

EUROPEAN UNIVERSITY OF LEFKE

FACULTY OF ENGINEERING

Software Engineering

Graduation Project II

Paw Time Mobile Application

Merve ALTINIŞIK

181375

I made a mobile application project that allows us to help the most innocent creatures of the world. By uploading photos of animals in my application and entering detailed information, if any, it allows other users to adopt this animal. Apart from adoption, as another option, it allows the animals to appear on the map thanks to the location information entered, and in this way, people close to that area have the opportunity to go to the location and help the animal with food, water, etc.

The purpose of this application is to enable people to have an easy access to animals in need of help and to find help more easily and faster.

Supervisor

Assoc. Prof. Dr. Cem Burak Kalyoncu

Publish Date

20/08/2023

Table Of Contents

Merve ALTINIŞIK	i
181375	i
Assoc. Prof. Dr. Cem Burak Kalyoncu.....	i
Publish Date.....	i
20/08/2023	i
1.Introduction.....	1
1.1 Problem definition	1
2. Literature Survey	3
3. Background Information.....	3
3.1 Required software.....	3
3.2 Other software	4
4. Modules.....	4
4.1 Hardware	5
4.2 Software.....	5
4.3 Front End.....	6
4.4 Design Documents	8
5. Metadology	9
6. Risk Analysis.....	42
7. Ethics	43
8. Conclusion	44
8.1 Benefits.....	44
a. Benefits to users :.....	44
b. Benefits to me :	44
8.2 Future Works	45
References.....	46

1.Introduction

1.1 Problem definition

We animal lovers want to help as many animals as possible, but unfortunately, we cannot reach, if we do not know where an animal is in need. I made this application to overcome this lack of information. I believe that this application will be a bridge of cooperation and communication between people and animals and will be useful.

I can explain the working logic of my application as follows; In my application, a home page with animal profiles welcomes you. By clicking on these profiles, you can see detailed information about the animal, including the location information of the animal, that is, you can see the location of the animal on the map by pressing the location button on the map from the selected animal's profile. With this feature, you have the chance to go to that location and help the animal. Another option is that you can adopt the animal if you want. You can adopt the selected animal by clicking the adopt button on its profile. After the animal is adopted, the animal's profile is deleted from the profiles on the home page.

There is an add animal button at the bottom centre of your home page. After clicking on this button, it directs you to the page for creating a new animal profile. On this page you can enter the name of the animal, the sex, type, age, address, and picture of the animal, then entered information is sent to the Firebase database and saved there. In this way a new animal profile is created, and this profile is then added to the home page.

Goals

- My goal in this project is to create a bond, a bridge between people and animals in need. To help animals, to ensure that they have a loving home and family by adopting them. One of the aims of this project is to raise awareness in people about the difficulties experienced by animals.

- Aim

To provide easy registration for users

- Details:

Thanks to the Firebase database used in the application, the user information entered instantly creates a new user profile.

- Benefits:

It provides a fast and secure registration

- Aim

To create and display profiles of animals to be helped

- Details:

Click on the + button on the home page to go to the Create a new animal profile page. Enter the desired information on this page, the name, sex, species, age, address and picture of the animal, the information entered is sent to the Firebase database and saved there and in this way a new animal profile is created, this profile is added to the main page.

- Benefits:

We can view the detailed information of the animals in the profiles.

- Aim

To provide the opportunity to adopt animals

- Details:

By entering the animal profiles on the home page, it offers the opportunity to adopt the animal by clicking the adopt button.

- Benefits:

It is to ensure that animals have a loving home and family by adopting animals.

- -Aim

To visualise the location of the animals to be helped on the map

- -Details:

By clicking on the animal profiles on the main page, we can view its location on the map by clicking the location button on the detailed profile page.

- -Benefits:

To increase the awareness of people in that location and to provide the opportunity to go to the animal close to them and help them.

- Aim

To edit the user profile.

- Details:

You can go to the user profile page by clicking the button at the bottom right of the home page. From here, the user can edit the user name, and add a user profile picture.

- Benefits:

Allowing the user to edit their profile.

2. Literature Survey

When we examine the applications similar to this mobile application, there are similar functions. When we compare this project with other applications, we encounter a simpler, understandable and user-friendly interface. Creating an easy, simple, fast and reliable user profile and creating an animal profile easily and quickly are among the situations that allow the user to use the application with love, fun and comfort. Our application is open to innovations and development according to the new needs and wishes of the user.

1- Apart from the location information in the profiles of the animals created by the users, showing it on the real-world map, there is a lack of projects with this integration, it is aimed to fill this gap.

2- The adoption function is provided for animals to have a warm and loving family.

3- By looking at the animals on the map, it allows people to easily reach help to animals close to their location.

3. Background Information

In this application that I developed using Java language, I use Firebase database and I developed in Android studio.

3.1 Required software

- **Java:**

I developed my mobile application in android studio, java language allows me to create my application easily, I use java because it gives me this opportunity.

- **Android Studio :**

Android Studio is used for mobile application development. Among the reasons why I prefer Android Studio for mobile application development, I can list the following;

It has Android Emulator. Android Studio has an Android Emulator that allows you to test applications on Android devices.

Google Service Integration: Android Studio allows Google services and APIs to be easily integrated. For example, it offers special plugins and templates for fa integration such as Google Maps, Firebase, Google Cloud services.

- **Firebase:**

Firebase can be easily integrated with Android Studio. Firebase has a Real Time Database feature. Firebase ensures that the data in your application is updated instantly with the real-time database. Thanks to this feature, it allows users to establish instant data connection and other users to see the changes instantly.

The Auto Scaling feature provides another usage advantage. Firebase automatically scales according to the traffic density of your application and responds to user demands. This ensures that your application runs smoothly even during periods of heavy traffic.

It has Login and Authentication feature. Firebase allows users to easily register to the application and provides authentication features that allow them to log in with their social media accounts or e-mail. This feature simplifies user management and security.

3.2 Other software

- **AdobeXD :**

For Project Poster.

- **Bitbucket:**

Used for repository.

4. Modules

An embedded system project may contain hardware and software side, an online sales application may contain front-end (for customers) and back-end (for workers). Explain each module in detail (at least half a page for each module, around two pages total). If you will be in a group project explain how much each member will work on a particular module.

4.1 Hardware

Map:

This module determines the location of the animal.

It makes instant location marking on the map using Google Maps API. Pets list is retrieved, and map operations are performed using the GoogleMap object.

4.2 Software

Camera module:

Allows the user to add photos to their profile.

The openGallery() function is called to open the user's gallery and allows the user to select an image.

Map integration:

The application needs to be integrated with a mapping service such as Google so that users can access the map and see the location of the animals.

It is used to display the location of the animal whose location information is entered by the user on a map.

User management:

The app will need a secure system to manage user accounts and permissions, allowing users to create profiles, upload photos and interact with the app. For this we use the Firebase database. Firebase allows the data entered by users to be updated instantly with the real-time database. This allows users to establish an instant data connection and allows other users to see the changes instantly.

Animal database:

The application will need to store data about the animals such as name, breed, age, sex, location, photo. For this we again use the Firebase database. Firebase creates tables of animal information data that users enter to create a new animal profile. With this feature, animal profiles are created.

4.3 Front End

Login Page:

The first page that greets us is the login page. On this page, the user is asked to enter their e-mail address and password. If he/she enters this information correctly, he/she is directed to the main page.

Register Page:

If it is a new user who wants to register, if an account has not been created before, clicking on the "Register for new user" text will be directed to the registration page. On this page, it asks the user to confirm the username, email address, password and password. After these are entered, when the register button is clicked, the new user's profile is automatically registered to the system, by connecting directly to firebase. If the user has registered before, "Already have an account?" Clicking on the text will be directed to the login page.

Home Page:

Profiles of animals created by users appear on the home page. Clicking on the profile of the animals opens a detailed page. There is a plus button in the bottom middle of the page, this button opens the page that allows the creation of an animal profile. In the lower right part, there is the profile button. This button opens the user profile. In the upper right corner, there is an exit button and when clicked, it allows exiting the application.

Animal profiles:

The main page displays the profiles of the animals that users have entered. These profiles include a photo, the animal's name, type, location and age. Users can browse the profiles by clicking on the selected animal's profile and the detailed profile of the animal will open. There is an adopt button on this page for those who want to adopt the animal. When they click on the button, the profile is deleted from the main page because the animal has been adopted. As a second option instead of adoption, help can be provided. By clicking on the location button, you can view the location of the animal on a real map, so you can go to the animal and help it with food, water, medical and anything else it needs.

Map view:

When you enter the animal profiles on the home page, when you click on the location button, the map opens and the location of the animal appears on this map, so users can

identify animals close to their location and go to them and help them with food, water, medical and anything else they need.

Create Animal Profile Page

In the middle of the main page there is an animal profile creation button with a plus icon. When the user clicks on this button, the user is directed to the page where the name, species, age, gender, photo of the animal and the address information of the animal are requested. After the user enters this information, the profile of the new animal appears directly on the main page.

User Profile :

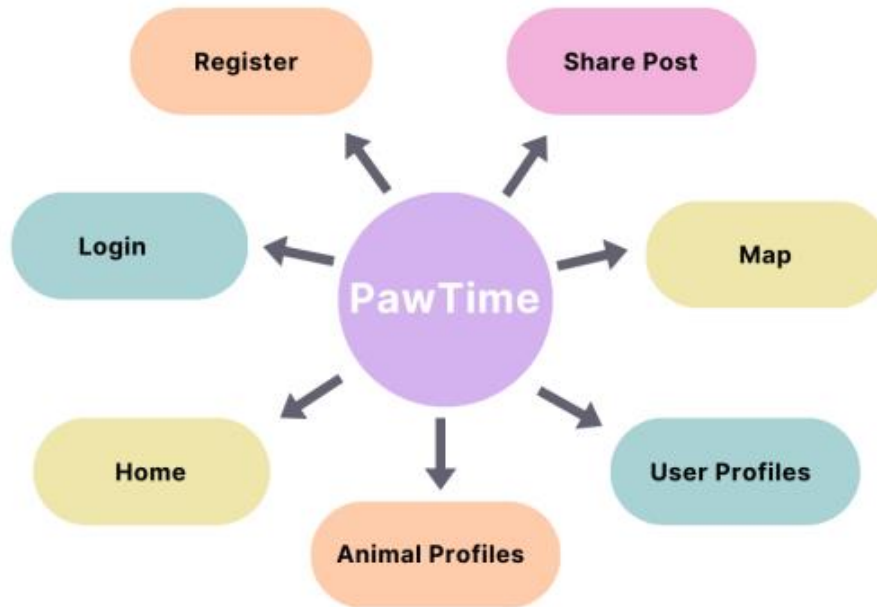
There is a user profile button with a human icon in the lower corner of the homepage. When the user clicks this button, he displays his profile. If the user wants to edit their profile, they can. He can edit his name, choose and upload a photo from his phone's gallery.

