



CS 492 Senior Design Project II Final 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

Table of Contents

1. Keywords	3
2. Introduction	4
3. Final Architecture and Design	5
3.1. Subsystem Decomposition	5
3.2. Hardware/Software Mapping	7
3.3. Persistent Data Management	7
3.4. Access Control and Security	8
4. Final Class Diagram	9
5. Class Interfaces	12
6. Current Status	17
7. Engineering Solutions and Contemporary Issues	18
8. Tools and Technologies	18
9. Resource Usage (Internet, book, paper etc.)	20
10. Similar Applications and Innovations	20
11. User Manual	21
12. References	29

1. Keywords

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: Initial screen that shows the most liked and recently updated tattoos.

Public/Private: Tattoos can be set to be either publicly available or private.

Like: Users can like the tattoos in the system.

Tag: People can set tags to the tattoos and search using those tags.

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

Profile: Every user will have a profile that is created upon first usage. 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.

Share: Users can take a screenshot and share the photo on their social media accounts.

Android: A mobile operating system developed by Google.

AR: Augmented Reality.

HTTP: Hypertext Transfer Protocol.

XML: Extensible Markup Language.

MVC: Model-View-Controller design pattern.

RDBMS: Relational Database Management System

2. Introduction

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.

3. Final Architecture and Design

3.1. Subsystem Decomposition

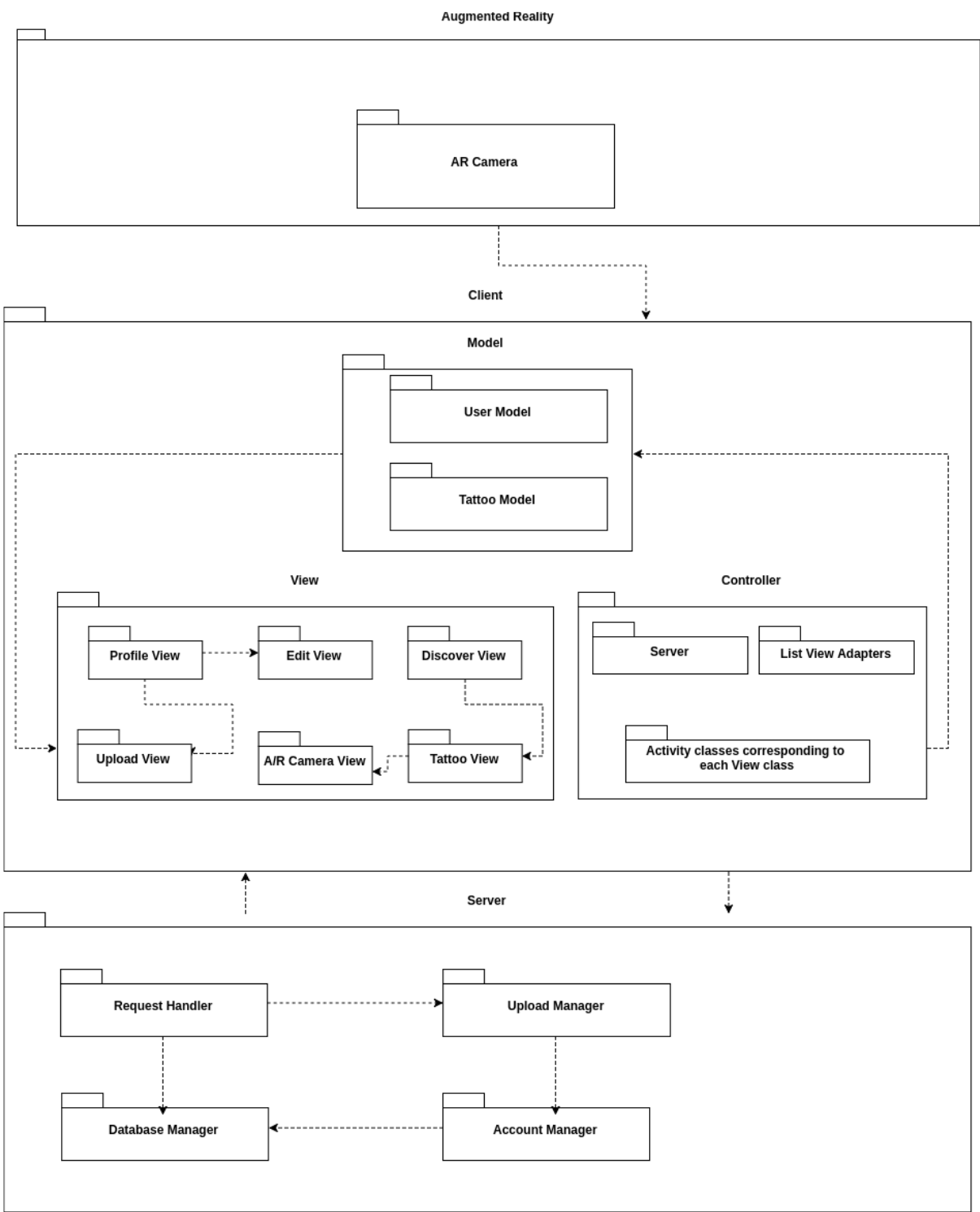


Figure 1. Subsystem Decomposition

Client Subsystem

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

Model Subsystem

Model subsystem consists of user and tattoo model classes. It includes the models of the persistent data.

View Subsystem

View subsystem consists of the visual representation of each model.

Controller Subsystem

Controller subsystem maintains the interaction between view subsystem and model subsystem.

Server Subsystem

Server will provide the connection between client, 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.

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 upload manager and performs object detection to newly uploaded tattoo designs.

Augmented Reality Subsystem

Augmented Reality subsystem will be responsible for the procedures executed on the AR camera. It communicates with the client subsystem.

