**Architectural Design**

**Overview**

Since our application will be used by large-scale of users, we have preferred to use a *“Multi-Tier Client-Server Architecture”*

The application will be composed of 3-tiers which are called as; “Presentation Tier”, *“Logic (Application Tier)”* and finally *“Data Tier”.*

The below diagram is tend to give a brief description about the structure of the tiers. For the simplicity of the diagram, the two *“call-response”* arcs pairs; (one from *“Application Server”* and the other from the *“Web Server”* ) are represented as a single arc pair that is going from the entire “Application Tier”. The purpose of doing is to make the diagram more readable and not to make the diagram messy with the arcs that are going over all the way on the diagram.

Figure 1 - 3-Tier Architectural Design



Presentation Tier: Presentation tier is the topmost layer which provides a graphical user interface for the user. Since our application have two different interfaces; one for the mobile application and one for the web browser, we have two sub components inside this tier. In order to be more precise about these sub components; we can say that since, *“Web Browser”* will only be used by third parties, we have included only one *“request/response”* pair between the *“3rd Party”* and the *“Web Browser”*. On the other hand, since *“Mobile Application”* will be used by both the *“Individual”* and *“3rd Party”,* it will respond to both type of the clients.

Logic Tier: Logic tier can be referred as the brain of the whole architecture. This layer is responsible from coordinating the application and processing the commands. It receives the client request from presentation layer, turns it into a query for the data tier, then send the respond of the data tier back to the presentation layer. As it can be seen from the above diagram, we have preferred to use two different sub components within this tier. We preferred this because, we have presented two different user interface components in our presentation layer and we though that it will be more accurate to separate the servers that those components will interact with. We have introduced the *“Web Server”* for the *“Web Browser”* and *“Application Server”* for the *“Mobile Application”.* Since, this layer will drive the core capabilities of our application, we have also included the link between our application and the external servers within this tier.

Data Tier: Data tier can be named as the layer of data access. It includes the database in which our whole data is stored. It is in connection with the logic layer and responsible from returning responses for the queries that are requested by the application tier. Also, this layer includes mechanisms for data persistence.

**High Level Architecture**



Figure 2 - High Level Architectural Design

The above diagram gives the whole representation of the architecture of our application. As it can be seen from the diagram two sub components of our logic tier; the *“Application Server”* and the *“Web Server”* are at the central part of the architecture. These components work as a bridge between the client and the database, and they control the interaction between them. Both the *“Mobile App”* and the *“Web Browser”* use these components (*“Application Server”* used by *“Mobile App”* and *“Web Server”* used by *“Web Browser”*) directly in order to perform any operation within the application.

**References**

<https://en.wikipedia.org/wiki/Multitier_architecture>

<https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/>

Software Engineering, 9th Edition, Ian Sommerville

Lecture Notes by Elisabetta di Nitto and Matteo Rossi