Exit Button:

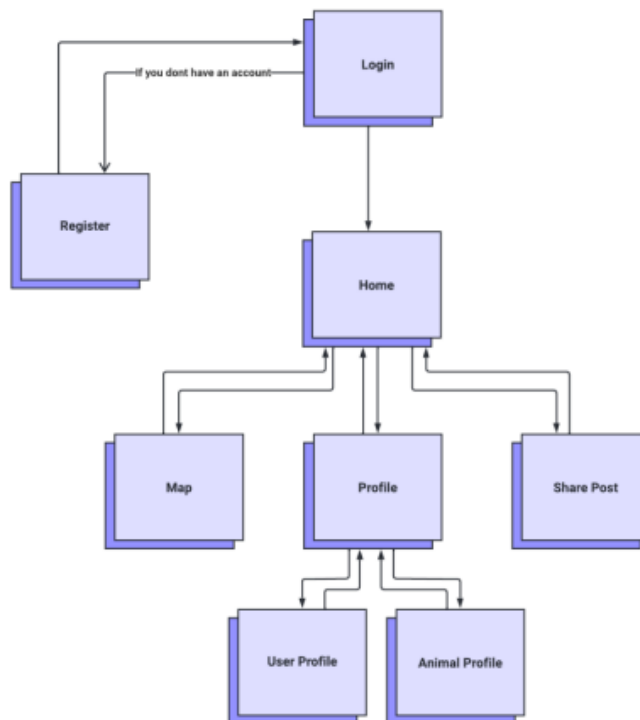
There is an exit button with a door icon in the lower right corner of the home page. When the user clicks this button, he logs out of his account in the application and is directed to the first page, namely the login page.

4.4 Design Documents

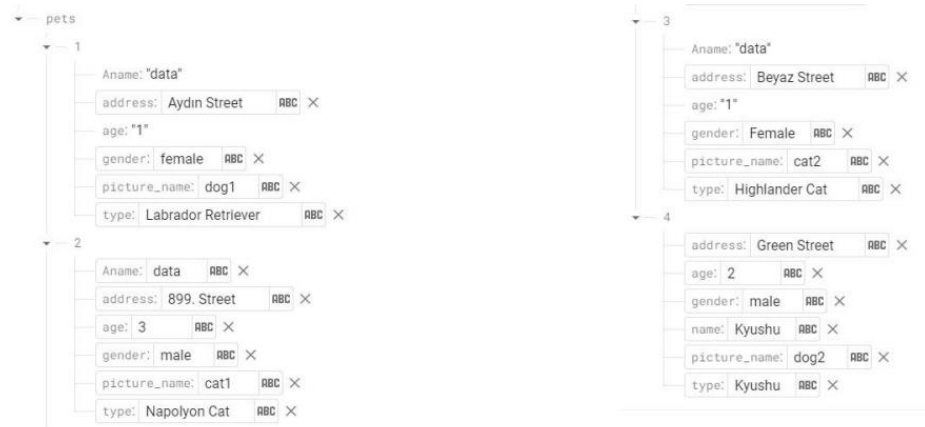
1.Context Diagram



2. Data Flow Diagram



3. Database Schemas



5. Metadology

5.1 Installation process

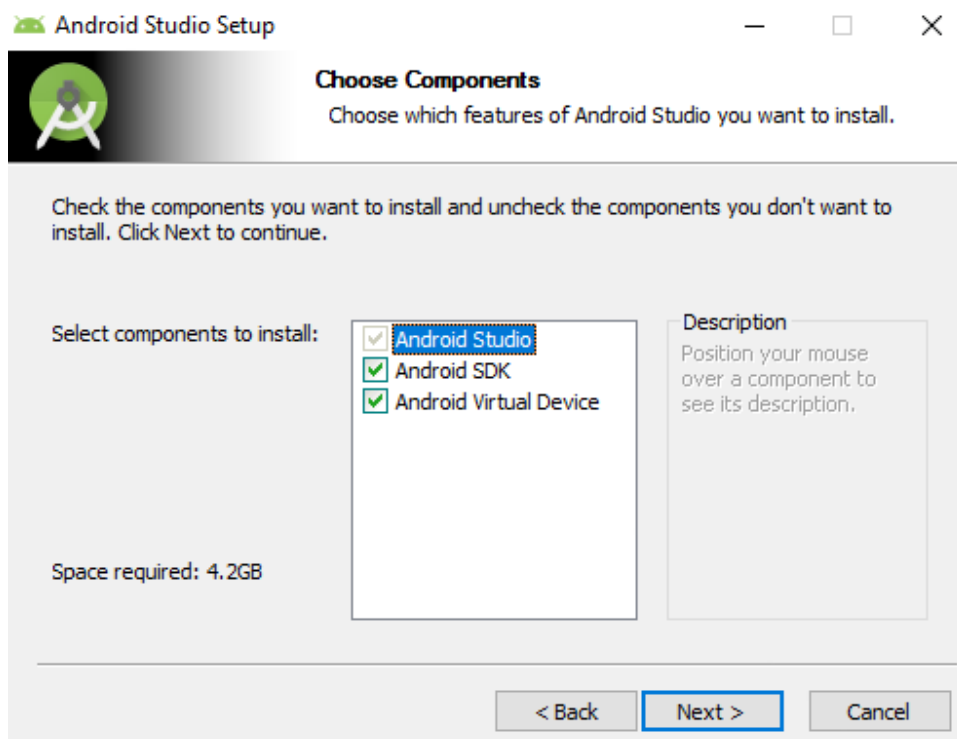
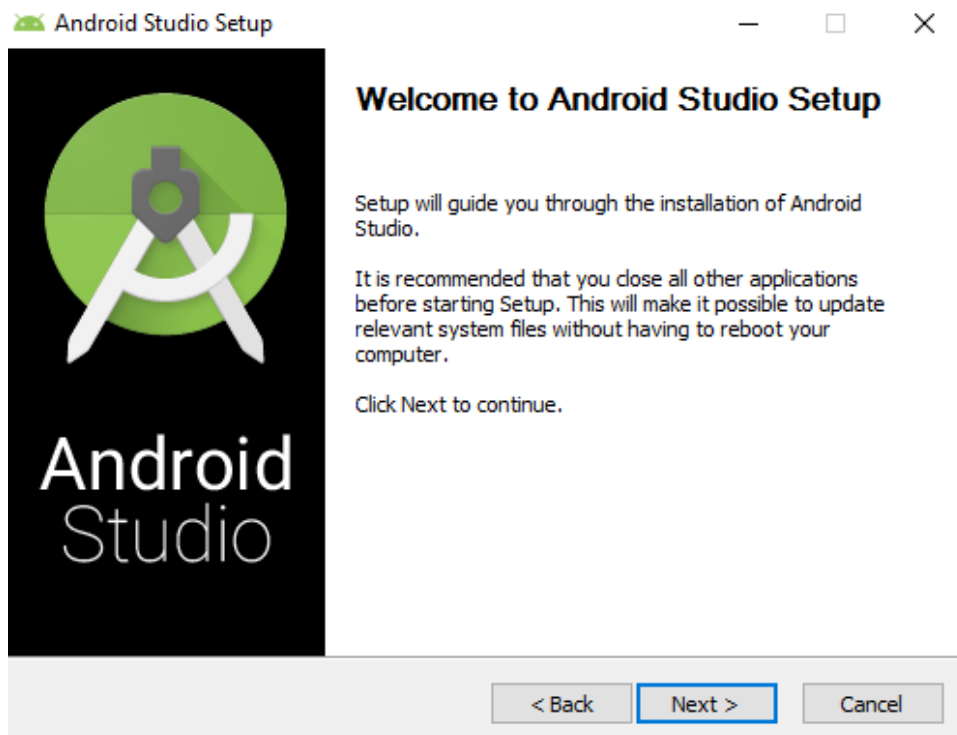
Android Studio:

- Go to the website to download Android Studio:

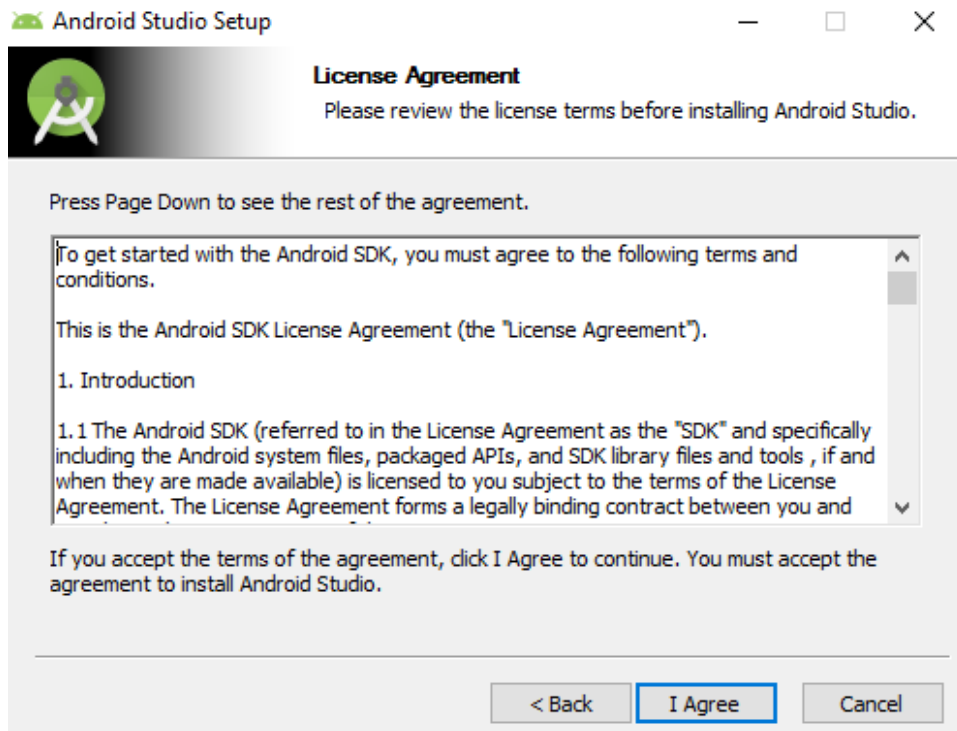
<https://developer.android.com/studio>.