3.2. Hardware/Software Mapping

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

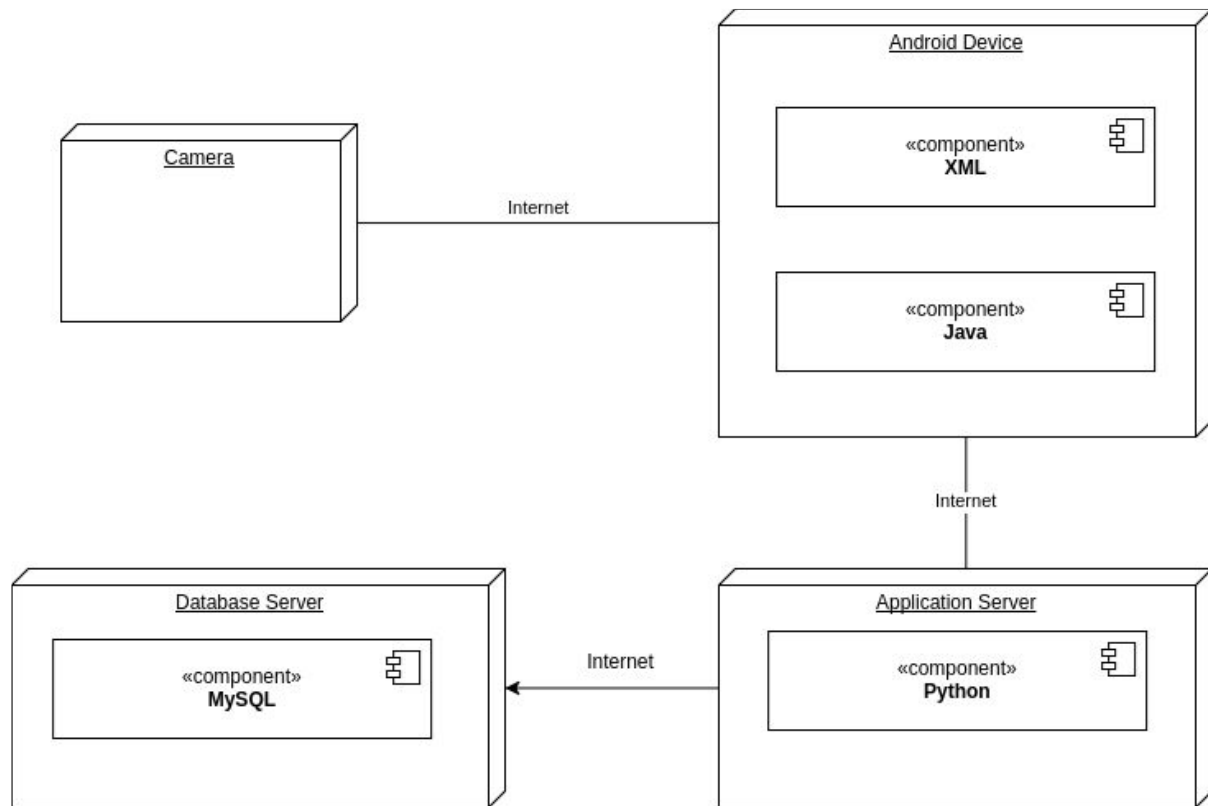


Figure 2. Hardware/Software Mapping

3.3. Persistent Data Management

Tables:

Tattoo(ID, owner_id, visibility, upload_time)

User (ID, mail, username, profile_photo)

Contains(tattoo_id, tag_id)

Tag (ID, description)

Likes(owner_id, tattoo_id)

All the data related to the users and tattoos will be stored inside the relational database management system in a remote server rented from DigitalOcean [1]. Information about tattoos, users and the relationships between them are stored on the database.

3.4. 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 Firebase authentication system integrated with Google Sign-in [2]. 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 not for authentication.

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 private. 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 as private.

Every request to the server requires a token that is used to identify the user and prevent unauthorized actions. Even though these tokens are provided by Firebase and handled by the mobile app in the client side, it is still controlled in the server side to prevent forged tokens.

