



POLITECNICO DI MILANO  
SCUOLA DI INGEGNERIA INDUSTRIALE E  
DELL'INFORMAZIONE  
M.Sc. IN COMPUTER SCIENCE AND ENGINEERING

---

# myTaxiService

Software Design Document

---

*Authors:*

Angelo GALLARELLO

Edoardo LONGO

Giacomo LOCCI

November 23, 2015

# Contents

<b>Contents</b>	<b>1</b>
<b>1 Overview</b>	<b>2</b>
<b>2 High Level Components</b>	<b>2</b>
2.0.1 Components Interaction . . . . .	3
<b>3 Component View</b>	<b>3</b>
3.1 Server . . . . .	3
3.2 User Client . . . . .	4

# 1 Overview

**myTaxiService** is a taxi service that will operate in a big city; the main purpose is to simplify the access of passengers to the service and to guarantee a fair management of the taxi queues.

The main stakeholders of the system are the *Users*, the *Taxi Drivers* and the *Operators* as highlighted in *section 1.3* of the *RASD*.

The system is composed of four main core applications :

- Mobile Application (User)
- Web Application
- Mobile Application (Taxi Driver)
- Back-End Application

as stated in *section 1.2.* of the *RASD*

## 2 High Level Components

The system could be divide in three main high level components that do not necessarily correspond only to one real application:

### Server

The Server component is the kernel of the service we want to provide, it incorporates most of the *business logic*, it stores most of the *data* and it provides programmatic interfaces to the clients.

### User Client

The User Client components is an high level representation of the real clients available to the users of our service. It's modeled as a *thin client* and it relies on the *Server* to fulfill its tasks.

### Taxi Driver Client

The Taxi Driver Client component is an high level representation of the real clients available to the taxi drivers registered to the service. It's modeled as a *thin client* and it relies on the *Server* to fulfill its tasks.

### 2.0.1 Components Interaction

From a high level perspective the system is design following the well known *client-server* paradigm.

The interaction between the components is handled by the Server that provides a programmatic interface that is able to receive remote call from the clients.

The clients never communicate directly with one another.

## 3 Component View

This section highlights the main features and roles of every component of the system.

### 3.1 Server

The Server is composed of:

#### Back-End Application

As stated in *section 1.2.2* of the *RASD*, the *Back-End Application* is the system component that handles most of the business logic.

The application is written in *Java EE* and to fulfill its tasks (see *section 3.5.3* of the *RASD*) it needs to interface with the Internet network using the *HTTPS protocol* and the *JAVA API for RESTful Web Service*<sup>1</sup>, with a *MySQL database* and with external Google Maps API.

#### MySQL Database

The MySQL database fulfills the task of storing and granting access to all the data generated and used by the service.

A *database dump* is performed daily during the period of minor activity of the service <sup>2</sup>.

The connection between the *Java EE* application and the database is supported by the *JDBC connector*<sup>3</sup>

---

<sup>1</sup>See <https://jax-rs-spec.java.net/>

<sup>2</sup>At first, when no activity data is available, the dump will be performed at 04:00 A.M

<sup>3</sup>See <http://dev.mysql.com/downloads/connector/j/>

## 3.2 User Client

Different real clients are available to the end users of the system.

As stated in *section 1.2.2* of the *RASD* a native mobile application is developed for Android, iOS, Blackberry and WP.

Moreover a Web Application is also available.

To fulfill the requirements expressed in *section 3.5.1* and *section 3.5.2* of the *RASD*, all the clients need to communicate with the Server making calls to the REST API using platform specific API for REST HTTP calls.