- Select your operating system and click the "Download Android Studio" button.
- Wait for the download to complete.
- Run the downloaded file.
- A list of components that will be required during installation will open. Choose the necessary ones.
- Start Android Studio. After the installation is complete, you will be given the option to launch Android Studio.

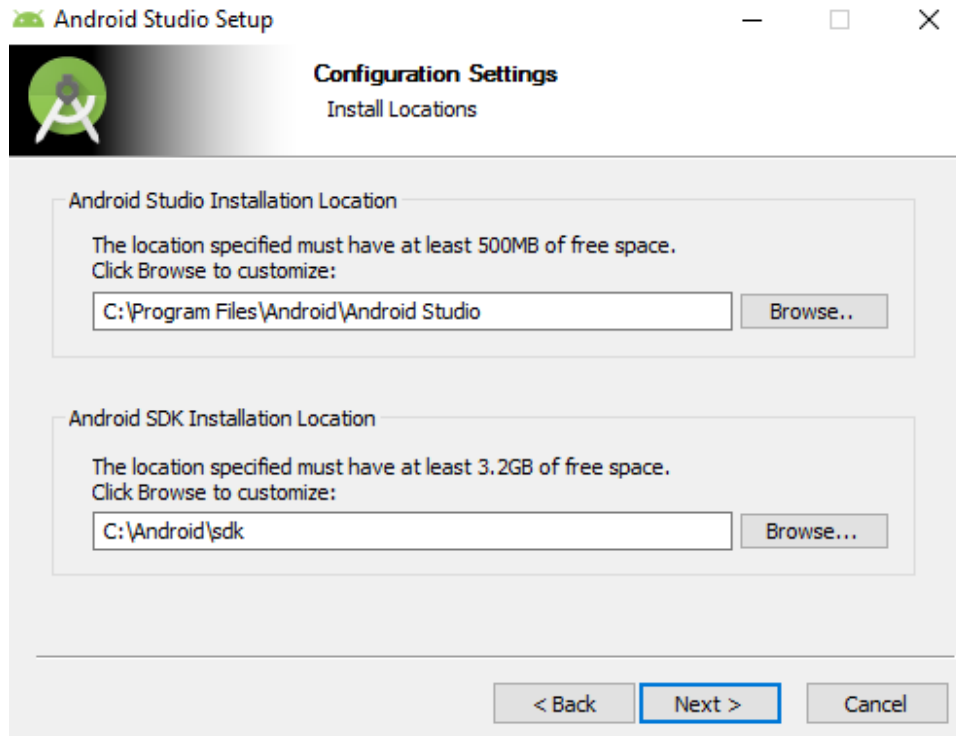
Wait for the installation to complete. Proceed by clicking the "Next" or "Install" button and the necessary files will be installed.



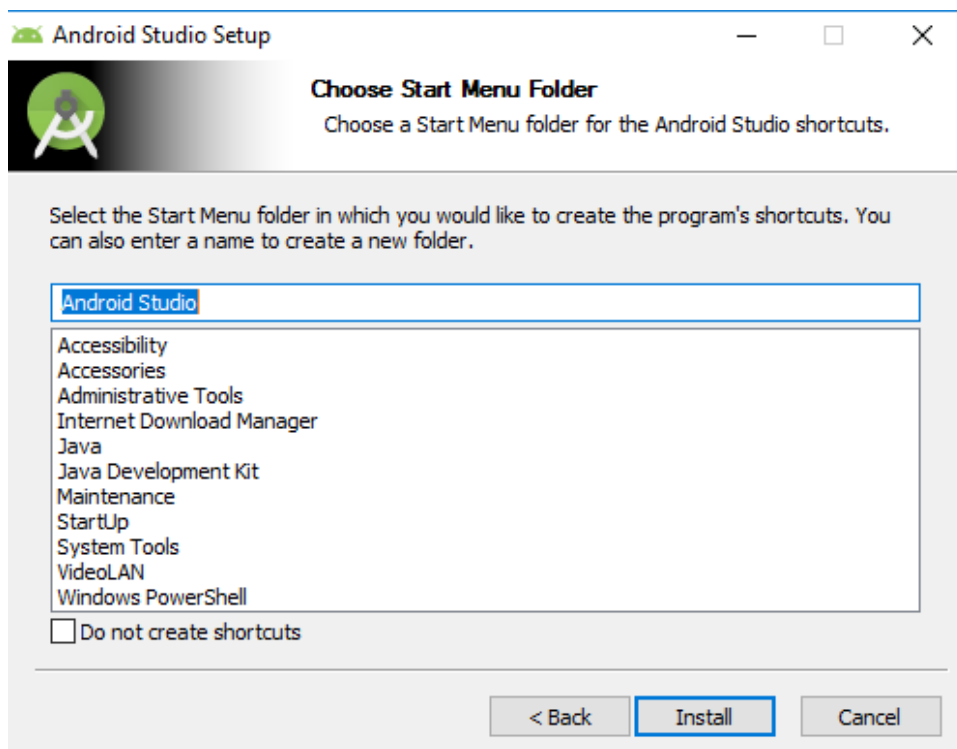
Click the "I Agree" checkbox to confirm the terms and conditions,



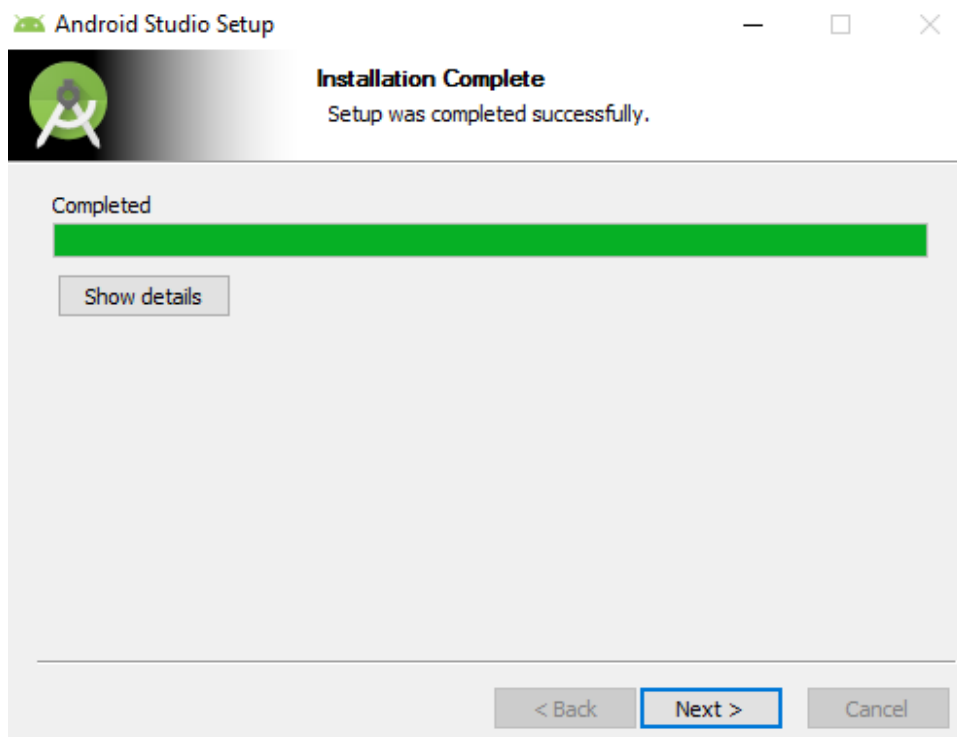
Specify the installation location



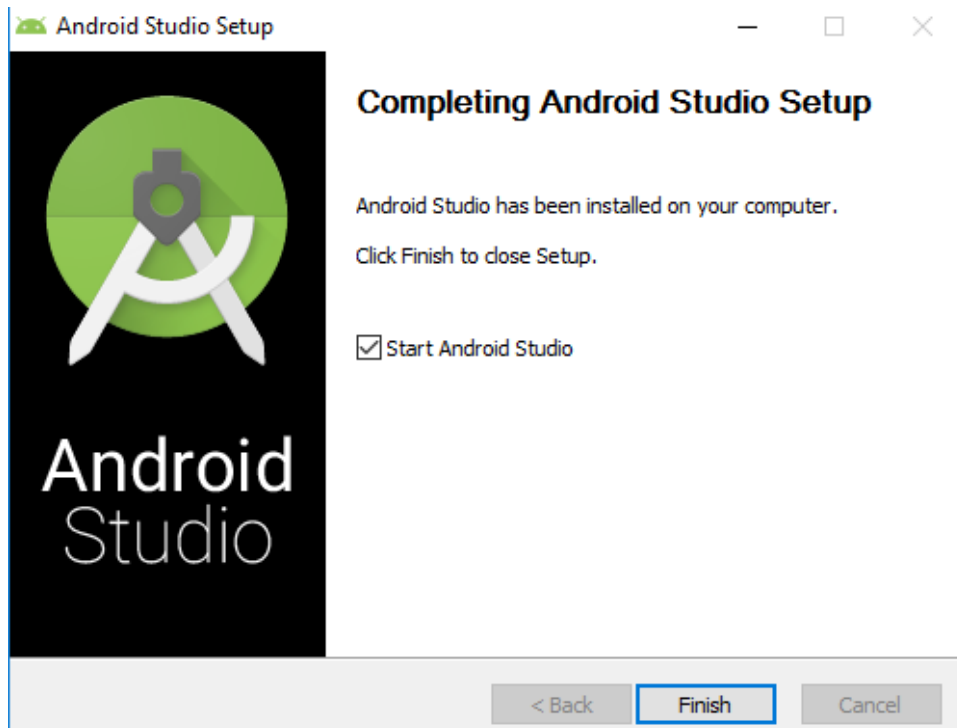
For installation click the install button