4. Final Class Diagram

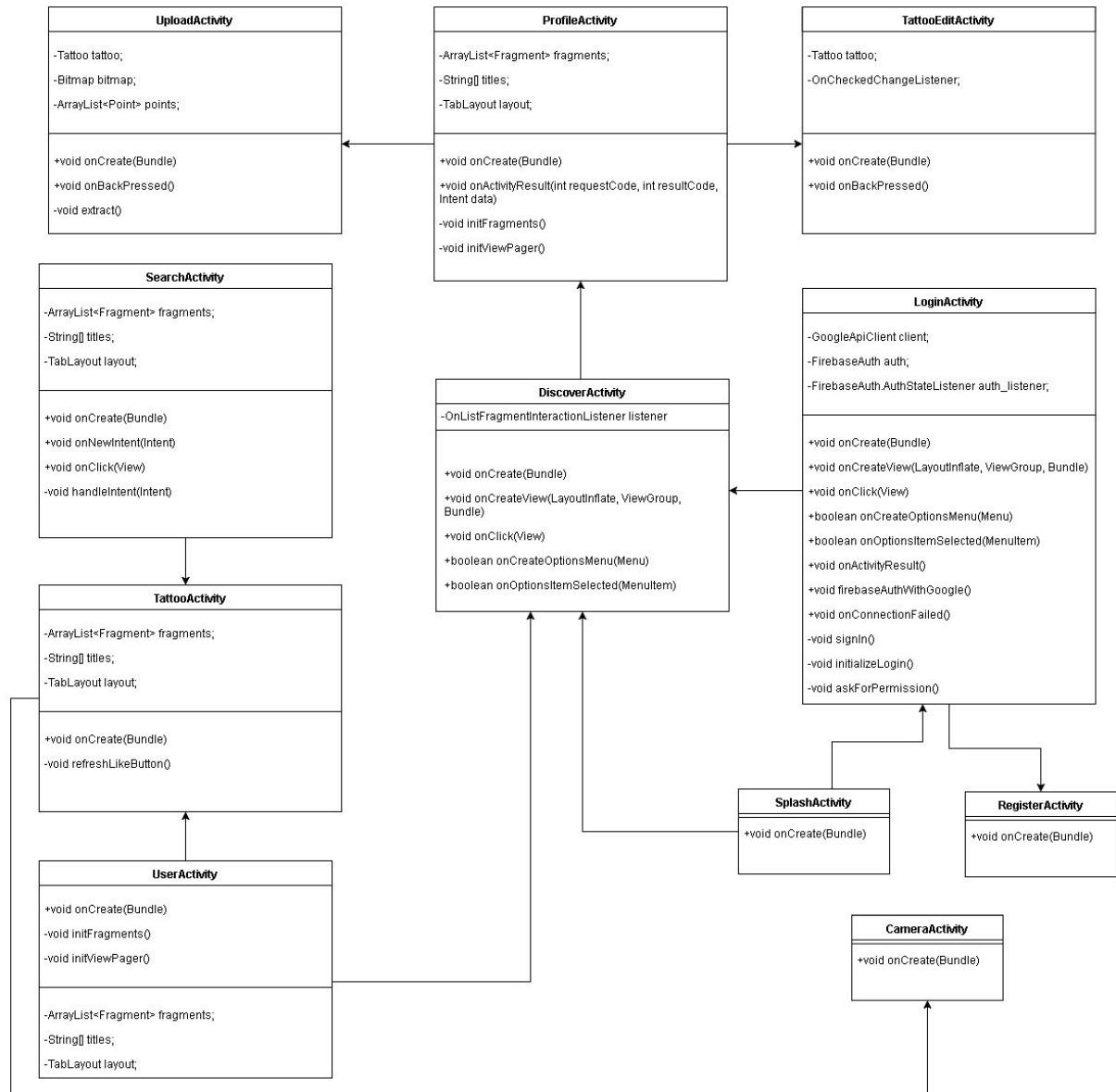


Figure 3. Android Class Diagram

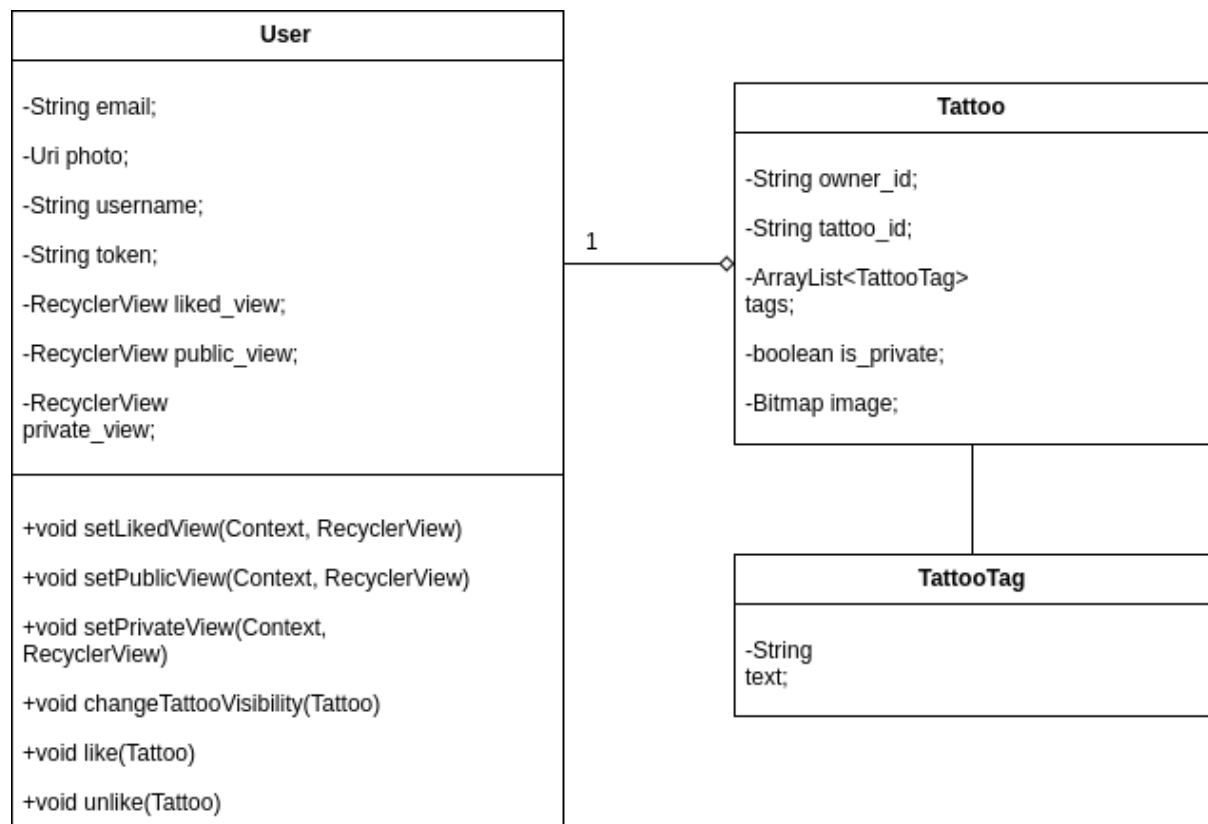


Figure 4. Models Class Diagram

Server class
interacts with all
of the
activity classes,
internally

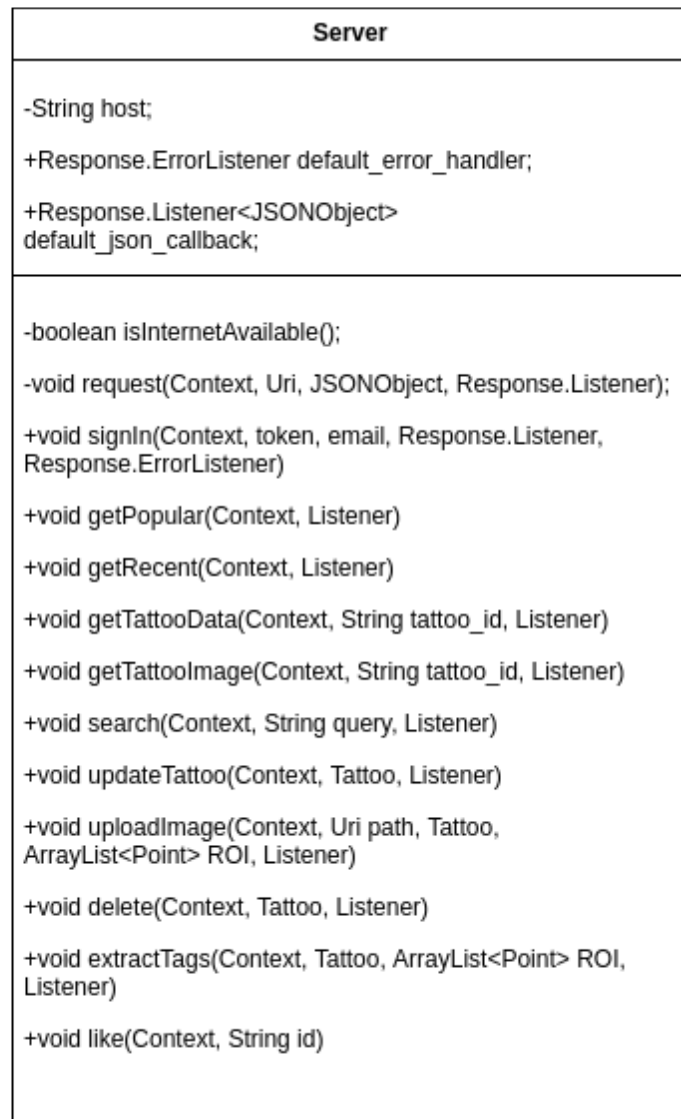


Figure 5. Server Class Diagram

5. Class Interfaces

Android Classes

DiscoveryActivity
Controller class for Discovery Page
-OnListFragmentInteractionListener listener
+void onCreate(Bundle) +void onCreateView(LayoutInflater, ViewGroup, Bundle) +void onClick(View) +boolean onCreateOptionsMenu(Menu) +boolean onOptionsItemSelected(MenuItem)

LoginActivity
Controller class for Login Page
-GoogleApiClient client; -FirebaseAuth auth; -FirebaseAuth.AuthStateListener auth_listener;
+void onCreate(Bundle) +void onCreateView(LayoutInflater, ViewGroup, Bundle) +void onClick(View) +boolean onCreateOptionsMenu(Menu) +boolean onOptionsItemSelected(MenuItem) +void onActivityResult() +void firebaseAuthWithGoogle() +void onConnectionFailed() -void signIn() -void initializeLogin() -void askForPermission()

RegisterActivity
Controller class for RegistrationPage
+void onCreate(Bundle)

ProfileActivity
Controller class for Profile Page
-ArrayList<Fragment> fragments; -String[] titles; -TabLayout layout;
+void onCreate(Bundle) +void onActivityResult(int requestCode, int resultCode, Intent data) -void initFragments() -void initViewPager()

SearchActivity
Controller class for Search Page
