



# CS 491 Senior Design Project I High Level Design Report

*Project short-name: Tattoo'd*

Ali Onur Geven

Anıl Sert

Beril Başak Tukaç

Nazlı Özge Uçan

Furkan Usta

Supervisor: Ercüment Çiçek

Jury Members: Selim Aksoy, Uğur Güdükbay

[tattood.github.io](http://tattood.github.io)

# Table of Contents

<b>1. Introduction</b>	<b>3</b>
1.1. Purpose of the System	3
1.2. Design Goals	3
1.3. Definitions, Keywords, Acronyms and Abbreviations	3
<b>2. Proposed Software Architecture</b>	<b>5</b>
2.1. Overview	5
2.2. Subsystem Decomposition	6
2.3. Hardware/Software Mapping	8
2.4. Persistent Data Management	9
2.5. Access Control and Security	9
2.6. Global Software Control	9
2.7. Boundary Conditions	10
2.7.1 Initialization	10
2.7.2 Termination	10
2.7.3 Failure	10
<b>3. Subsystem Services</b>	<b>10</b>
3.1. Server Subsystem	10
3.2. Image Processing Subsystem	10
3.3. Database Subsystem	10
3.4. Android Subsystem	11
3.5. Augmented Reality Subsystem	11
3.6. Login Subsystem	11
<b>4. References</b>	<b>12</b>

# 1. Introduction

## 1.1. Purpose of the System

Mobile phones entered our lives a few years ago but it has quickly become the essential part of our daily routines. They offer their users increasing productivity, make budget management, entertainment with the games or utility tools with providing the applications created by software engineers. People are generally using these applications to solve some problems in their lives. The problems and difficulty in deciding which tattoo to make is the problem that we want to solve with our mobile application.

Tattoo'd is a mobile application that is working on smart phones and the purpose of this application is to help people that are in the process of considering to make a tattoo to their body with showing them how their demanded tattoo will look like at their body.

Users need to register to the system with their Google accounts and they can fully experience the application after this registration. People can share tattoos within the application and search specific tags to find tattoos they are looking for or browse other public tattoos shared from the community. After selecting the tattoo they like they can use phone's camera to place tattoo to the body part and look or take a photo of it to share from social media platforms.

## 1.2. Design Goals

**Scalability:** The application and the server should be available for its users at all times because mobile applications can be used any time and we should provide our users a available service when they want to use it. Also, the application should support multiple users simultaneously without any setback.

**Performance:** Downside of the mobile phones compared to the other electronic devices they have limited hardware and battery capabilities, since of their small sizes so the mobile application should use less energy, internet data and provide its functionality with high-performance.

**Usability:** Applications user interface must be intuitive and easy to understand since its purpose is to help people with their specific problem not create them new challenges like understanding how to interact with the application.

## 1.3. Definitions, Keywords, Acronyms and Abbreviations

**Tattoo:** A body modification made by inserting ink to the skin. Application can show virtual tattoos on the body of user using camera and augmented reality.

**Discover:** Applications browse section that users can search tattoos or look other tattoos that are shared by other people.

**Public/Private:** Tattoos can be both public and private. Public means everyone can see your uploaded tattoos and private means you are hiding the tattoos that you are uploaded to the application.

**Like:** Users can like the tattoos in the system.

**Tag:** People can give tags to the tattoos and other users search these tags to find tattoos with the searched tags.

**Search:** Users can search and find tattoos they are interested with using tags.

**Profile:** Every user will have a profile and they can create their profiles with registering to the system. Users can see their uploaded and liked tattoos in their profiles.

**Upload:** Users can upload images to the system and create tattoo using that image. After uploading they can share it publicly or don't share with the other users of the system.

**Share:** Users can share the photo which is taken during the tattoo demonstration on the social media accounts.

**Android:** A mobile operating system developed by Google [\*].

**AR:** Augmented Reality [\*].

**HTTP(S):** Hypertext Transfer Protocol (Secure) [\*].

**XML:** Extensible Markup Language [\*].

**MVC:** Model-View-Controller design pattern [\*].

**DBMS:** Database Management System.

## 2. Proposed Software Architecture

### 2.1. Overview

In this part planned architecture of Tattoo'd will be discussed in further detail. The MVC and 3-Tier architectural patterns are internalised for the application. At first the brief information about the patterns above are given. Then the subsystems of the application will be introduced. After that the software/hardware interaction will be covered and that will be followed by the data management of the application. Moreover, Access Control and Security issues will be explained in detail, and lastly boundary conditions will be described.