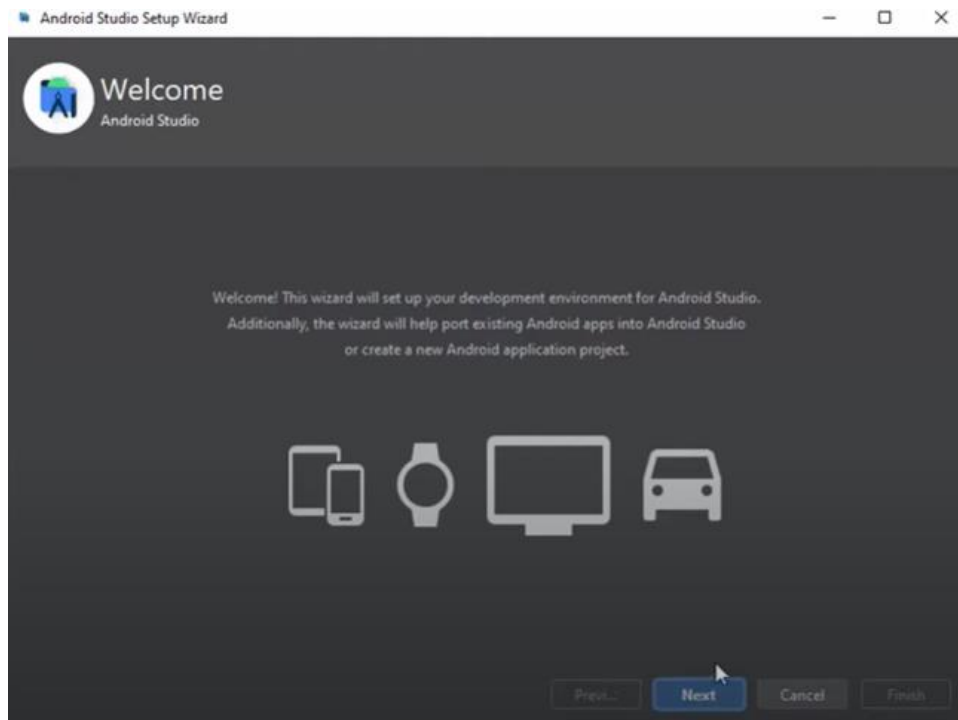


After successful installation click the install button

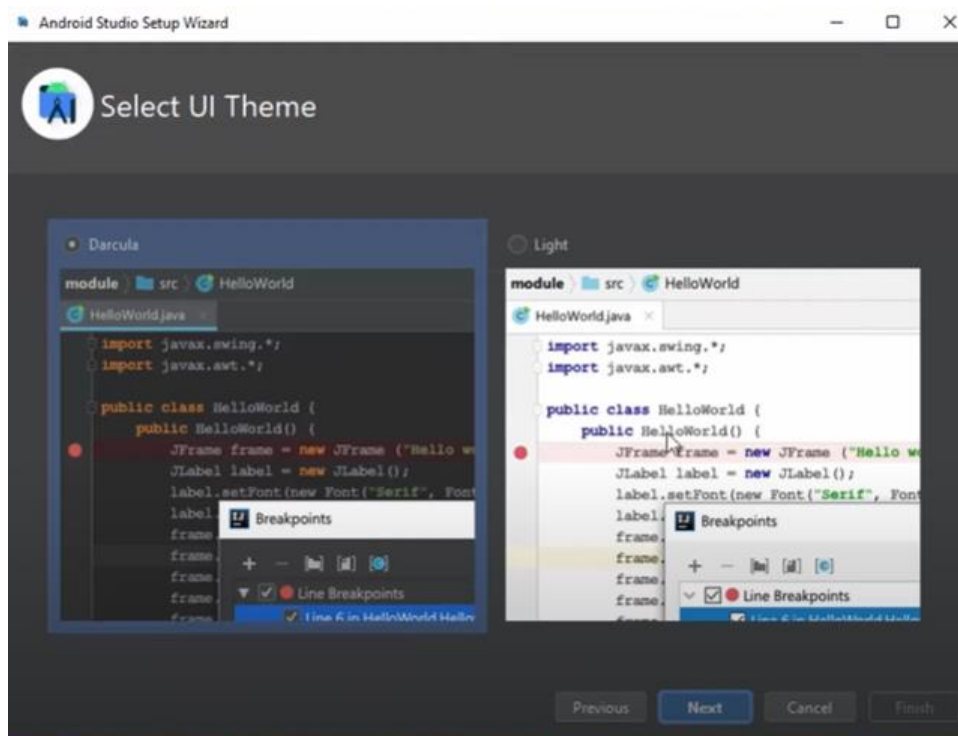


Then check the option and click the finish button

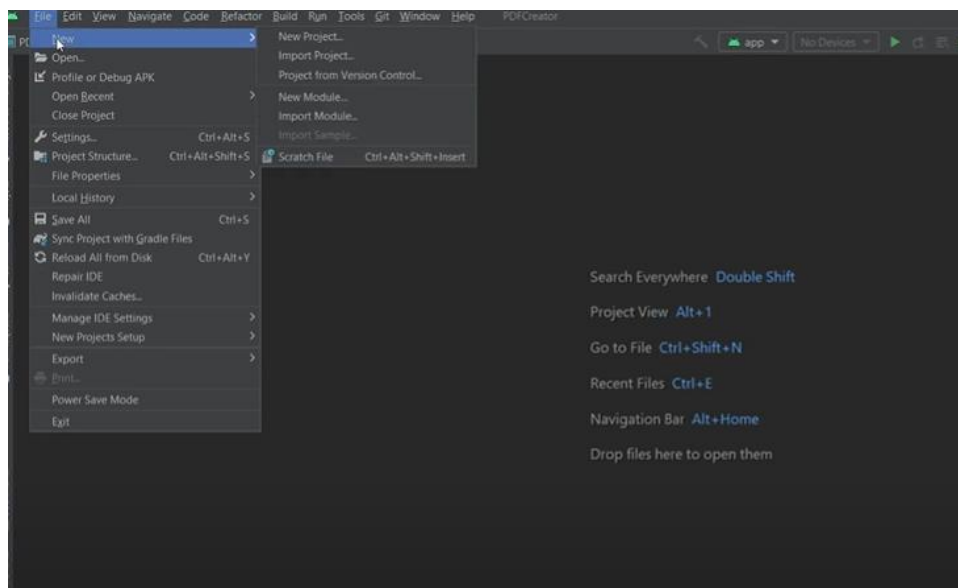




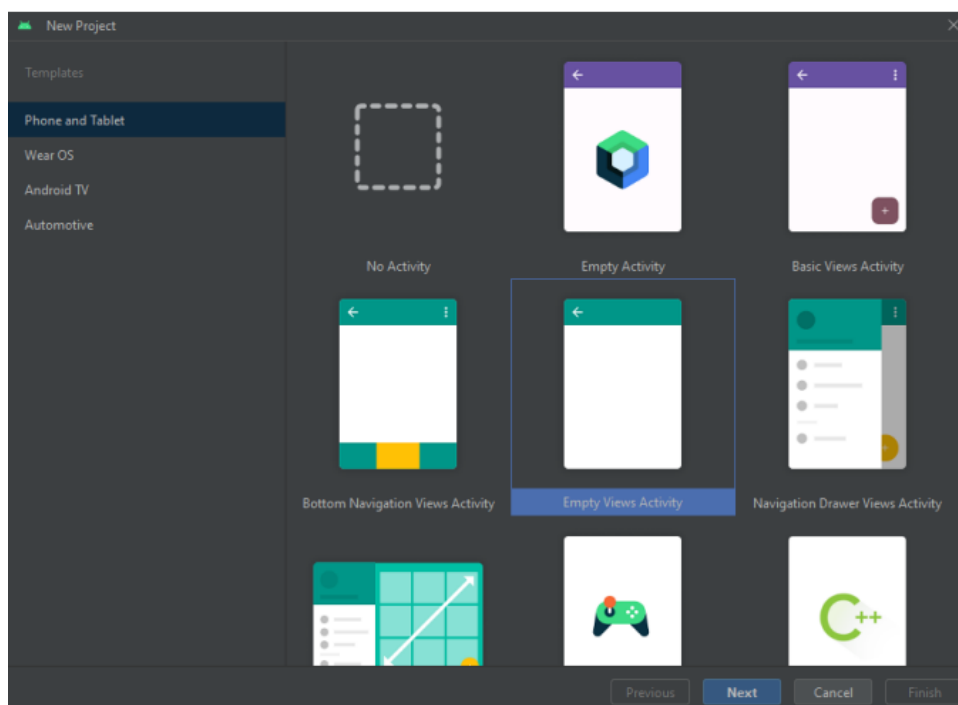
you can choose dark or light theme



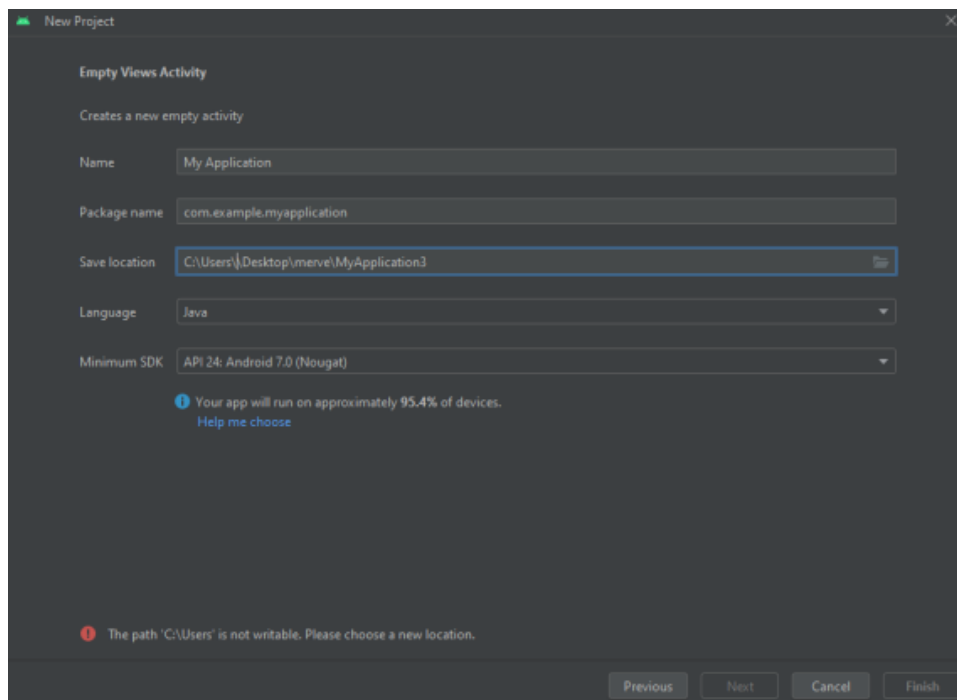
We start a new project. File->New-> New Project