-ArrayList<Fragment> fragments; -String[] titles; -TabLayout layout;
+void onCreate(Bundle) +void onNewIntent(Intent) +void onClick(View) -void handleIntent(Intent)

TattooEditActivity
Controller class for Tattoo Edit Page
-Tattoo tattoo; -OnCheckedChangeListener;
+void onCreate(Bundle) +void onBackPressed()

TattooActivity
Controller class for Tattoo Page
-ArrayList<Fragment> fragments; -String[] titles; -TabLayout layout;
+void onCreate(Bundle) -void refreshLikeButton()

UploadActivity
Controller class for Upload Page
-Tattoo tattoo; -Bitmap bitmap; -ArrayList<Point> points;
+void onCreate(Bundle) +void onBackPressed() -void extract()

UserActivity
Controller class for Other User Page
-ArrayList<Fragment> fragments; -String[] titles; -TabLayout layout;
+void onCreate(Bundle) -void initFragments() -void initViewPager()

SplashActivity
Controller class for Splash Screen
+void onCreate()

CameraActivity
Controller class for Camera Page
+void onCreate()

Tattoo
Model class for Tattoo
-String owner_id; -String tattoo_id; -ArrayList<TattooTag> tags; -boolean is_private; -Bitmap image;

User
Model class for User
-String email; -Uri photo; -String username; -String token; -RecyclerView liked_view; -RecyclerView public_view; -RecyclerView private_view;
+void setLikedView(Context, RecyclerView) +void setPublicView(Context, RecyclerView) +void setPrivateView(Context, RecyclerView) +void changeTattooVisibility(Tattoo) +void like(Tattoo) +void unlike(Tattoo)

TattooTag
Model class for Tattoo
-String text;

