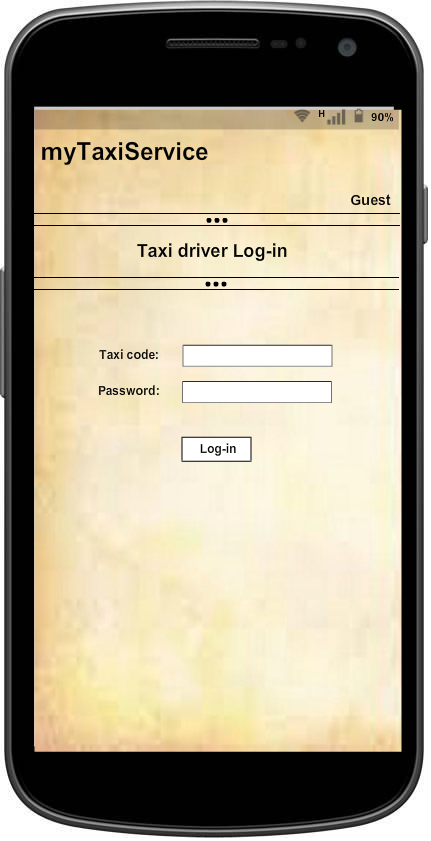
**

* **Taxi driver**
* **Guest taxi driver mobile app home page:** is the page of the mobile app provided to taxi drivers, in which a guest can only log-in to take taxi drivers privileges.

The taxi driver logs in submitting:

* *Taxi Code*
* *Password*

and then click on the “Log-in” button.

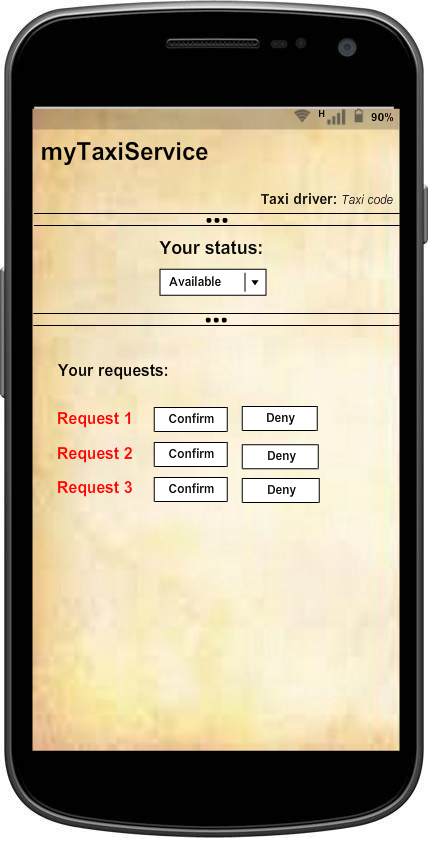


* **Taxi driver mobile app work page:** is the page of the mobile app provided to taxi drivers in which a taxi driver can set his status and manage requests made by users and forwarded by the server.

The taxi driver can choose his status among “available” or “unavailable” from a list.

For request management, the taxi driver sees the request as a list and can choose to confirm or deny the request.

The image only shows a sketch of the interface; strings “request 1”, “request 2” etc. are thought to be substituted with city street and nearest home number to where the user who sent the request is located.



**2.5 Software interfaces**

* The operating system of the mobile devices has to be android 6.0, last android version at the drawing up of this document. Source: [www.android.com](http://www.android.com) ;
* **2.6 Communication interfaces**
* All messages among clients and server will be exchanged with HTTP protocol;

**2.7 Product functions**

Here a list of the major product functions of the system.

Functions offered to users:

* [F1] Sign-up;
* [F2] Log-in;
* [F3] Request of a taxi ride;

Functions offered to Taxi drivers:

* [F4] Log-in;
* [F5] User request confirmation or deny;

General functions of the system:

* [F6] Zone queue management;
* [F7] User requests handling;

**2.8 User characteristics**

Users and taxi drivers must be able to use all major functions of their mobile devices and of the android operative system, in order to correctly run and use myTaxiService mobile app.

**3 Specific requirements**