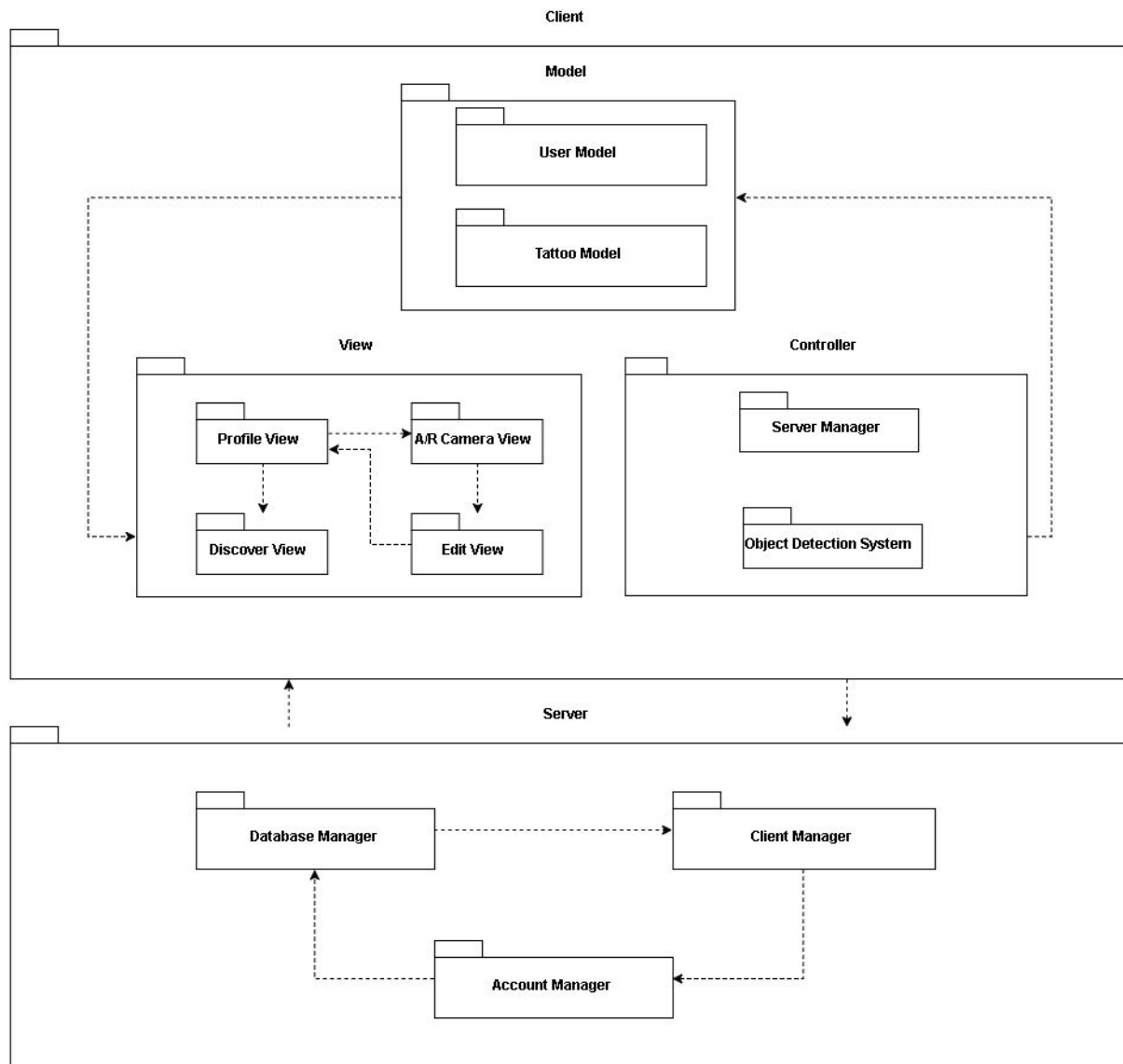
### **3-Tier Architecture**

The 3-Tier architectural model will be used in order to achieve an abstraction regarding the level of responsibility. The three tiers will represent the database/server layers which aims to receive request from the layer below, and does the computation. The layer below is the communication layer which enables the communication between layers below and above. The last layer is the user interface layer in which the abstraction of the application and the user interaction is achieved. The reason we chose this architectural model is to achieve this concept of abstraction between layers and divide the responsibility.

### **Model View Controller Architecture**

MVC architecture will be used for separating the Android application and the server side. While the application is used, the user will have no information of the internal logic behind it. The main parts, such as the image processing algorithms will be done on the server side. We chose this design pattern in order to achieve abstraction among the users and internal logic.