Server
Utility class that provides the interaction between the client and Server subsystem
-String host; +Response.ErrorListener default_error_handler; +Response.Listener<JSONObject> default_json_callback;
-boolean isInternetAvailable(); -void request(Context, Uri, JSONObject, Response.Listener); +void signIn(Context, token, email, Response.Listener, Response.ErrorListener) +void getPopular(Context, Listener) +void getRecent(Context, Listener) +void getTattooData(Context, String tattoo_id, Listener) +void getTattooImage(Context, String tattoo_id, Listener) +void search(Context, String query, Listener) +void updateTattoo(Context, Tattoo, Listener) +void uploadImage(Context, Uri path, Tattoo, ArrayList<Point> ROI, Listener) +void delete(Context, Tattoo, Listener) +void extractTags(Context, Tattoo, ArrayList<Point> ROI, Listener) +void like(Context, String id)

Server Classes

RequestHandler
Handles the incoming requests and performs the necessary actions
+String database_uri
+int login(token, email) +int register(token, email, username) +int logout(token) +List<Tattoo> popular(limit) +List<Tattoo> recent(limit) +List<Tattoo> user_likes(token, username, limit) +List<Tattoo> user_tattoo(token, username, limit, visibility) +int like(token, tattoo_id) +int unlike(token, tattoo_id) +Image tattoo(token, tattoo_id) +int tattoo_update(token, tattoo_id, tags, visibility) +List<Tag> extract_tags(token, image, points) +int tattoo_upload(token, image, visibility, tags, points) +int tattoo_delete(token, tattoo_id) +List<Tattoo> search(token, query, limit, latest=None)

ProcessingUnit
Processing unit that is executed during upload
+image crop(path, points) +image make_transparent(image) +List<Tag> classify(image, threshold)

6. Current Status

All of the requirements given in the requirements elicitation step of Tattoo'd are completed successfully. As the project group, we are trying to improve the realistic look of the placed tattoo on AR Camera. Moreover, we are working on to reduce the data and battery usage of the mobile application and we are also working on other optimizations for the tracking algorithms and improve the accuracy of our model using various Image Processing techniques.

However, even though our project correctly performs the tasks we are aimed to do, we have some constraints that user needs to comply, such as placing the tattoo at the center of the camera and having a uni-coloured, simple background.

7. Engineering Solutions and Contemporary Issues

Tattoo'd aims to develop innovative augmented reality for Tattoos such that it is as realistic as trying the tattoo in real life. By processing the camera's vision input relative models, effects and gradients are used for more realistic results. Also using simultaneous localization and mapping algorithms (SLAM) [3], innovative solutions are acquired different from the applications on the market.

8. Tools and Technologies

Android SDK

Android SDK is a set of development tools to develop applications for the Android operating system. Android SDK written in Java and developed by Google and includes a debugger, libraries, emulator, documentation, sample code, and tutorials [4].

Android Studio

Android studio is an IDE specifically created for the Android development.

Python/Flask

Flask is an open source micro web framework written in Python [5]. Server side of the project that communicates with database and the Android application is written using Python and Flask.

MySQL

MySQL is the world's most popular open source relational database management system which is using SQL for interacting with the database [6]. Every database communication on the server side accomplished by using MySQL.

Tensorflow

Tensorflow is an open source software library for numerical computation using data flow graphs [7]. It is generally used across machine learning and deep learning tasks. It is firstly developed by Google engineers to meet their needs to construct and train neural networks for their products [8]. In project, this powerful library is used for the automatic tagging of tattoos that are uploaded by application users.

OpenCV

Open Source Computer Vision Library (OpenCV) is an open source computer vision and machine learning library that has more than 2500 algorithms from face detection to movement tracking to use in these areas [9]. In the project, we use OpenCV from image cropping, edge detection and distance calculations on images.

Firebase

Firebase is a mobile and web application development platform that helps developers and companies to get analytics from their products, get better and easier cloud development environment and company growth tools like ad management tools. It also has great integration with most of the Google products and services. We use Firebase services in our application for user authentication on android devices.

Digital Ocean

DigitalOcean, Inc. is an American cloud infrastructure provider headquartered in New York with data centers worldwide. DigitalOcean provides developers cloud services that help to deploy and scale applications that run simultaneously on multiple computers. As of December 2015, DigitalOcean was the second largest hosting company in the world in terms of web-facing computers. [10]

Unity3D

Unity3D is the cross platform game engine that is mainly used for video game, simulation and computer vision application development. In project Unity3D used for AR camera development, virtual 3D world, 3D model and texture creation.

Wikitude SDK

Wikitude SDK is a development framework utilizing image recognition, tracking and geolocation technologies with the AR technology. Wikitude SDK is the backbone of AR development of the project and used for tracking tattoo placement and creating AR graphics on camera screen.

Git/Github

Git is a free and open source distributed version control designed for tracking changes in computer files and coordinate people who work on these files [11]. Github is our preferred version control repository that is using git for collaboration and issue tracking tools.

Various Github Repositories

KarlGT0811/material-round-button: Used for button style.

ArthurHub/android-image-cropper: Used for cropping images on the load function.

Cutta/tagview: Used for tagging in style.

facebook/Fresco: Used for caching.

Taishi-Y/InstagramLikeColorTransitionAndroid: Used for dynamic colour transition on the background.

jd-alexander/LikeButton: Used for animating like button.

9. Resource Usage (Internet, book, paper etc.)

We mostly collected information using the official documentation of the used tools such as Wikitude, Android SDK and Tensorflow. Moreover, we used Stack Overflow and #android IRC channel on freenode to solve our problems, and lastly we used other open source Android projects on Github to learn how to solve the similar problems that we had.

10. Similar Applications and Innovations

InkHunter

The only application that is similar to our projects is a mobile application named InkHunter [12]. InkHunter is trying to help people see tattoos on their body parts but it has some constraints that we want to eliminate with our project. For example, InkHunter needs a pattern to draw on body to work but not everyone brings a pen with them to use an application. Also, when InkHunter places the tattoo and the tattoo is too big for that body part image stays outside of the body and seen on the background.