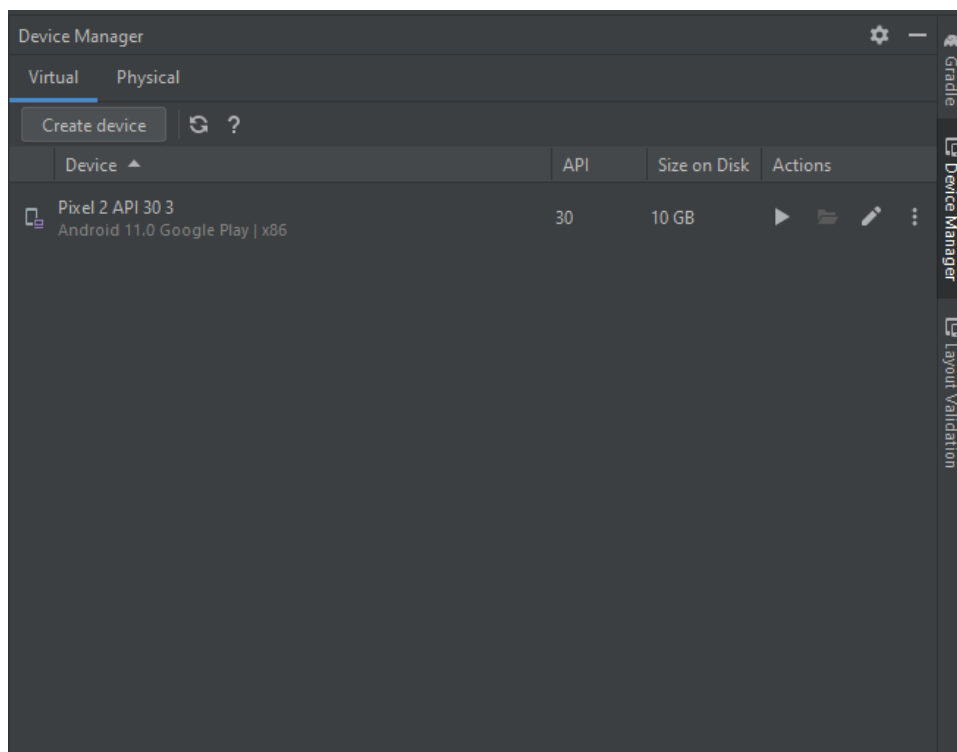
Then choose the Empty View Activity



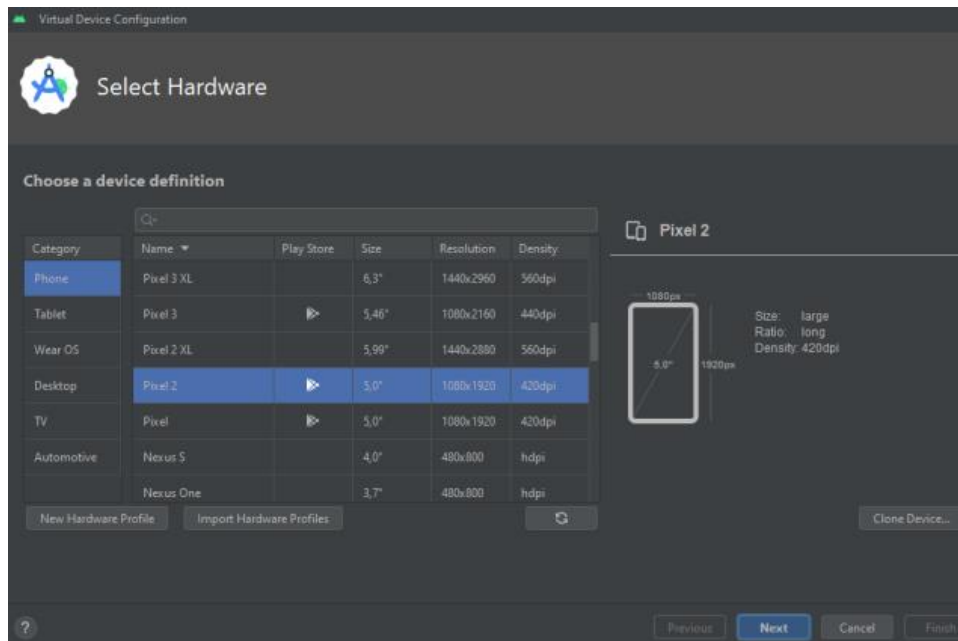
We select the file where we want the project to be found and write the name of the project.



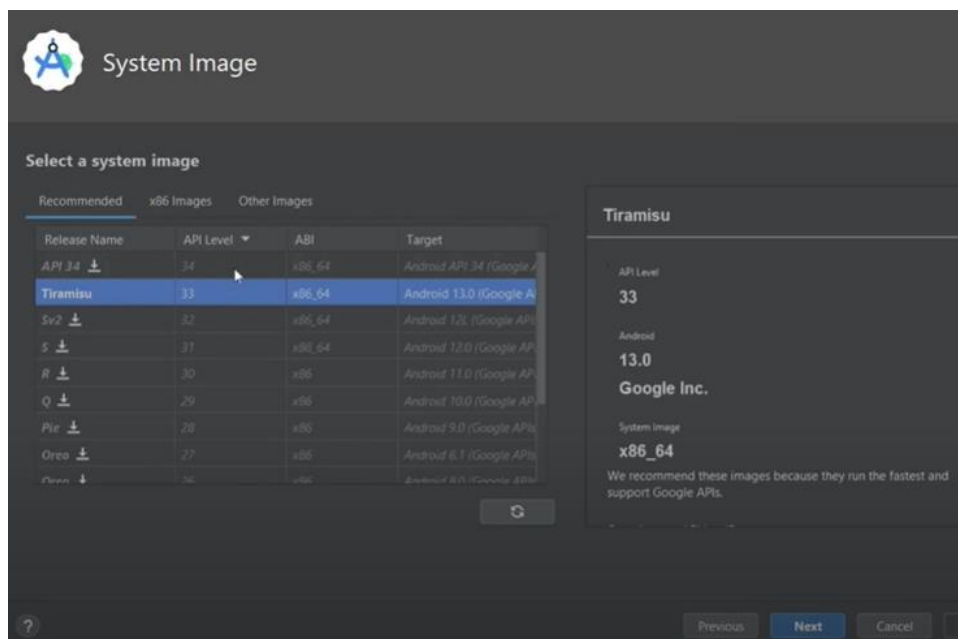
We create a device which is Pixel 2 API 30



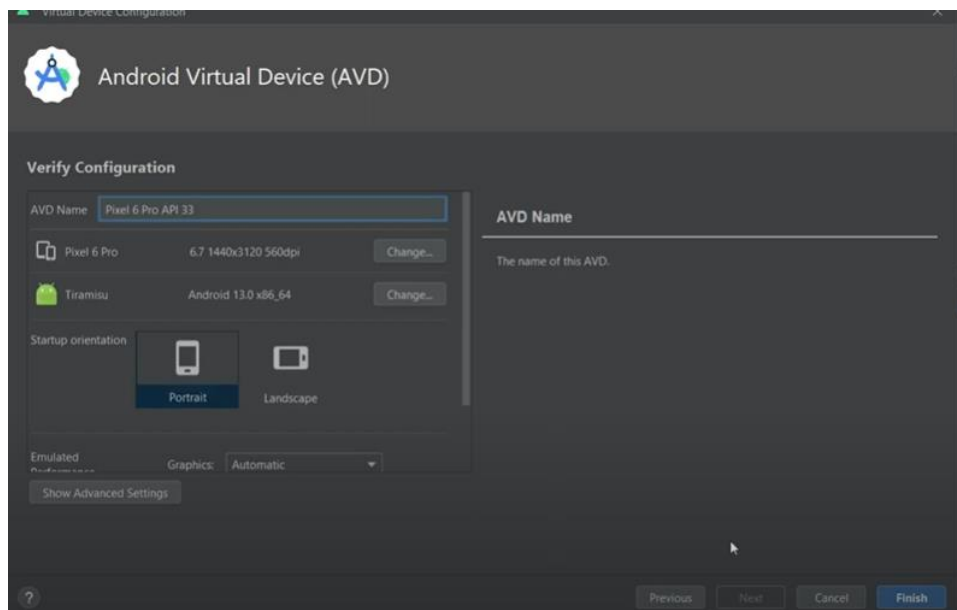
we choose the emulator model we want from here



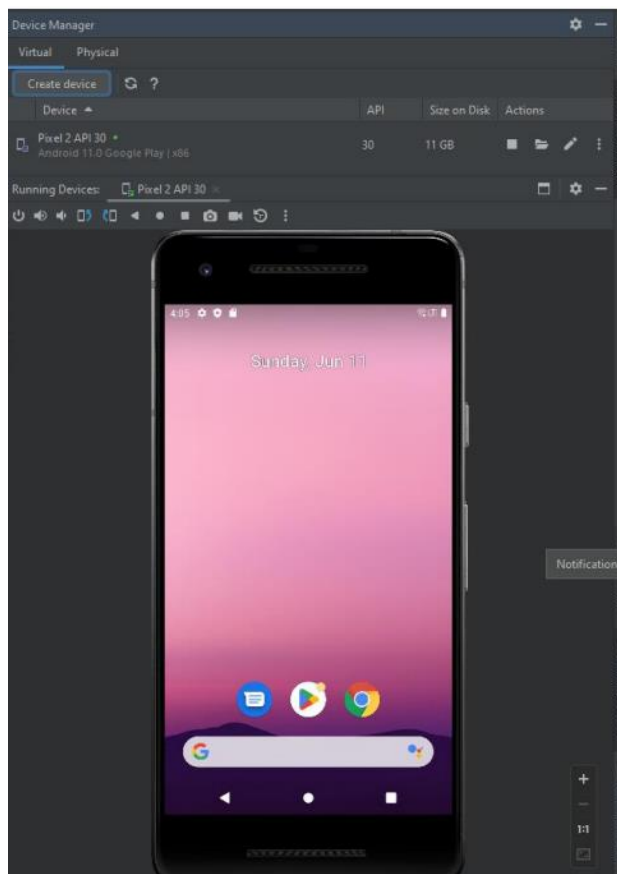
On this page, we download the file that needs to be downloaded for the emulator and select it.



