



**myTaxiService**  
**Design Document**  
A.Y. 2015/2016  
Politecnico di Milano  
Version 1.0

Cattaneo Michela Gaia, matr. 791685  
Barlocco Mattia, matr. 792735

December 4, 2015

# Contents

<b>1</b>	<b>Architectural Design</b>	<b>2</b>
1.1	Overview . . . . .	2
1.2	High level components and their interaction . . . . .	3
1.3	Component view . . . . .	4
1.4	Deployment view . . . . .	4
1.5	Runtime view . . . . .	4
1.6	Component interfaces . . . . .	4
1.7	Selected architectural styles and patterns . . . . .	4
1.8	Other design decisions . . . . .	4

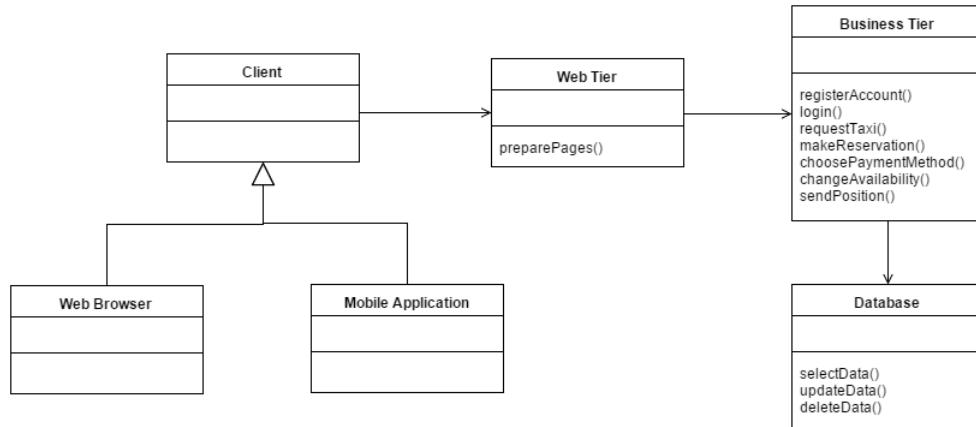
# 1 Architectural Design

## 1.1 Overview

It has been adopted a four-tier architectural model, composed by thin client, web server, application server and database.

This architecture is the best choice for our system, even if it has some cons, such as the complexity of the structure and the difficulty of set up and maintenance, it still has several pros. For example, it guarantees increased performance, great flexibility, useful if there will be any future change concerning the architecture, and great security for each level, as there is no direct route from the web server to the database, with the introduction of a middle tier, which is essential as the application deals with several personal data.

## 1.2 High level components and their interaction



- **Client tier**: this part runs on the client devices via a Web browser or the mobile application. It allows the users to insert and submit the data in the input forms, that are sent to the web tier. On the other hand, the taxi drivers can send information about their availability to the server and the application client monitors their GPS position in order to move the taxi drivers to another queue if they change their area.
- **Web tier**: this part runs on the JEE server. It always listens to all the clients requests and forwards them to the business tier, if they need to be processed. It is the responsible for the creation of the faces and pages of the client interfaces with the data fetched from the business application.
- **Business tier**: this part runs on the JEE server, too. It contains the logical part of the application, collecting and managing the information of the other tiers. It analyses the data coming from the web tier and, according to the request, it modifies or asks for the required information stored in the database, then it is able to send the result to the web tier.
- **EIS tier**: this part contains the database where all the application data are stored. It is not only accessed by the business tier, but also by the administrators, who can directly add a taxi driver account to the database.

- 1.3 Component view
- 1.4 Deployment view
- 1.5 Runtime view
- 1.6 Component interfaces
- 1.7 Selected architectural styles and patterns
- 1.8 Other design decisions