We bring innovative solutions to these constraints that InkHunter has. We use an instant tracker to track the body part of the user that s/he wants to put a tattoo so our application does not need any drawing or pattern for tracking user movements. Because we use a 3D models for our tattoos to place them for tracking, we can wrap the tattoo to that body part so there is no part of the image stays outside of the body part.

11. User Manual

The following images are screenshots from the actual application for the purpose of explaining the usage information of Tattoo'd.

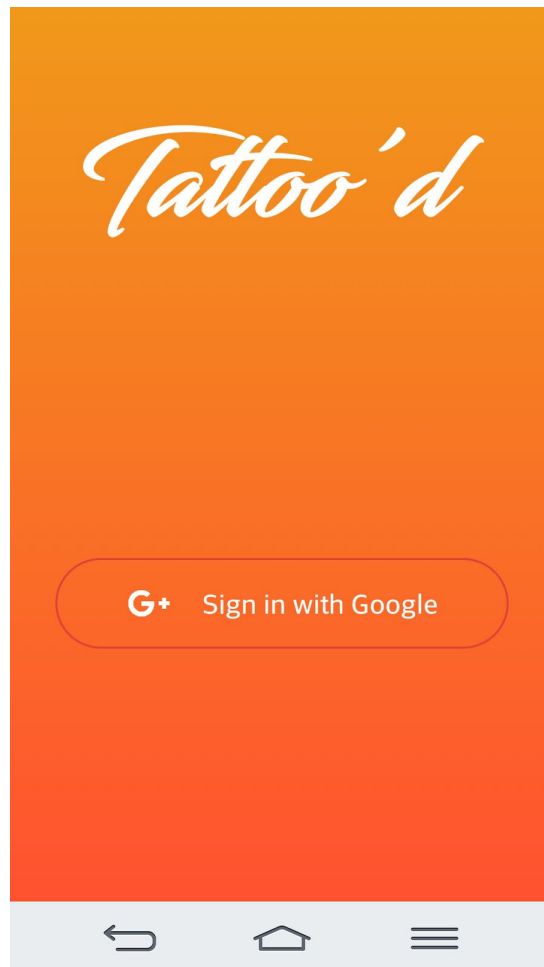


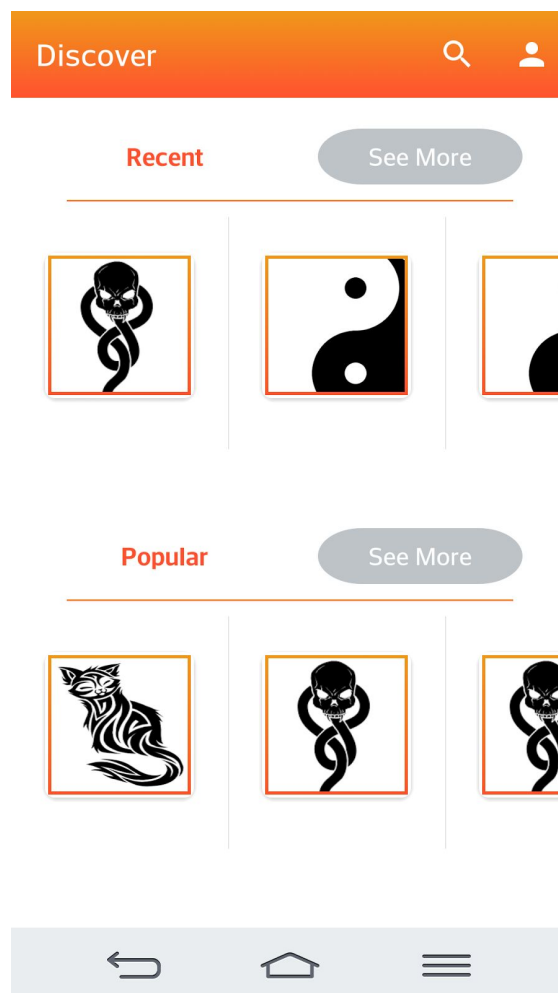
Figure 6. Login Page

Login Page

Login page is opening page of the application where a user can connect to the application server by simply logging in with their Google Account. If the user is not registered to the system, the application will redirect to registration page.

Register Page

If the user logs successfully with their Google Account and they are not already registered to the system, they need to register a name for their account name to use the application and create a profile with their selected account name.



Discovery Page

This is the page for discovering new tattoos. There are two sections which are: Recent and Popular. Recent section includes the latest tattoos and Popular section includes the most liked tattoos that are uploaded to Tattoo'd.