If there is a name we want for the emulator on this page, we write it.



And our emulator is created

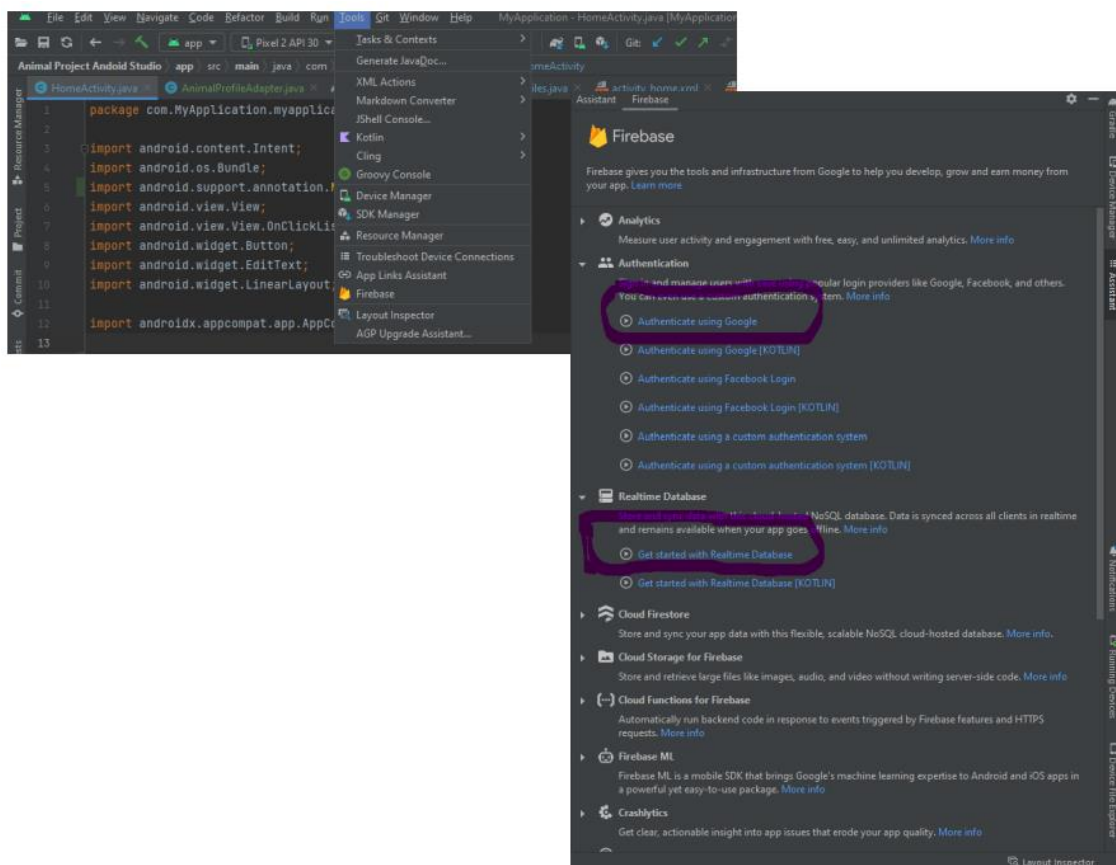


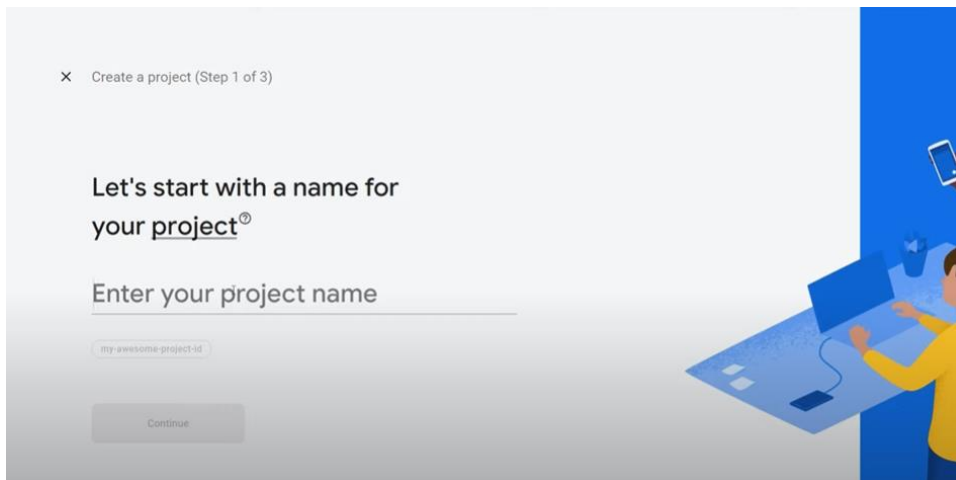
5.2 Firebase

It is a mobile and web application development platform. Submitted by Google.

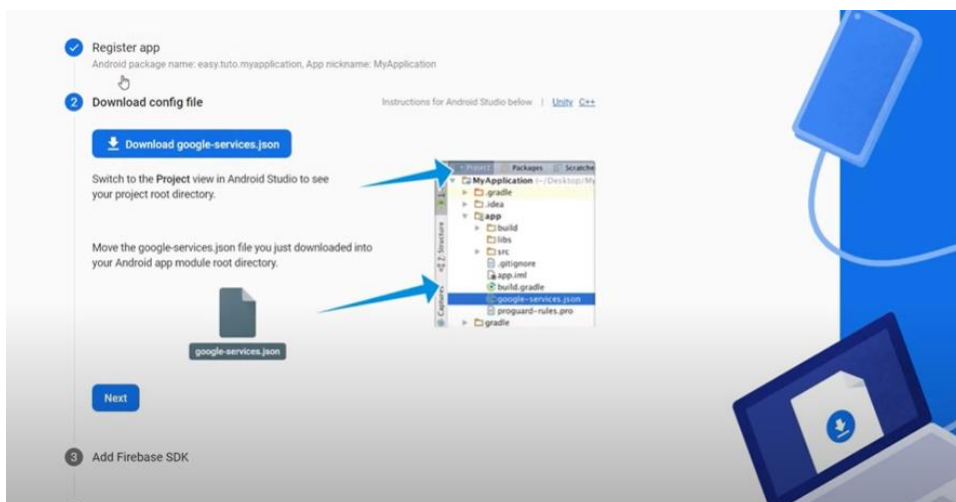
To connect Firebase to your project;

- Go to Firebase site(<https://console.firebase.google.com/>) and login or create an account.
- Click on the add project button and the project we made is selected
- The file "google-services.the"json" is downloaded and saved on the computer.
- Then the Firebase SDK is added.
- Open "gradle (Module: app)" file and add dependencies.
- Later, Firebase Authentication and Firebase Realtime Database are added.

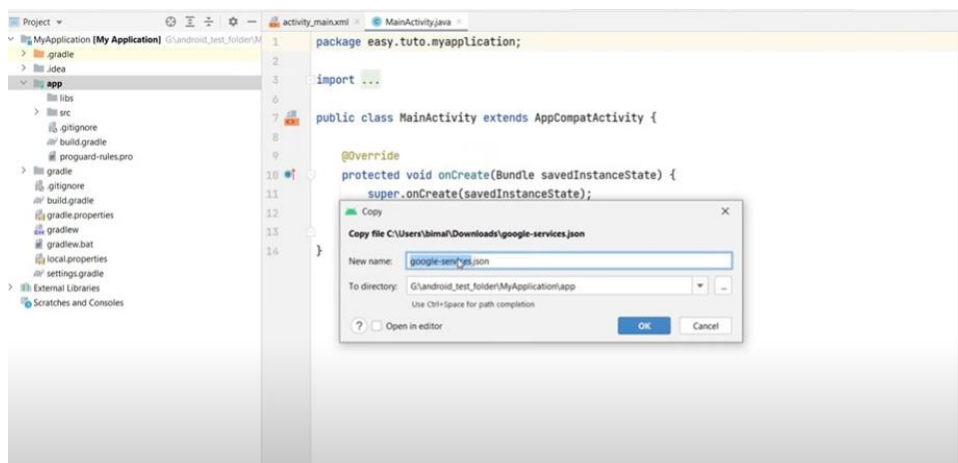
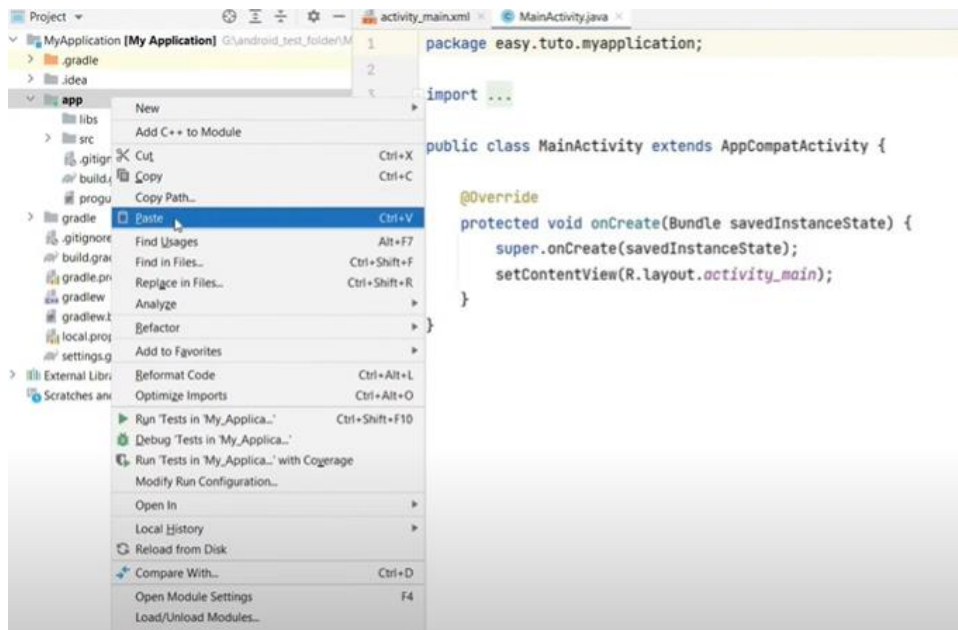




