**Overview**

MyTaxiService involves different users communicating over the internet with a single system. Such users may use different platform (mobile and web) and can send requests of different types. The system must not only accept those requests and elaborate an answer in a short time, but it is required that it notifies multiple users of the occurring of some events. Usually a single event provide notifications for two types of users: taxi drivers and customers. Events notification and users requests might also necessitate to access stored data, like taxi identifier or users information.

This brief analysis clearly highlight the need of implementing MTS as a client-server-like architecture, eventually subdivided into multiple physical tiers and logical layers: this will allow to model properly the request-answer requirement. The notification and updates part, instead, requires in our opinion an event-based paradigm (also called publish-subscribe) so that user (the subscribers) would be notified by an entity (the publisher) on specific topics (a ride, for example).

These styles will be explained in detail in the following chapter.

**Selected architectural styles and patterns**

*Three-Tier Architecture*

The image shows the tier architecture of the MTS system, composed by three physical tiers. We will now analyze every tier and explain its logical functions.

* *Top tier (Client)*

The users’ machines, that in our domain are phones and computers, will have the only purpose to load the Graphical User Interface (GUI), which will show the services that can be requested from the MTS’s system. No application logic is involved at this level: Clients will only be able to send requests.

Notice that users identified as clients are limited to the followings: Taxi Drivers, Customers, Guest.

* *Middle tier*

This tier encapsulate:

* The Admin’s GUI, which is the specific interface for Administrators. It allows them to access only to their exclusive functions dialoguing directly with the Business Logic layer. It’s completely disjointed from the other users’ GUI and functions.
* The Web Server, which is the component of the system that receives and accepts the client’s requests and forward them to the business logic layer.
* The Business Logic is the part of the application that encodes the rules that determine how data can be created, displayed, stored and changed. This is the central point of the application, and connects most of the system.
* *Bottom tier (Database)*

This tier, which will be separated from the previous one with a (possibly local) network, contains all the data that MTS needs to store, ranging from the users’ information to the city map.