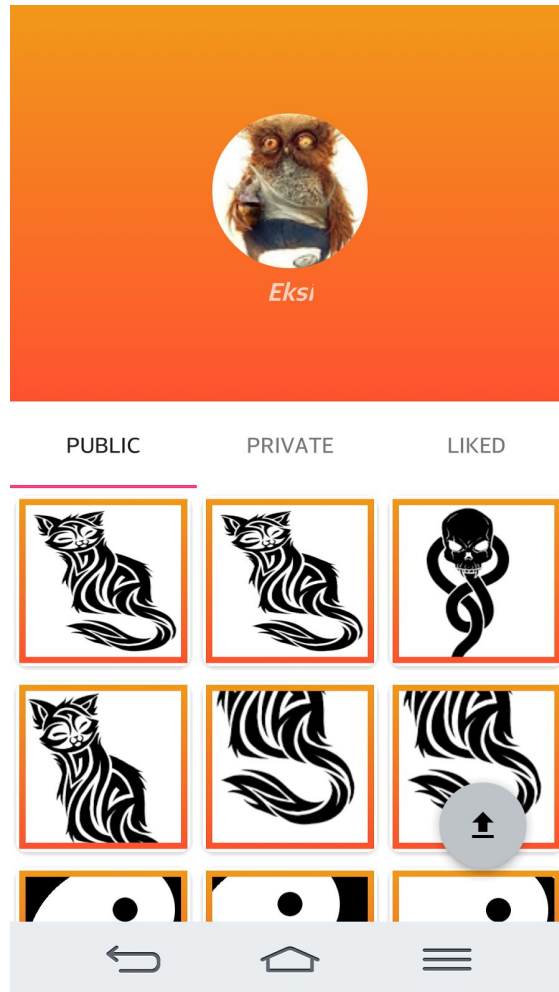


Figure 7. Profile Page

Profile Page

Users can see their own profile information like profile photo, account name and get information about the tattoos they upload and like with this page. User can reach his/her tattoos from this page.

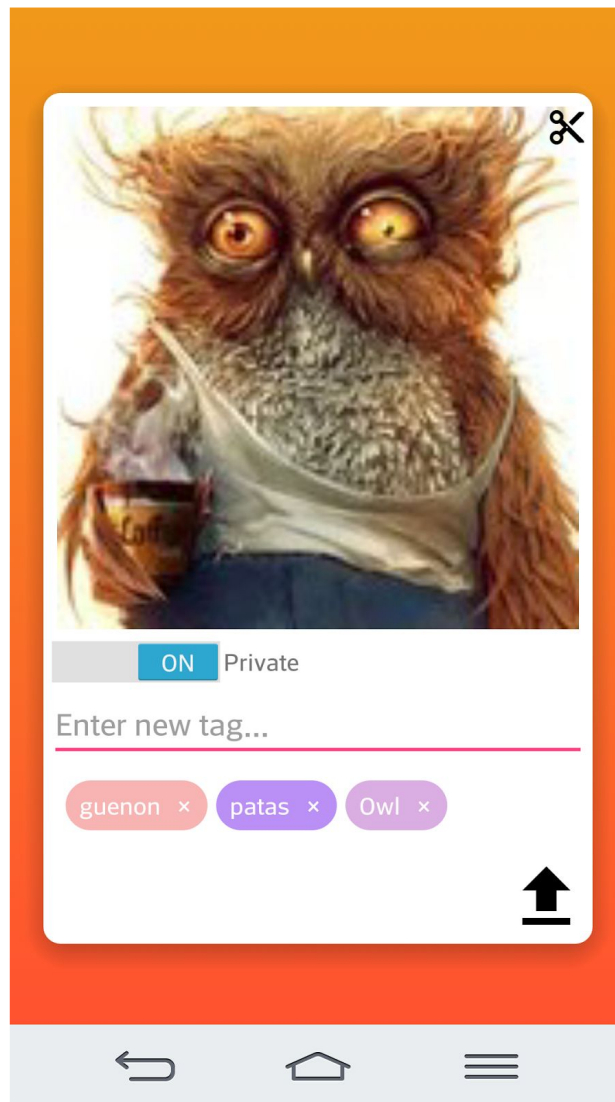


Figure 8. Tattoo Edit Page

Edit Tattoo Page

A user can select any of his/her tattoos to modify. Tattoos can be set as public or private. Tags of the tattoo can be changed or deleted in this page or the tattoo can be deleted.

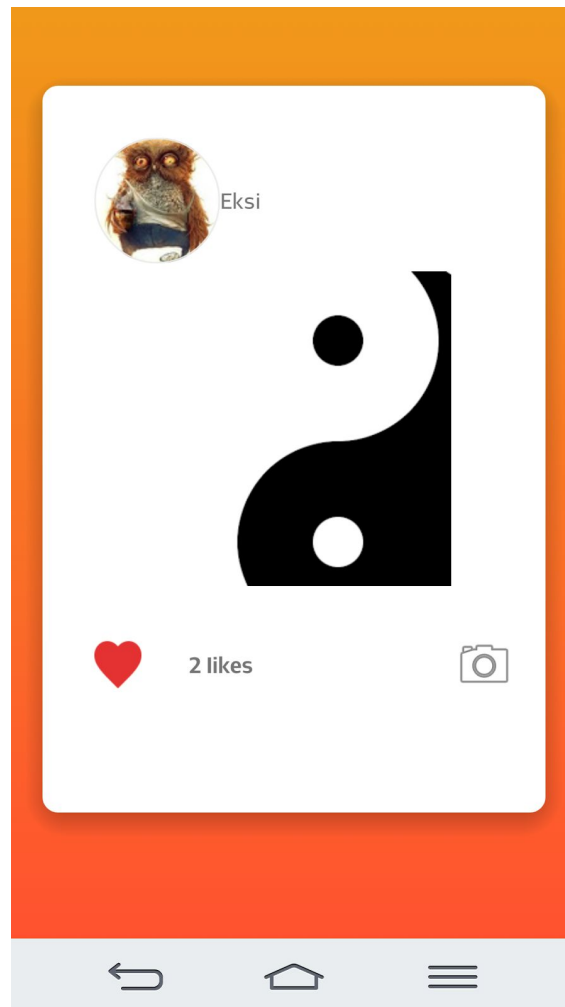


Figure 9. Tattoo Page

Tattoo Page

Users can display other user's tattoos using this page. On this page users can navigate to the other user's profile page, like/dislike the tattoo, try this tattoo using AR camera or search other tattoos with the same tag.

Other User Page

User can reach other users profile pages and see only the public tattoos of the other users and also try them for their own.