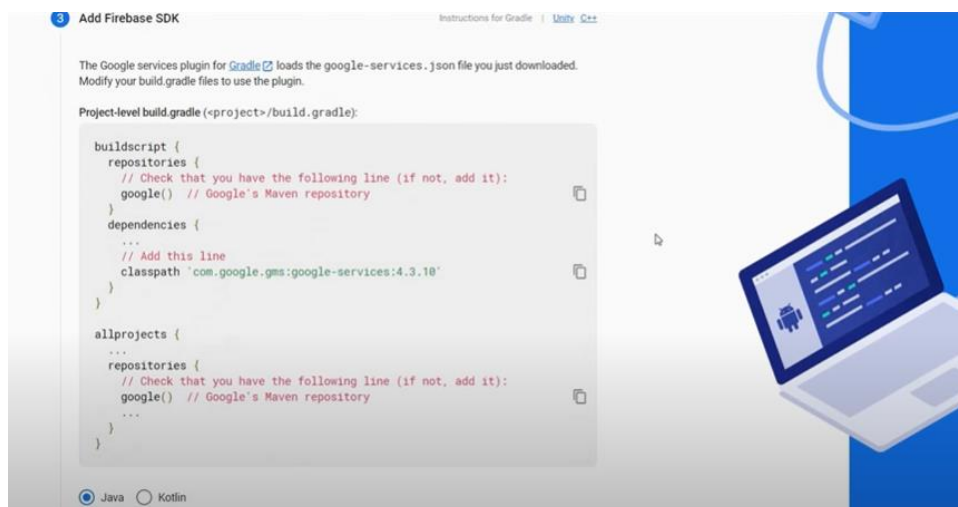
this is firebase in site and here it shows steps to connect to android studio

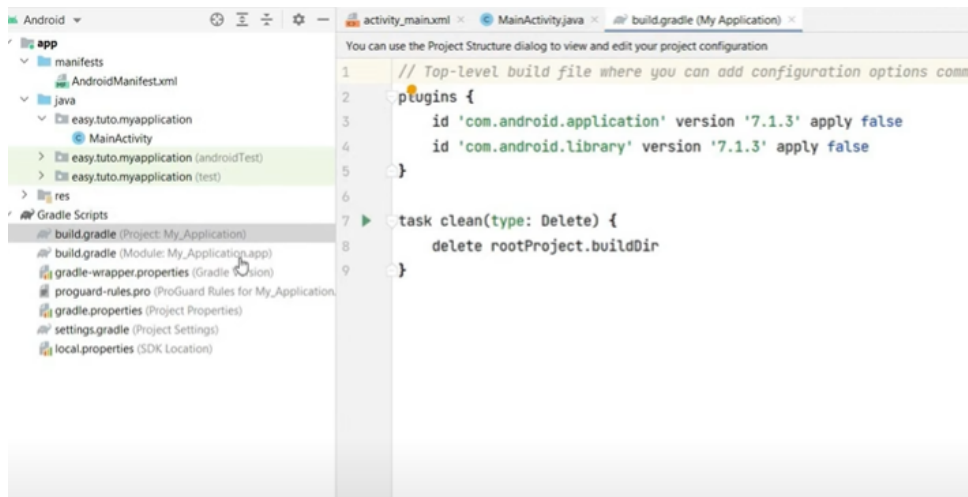


After downloading the google-services.json file from the site, we paste it into the app file in the project.



Then we add the codes shown on the site to the gradle file of the project





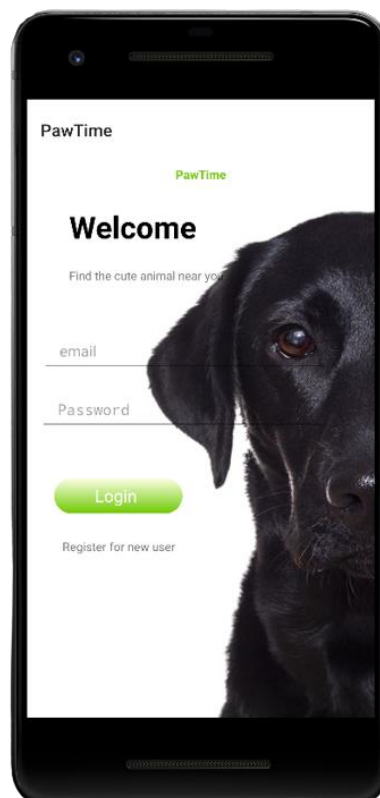
Connecting to Firebase is complete. We can use it by adding the necessary libraries.

5.3 Project Pages

1. Login Page

The first page that appears when we open the application is the login page. If the user has registered to the application before, he enters his e-mail address and password, clicks the login button and the home page opens.

If it has not been registered before, if there will be a new registration, click on the text "Register for new user" and it will be directed to the registration page.



login.xml

```
<EditText
    android:id="@+id/editTextLoginPassword"
    android:layout_width="338dp"
    android:layout_height="51dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:paddingLeft="20dp"
    android:paddingTop="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.219"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.487"
    app:layout_constraintTop_toBottomOf="@+id/editTextLoginUserName"
/>

<Button
    android:id="@+id/buttonLogin"
    android:layout_width="154dp"
    android:layout_height="50dp"
    android:contentDescription=" "
    android:elevation="50dp"
    android:text=" "
    android:background="@android:color/transparent"
    app:backgroundTint="#75C614"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.062"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.612" />
```

XML code of login page. We use the text property for e-mail and password. and there is a button to login. There is a "Register for new user" text to be used to redirect to a deregister page.

login.java

```
1 package com.MyApplication.myapplication;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.content.SharedPreferences;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ProgressBar;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import androidx.annotation.NonNull;
16
17 import com.google.android.gms.tasks.OnCompleteListener;
18 import com.google.android.gms.tasks.Task;
19 import com.google.firebase.auth.AuthResult;
20 import com.google.firebase.auth.FirebaseAuth;
21 import com.google.firebase.auth.FirebaseUser;
22
23 public class LoginActivity extends Activity {
24     EditText edUsername, edPassword;
25     Button btn;
26     TextView tv;
27     FirebaseAuth mAuth;
28     ProgressBar progressBar;
29 }
```

Libraries and variables used in login.java

```

└─ Merve Altınkılıç
@Override
public void onStart() {
    super.onStart();
    FirebaseAuth currentUser = mAuth.getCurrentUser();
    if(currentUser != null){
        Intent intent = new Intent(getApplicationContext(),MainActivity.class);
        startActivity(intent);
        finish();
    }
}

└─ Merve Altınkılıç
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    mAuth = FirebaseAuth.getInstance();
    progressBar = findViewById(R.id.progressBar);
    edUsername = findViewById(R.id.editTextLoginUserName);
    edPassword = findViewById(R.id.editTextLoginPassword);
    btn = findViewById(R.id.buttonLogin);
    tv = findViewById(R.id.textViewNewUser);
    └─ Merve Altınkılıç
    tv.setOnClickListener(new View.OnClickListener() {
        └─ Merve Altınkılıç
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: LoginActivity.this, RegisterActivity.class);
            startActivity(intent);
            finish();
        }
    });
}

```

If the user has previously registered with the application, the code that enables it to be opened.

The code that allows defining texts, buttons etc. in the login xml file to the java file. It opens the Register.Activity page when clicking on the "Register for new user" text

```

└─ Merve Altınkılıç
btn.setOnClickListener(new View.OnClickListener() {
    └─ Merve Altınkılıç
    @Override
    public void onClick(View v) {
        progressBar.setVisibility(View.VISIBLE);
        String username = edUsername.getText().toString();
        String password = edPassword.getText().toString();
        String email = edUsername.getText().toString(); // User entered e-mail

        if (username.length() == 0 || password.length() == 0 ) { //we compare to confirm password
            Toast.makeText(getApplicationContext(), text: "Please Fill All Details", Toast.LENGTH_SHORT).show();
            return;
        }
        mAuth.signInWithEmailAndPassword(email, password)
        └─ Merve Altınkılıç
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            └─ Merve Altınkılıç
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                progressBar.setVisibility(View.GONE);
                if (task.isSuccessful()) {
                    SharedPreferences sharedPref = getPreferences(Context.MODE_PRIVATE);
                    SharedPreferences.Editor editor = sharedPref.edit();
                    editor.putString("username", username);
                    editor.apply();

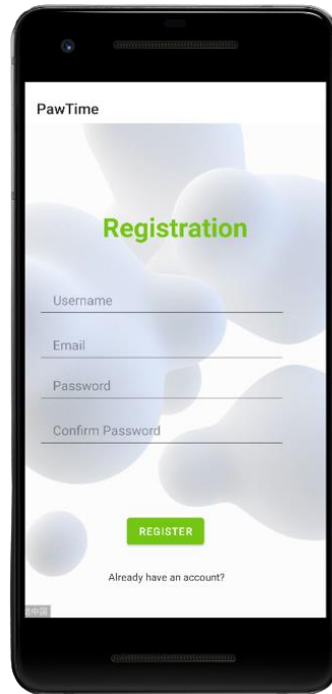
                    Toast.makeText( context: LoginActivity.this, text: "Login Successful", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(getApplicationContext(),MainActivity.class);
                    startActivity(intent);
                    finish();
                } else {
                    System.out.println("@@@@@@ $" +task.getResult());
                    Toast.makeText( context: LoginActivity.this, text: "Authentication failed", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

It confirms the e-mail and password entered by the user by connecting with the firebase database. Checks email and password.

2. Registration Page

"Do you already have an account?" On this registration page that opens after clicking on the option, users are asked to enter username, email, password, and password confirmation. After pressing the register button, it is directed to the main page.



registration.xml