## 2.2. Subsystem Decomposition



**FIGURE 1. Subsystem Decomposition**

### **Android**

Android application will gather the user input and perform the corresponding actions. It will also connect with web server when it is necessary.

### **Database Manager**

Database Manager will control the information that comes to the database, and it also provides structured results to the queries. Users, tattoos and tags will be collected in the database.

### **Processing Unit**

Processing Unit will reside in the server and perform image analysis for newly uploaded tattoo designs.

**Web Server**

Web server will provide the connection between Android, application, database and the processing unit. When a new user is registered, a search is performed or a new tattoo design is uploaded application will trigger some actions on the web server.

### 2.3. Hardware/Software Mapping

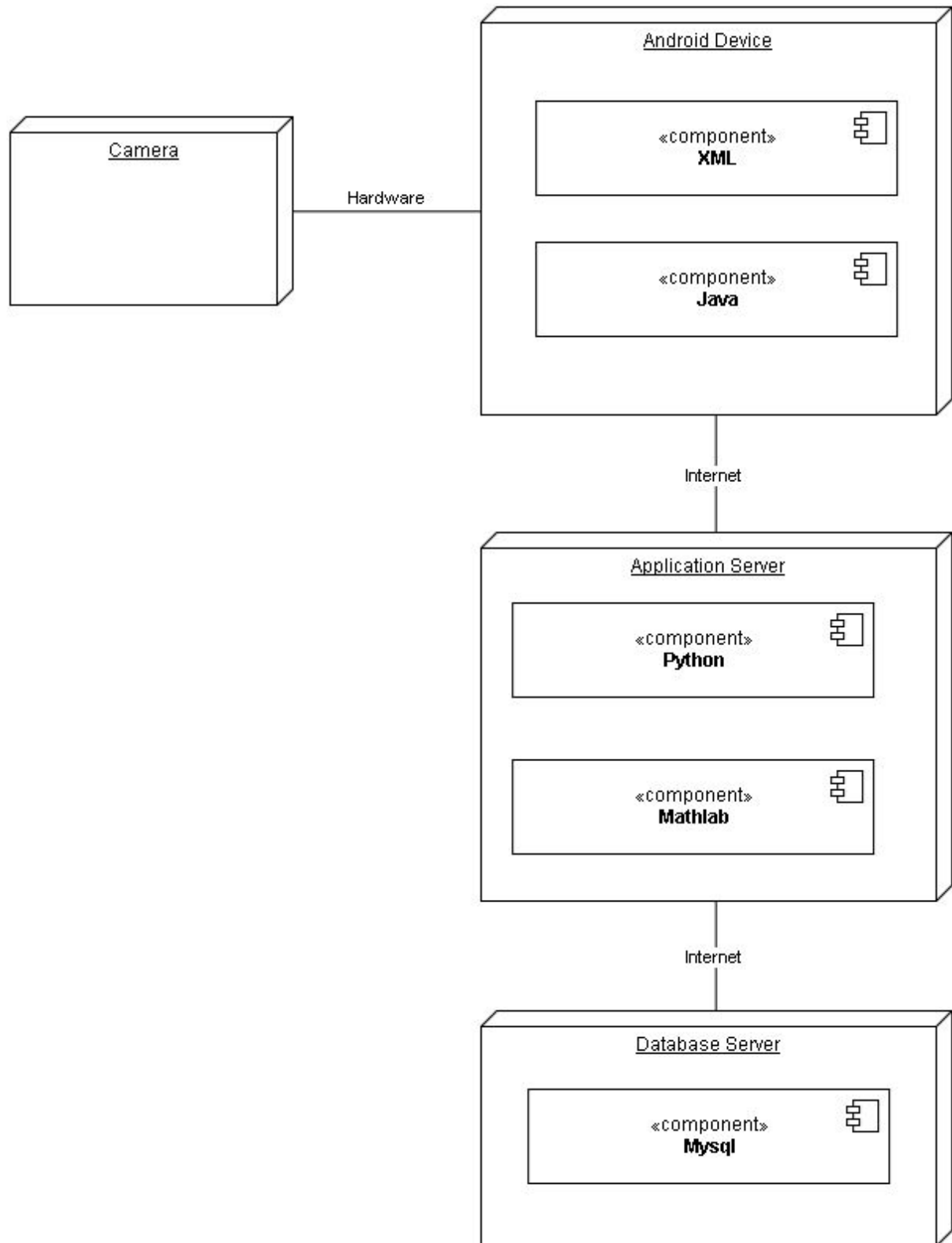


FIGURE 2. Hardware/Software Mapping



User will interact with the application using their phone and its camera. It will require an internet connection for requests to the web server.