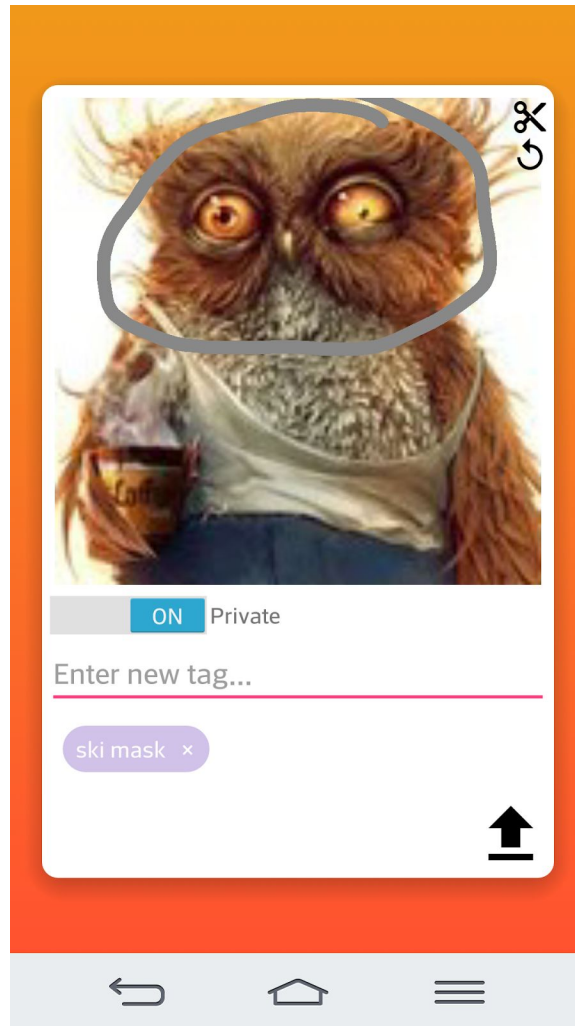


Figure 10. Upload Page

Upload Tattoo Page

User can select tattoo images from their mobile phone and upload to the system. When uploading the tattoo, user can set its privacy value as public or private and add tags. User can also crop the image as desired in order to use only desired part of the image.

Search Screen

Users can manually search a tag or click a tag that another tattoo has to search other tattoos uploaded and publicly available and see them using this page.



Figure 11. See More Page

See More Page

This page can only be opened from the discovery page by clicking see more buttons of the related section such as Popular or Recent in order to see all of the images that belongs to the section that it belongs to.

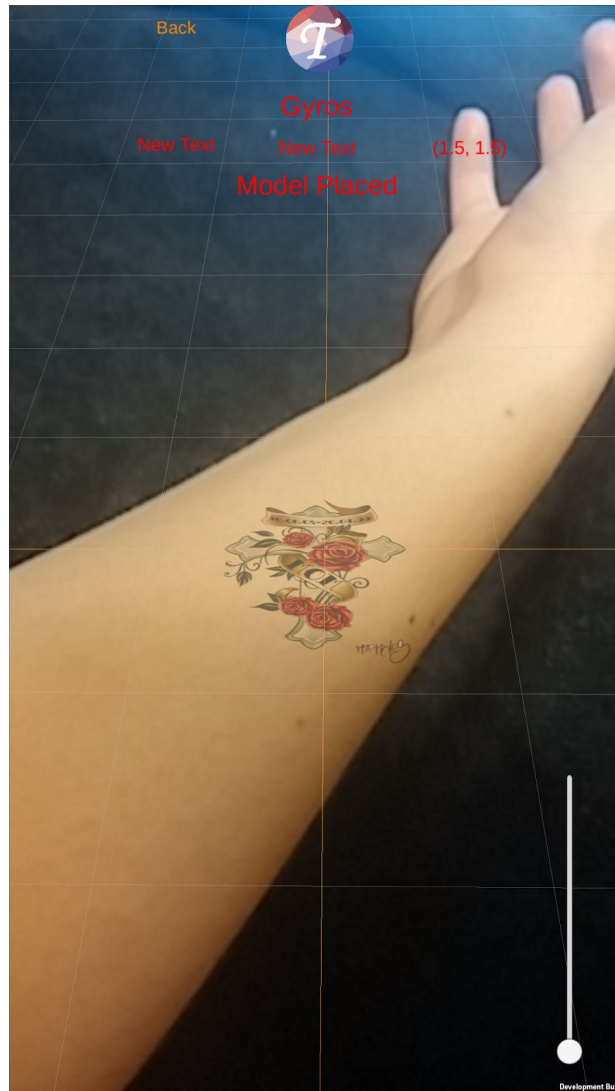


Figure 12. AR Camera Screen

AR camera Screen

When the user wants to try a new tattoo clicking the try tattoo button on tattoo page they automatically navigate to the camera application that users can try that tattoo on their body parts using AR technology. Users can adjust the position of the tattoo with bringing the crosshair on the screen to the place they wanted to put on. If they are happy with the position of the tattoo they can initialize and put the tattoo using the “Initialize” button. After putting the tattoo they can change the curvature of the tattoo using slider on the bottom right corner of the screen and return back to the initialization screen using “Back” button on upper left corner.

12. References

- [1] <http://digitalocean.com>
- [2] <https://firebase.google.com/docs/auth/android/google-signin>
- [3] https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
- [4] https://en.wikipedia.org/wiki/Android_software_development
- [5] [https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework))
- [6] <https://en.wikipedia.org/wiki/MySQL>
- [7] <https://www.tensorflow.org>
- [8] <https://en.wikipedia.org/wiki/TensorFlow>
- [9] <http://opencv.org/about.html>
- [10] <https://en.wikipedia.org/wiki/DigitalOcean>
- [11] <https://git-scm.com>
- [12] <http://inkhunter.tattoo>