## 2.4. Persistent Data Management

### **Tables:**

Image (ID, visibility)

User (ID, Mail, username)

Owns (UserID, ImageID)

Contains(ImageID, TagID)

Tag (ID, text)

All the data related to the users and tattoos will be stored inside the database such as user ids, tattoo images and tags. Only thing that can be updated is visibility of the image. Other than that there will only be insert and delete operations on the database.

## 2.5. Access Control and Security

There are two important points in Tattoo'd regarding access control and security. First, registration and login system and the second is privacy of the user profiles.

For the login system we will use Google Sign-in for Android [6]. This will ease our job for checking the validity of the entered e-mail address. Although the user will select a username, there will not be any passwords and username will be used only to distinguish users.

For privacy of the data we will give options to the user to choose whether s/he wants to share the tattoos with everyone or keep it secret. Since search results will be returned from the server, all the necessary checks will be performed to prevent other users from accessing the tattoos that are marked private.

## 2.6. Global Software Control

In Tattoo'd, there is one main global software control system, which is the android web-based phone application. Each user connects to the same server. The application logic is based on the internal control flow as well as the event-driven programming. Based on the event-driven programming, whenever the user sends a request to the server, he/she communicates with the server and receives a reply. In terms of internal flow, whenever the user requests, the flow of events occur within the local system.

For authentication, login and register options, the application will interact with the outside services such as Google authentication [6].

## 2.7. Boundary Conditions

### 2.7.1 Initialization

When the user opens the application for the first time he/she will be greeted with a login screen. After successfully signing-in to their Google account users will be asked to a username if it is their first visit to the application. If the user is an existing user and simply returning to the application user will pass through that step and will continue to use his/her existing username.

### 2.7.2 Termination

After logging in the user will stay as it is until s/he deletes the application. This way, they will not be prompted to the login page every time they open the application.

### 2.7.3 Failure

In case of a failure event the user will be informed and the application will try to continue from the nearest safe state. To protect the integrity of the database all the requests in the server side will be atomic operations. Thus, if an error occurs halfway through the series of operations, all the changes will rollback and the database will remain unchanged.

## 3. Subsystem Services

### 3.1. Server Subsystem

Server subsystem will provide an internal communication in order to be able to perform operations that require many resources. Based on event driven programming, it will be responsible for getting client requests and forwarding them to the relevant subsystems for a response.

### 3.2. Image Processing Subsystem

Image processing subsystem will be responsible for image processing related procedures such as object recognition in the upload process or automatic tag assignment. The subsystem will perform according to the requests that comes from server subsystem.

### 3.3. Database Subsystem

Database subsystem is important to maintain persistent data management. Persistent data information such as users, tattoos and tags will be stored. The system will use it frequently for events such as uploading a tattoo or displaying tattoos according to searched tags. It will perform according to the requests that comes from server subsystem.

### 3.4. Android Subsystem

Android subsystem is the client side of the system. It must be able to provide a communication with the server through a user interface. Thus, it will be responsible for sending requests according to prompts of the user.

### 3.5. Augmented Reality Subsystem

Image processing subsystem will be responsible for augmented reality related procedure of the system. The subsystem will be in communication with android subsystem. According to the prompts of the user augmented reality subsystem will be activated and perform.

### 3.6. Login Subsystem

Login subsystem will work with Google Authentication[6]. It will be also in communication with android subsystem. After the authentication of the user, it will send a request to the android subsystem to get the specialized profile of the user from server.

## 4. References

[1] Android. (2016). *Android*. [online] Available at: <https://www.android.com/> [Accessed 29 Dec. 2016].

[2] En.wikipedia.org. (2016). *Augmented reality*. [online] Available at: [https://en.wikipedia.org/wiki/Augmented\\_reality](https://en.wikipedia.org/wiki/Augmented_reality) [Accessed 29 Dec. 2016].

[3] En.wikipedia.org. (2016). *Hypertext Transfer Protocol*. [online] Available at: [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [Accessed 29 Dec. 2016].

[4] W3.org. (2016). *Extensible Markup Language (XML)*. [online] Available at: <https://www.w3.org/XML/> [Accessed 29 Dec. 2016].

[5] Msdn.microsoft.com. (2016). *Model-View-Controller*. [online] Available at: <https://msdn.microsoft.com/en-us/library/ff649643.aspx> [Accessed 29 Dec. 2016].

[6] "Google Sign-In for Android | Google Developers", *Google Developers*, 2016. [Online]. Available: <https://developers.google.com/identity/sign-in/android>. [Accessed: 30- Dec- 2016].