```
11 <ImageView
12     android:id="@+id/imageView33"
13     android:layout_width="match_parent"
14     android:layout_height="match_parent"
15     android:scaleType="centerCrop"
16     android:src="@drawable/wall2"
17     tools:layout_editor_absoluteX="0dp"
18     tools:layout_editor_absoluteY="0dp"
19 />
20 <TextView
21     android:id="@+id/textView5"
22     android:layout_width="wrap_content"
23     android:layout_height="wrap_content"
24     android:fontFamily="@font/markazi_text"
25     android:text="Registration"
26     android:textColor="#75C614"
27     android:textSize="36sp"
28     android:textStyle="bold"
29     app:layout_constraintBottom_toBottomOf="parent"
30     app:layout_constraintEnd_toEndOf="parent"
31     app:layout_constraintHorizontal_bias="0.497"
32     app:layout_constraintStart_toStartOf="parent"
33     app:layout_constraintTop_toTopOf="parent"
34     app:layout_constraintVertical_bias="0.187" />
35
36 <EditText
37     android:id="@+id/editTextRegUserName"
38     android:layout_width="338dp"
39     android:layout_height="51dp"
40     android:ems="10"
41     android:hint="Username"
42     android:inputType="textPersonName"
43     android:paddingLeft="20dp"
44     android:paddingTop="10dp"
45     android:paddingRight="10dp"
46     android:paddingBottom="10dp"
47     app:layout_constraintBottom_toBottomOf="parent"
48     app:layout_constraintEnd_toEndOf="parent"
49     app:layout_constraintHorizontal_bias="0.497"
50     app:layout_constraintStart_toStartOf="parent"
51     app:layout_constraintTop_toTopOf="parent"
52     app:layout_constraintVertical_bias="0.374" />
```

These are the xml codes of the registration page. There is a textvie showing the register header. There are username email, password, comfirm password edittexes that provide data entry from the user. There is a register button that allows the user to register by connecting to Firebase.

```

package com.MyApplication.myapplication;

import static android.app.ProgressDialog.show;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.MyApplication.LoginActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthUserCollisionException;
import com.google.firebase.auth.FirebaseUser;

5 usages  Merve Altınışık
public class RegisterActivity extends AppCompatActivity {

    2 usages
    EditText edUsername, edEmail, edPassword, edConfirm;
    2 usages
    Button btn;
    2 usages
    TextView tv;
    3 usages
    FirebaseAuth mAuth;
    3 usages
    ProgressBar progressBar;

```

Libraries and variables used in register.java

```

@Override
public void onStart() {
    super.onStart();
    FirebaseUser currentUser = mAuth.getCurrentUser();
    if (currentUser != null) {
        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(intent);
        finish();
    }
}

Merve Altınışık
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    mAuth = FirebaseAuth.getInstance();
    edUsername = findViewById(R.id.editTextRegUserName);
    edPassword = findViewById(R.id.editTextRegPassword);
    edEmail = findViewById(R.id.editTextRegEmail);
    edConfirm = findViewById(R.id.editTextRegConfirmPassword);
    btn = findViewById(R.id.buttonRegister);
    tv = findViewById(R.id.textViewExistingUser);
    progressBar = findViewById(R.id.progressBar);

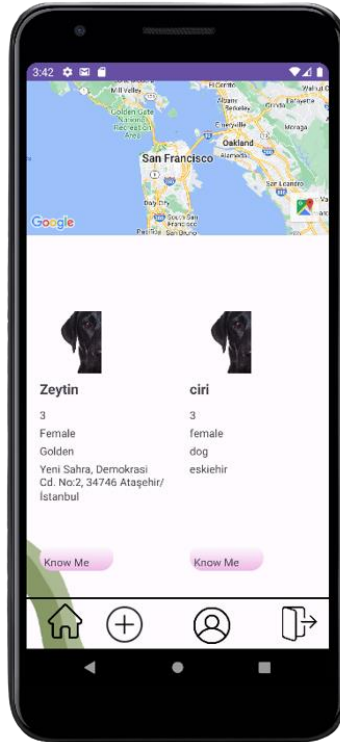
    Merve Altınışık
    tv.setOnClickListener(new View.OnClickListener() {
        Merve Altınışık
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
            startActivity(intent);
        }
    });
}

```

Function that enables it to be opened after the user has logged in.
 Definitions that allow to define texts, buttons etc in the register xml file to the java file; for example "
 edUsername =
 findViewById(R.id.editTextRegUserName); "
 "Already have an account?"
 Function that redirects to login page when textview is clicked

3. Home page

Profiles of animals created by users appear on the home page. Clicking on the profile of the animals opens a detailed page. There is a plus button in the bottom middle of the page, this button opens the page that allows the creation of an animal profile. In the lower right part, there is the profile button. This button opens the user profile. In the upper right corner, there is an exit button and when clicked, it allows exiting the application.



```
11 <ImageView
12     android:id="@+id/imageView33"
13     android:layout_width="match_parent"
14     android:layout_height="match_parent"
15     android:scaleType="centerCrop"
16     android:src="@drawable/wall2"
17     tools:layout_editor_absoluteX="0dp"
18     tools:layout_editor_absoluteY="0dp"
19 />
20 <TextView
21     android:id="@+id/textView5"
22     android:layout_width="wrap_content"
23     android:layout_height="wrap_content"
24     android:fontFamily="@font/markazi_text"
25     android:text="Registration"
26     android:textColor="#75C614"
27     android:textSize="36sp"
28     android:textStyle="bold"
29     app:layout_constraintBottom_toBottomOf="parent"
30     app:layout_constraintEnd_toEndOf="parent"
31     app:layout_constraintHorizontal_bias="0.497"
32     app:layout_constraintStart_toStartOf="parent"
33     app:layout_constraintTop_toTopOf="parent"
34     app:layout_constraintVertical_bias="0.187" />
35
36 <EditText
37     android:id="@+id/editTextRegUserName"
38     android:layout_width="338dp"
39     android:layout_height="51dp"
40     android:ems="10"
41     android:hint="Username"
42     android:inputType="textPersonName"
43     android:paddingLeft="20dp"
44     android:paddingTop="10dp"
45     android:paddingRight="10dp"
46     android:paddingBottom="10dp"
47 />
```

xml code of home page. There is an
ImageView for the background.
<com.google.android.gms.maps.Map
View for map.

there are features for animal profiles;

<androidx.recyclerview.widget.Recy
clerView.

There are buttons to go to the user
page, to add an animal profile and to
log out.

Home.java

```
package com.MyApplication.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import com.google.android.gms.maps.MapInitializer;
```

Libraries used in home.java

```
public class HomeActivity extends AppCompatActivity implements PetsAdapter.OnPetClick {

    // 11 usages
    private MapView mapView;
    // 2 usages
    private ArrayList<PetModel> petModelArrayList = new ArrayList<>();
    // 1 usage
    private ViewGroup rootView;
    // 13 usages
    private MapView largeMapView;
    private LinearLayout profileContainer;
    private DatabaseReference databaseRef;
    Button btnS, btnAnimalCreate;
    // 3 usages
    EditText CreatedAnimalName1, CreatedAnimalType1, CreatedAnimalAge1, CreatedAnimalGender1, CreatedAnimalAddress1, CreatedAnimalPicture1, CreatedAnimalLatitude1, CreatedAnimalLongitude1;
    DatabaseReference mDatabase;
    // 2 usages
    ImageView imageViewS;
    private static final int SPLASH_SCREEN = 2000; // 2 saniye

    // Merve Altınkır *
    public HomeActivity() {
    }
    // Merve Altınkır *
    public HomeActivity(EditText animalName, EditText type, EditText age, EditText gender, EditText address, EditText pictureUrl, EditText latitude, EditText longitude) {
        this.CreatedAnimalName1 = animalName;
        this.CreatedAnimalType1 = type;
        this.CreatedAnimalAge1 = age;
        this.CreatedAnimalGender1 = gender;
        this.CreatedAnimalAddress1 = address;
        this.CreatedAnimalPicture1 = pictureUrl;

        this.CreatedAnimalLatitude1 = latitude;
        this.CreatedAnimalLongitude1 = longitude;
    }
    // Merve Altınkır
    public HomeActivity(String animalNameText, String typeText, String ageText, String genderText, String addressText, String pictureText, String latitudeText, String longitudeText) {
    }
}
```

Variables and properties used for map and animal data

```

Merve Altınışık
public HomeActivity(String animalNameText, String typeText, String ageText, String genderText, String addressText, String pictureText, String latitudeText, String longitudeText) {
}

Merve Altınışık
public HomeActivity(EditText createdAnimalName1, EditText createdAnimalType1, EditText createdAnimalAge1, EditText createdAnimalGender1, EditText createdAnimalAddress1, EditText createdAnimalPicture1, EditText createdAnimalLatitude1, EditText createdAnimalLongitude1) {
}

Merve Altınışık
public String getAnimalName() { return CreatedAnimalName1.getText().toString(); }

Merve Altınışık
public String getType() { return CreatedAnimalType1.getText().toString(); }

Merve Altınışık
public String getAge() { return CreatedAnimalAge1.getText().toString(); }

Merve Altınışık
public String getGender() { return CreatedAnimalGender1.getText().toString(); }

Merve Altınışık
public String getAddress() { return CreatedAnimalAddress1.getText().toString(); }

Merve Altınışık
public String getPicture() { return CreatedAnimalPicture1.getText().toString(); }

Merve Altınışık
public String getLatitude() { return CreatedAnimalLatitude1.getText().toString(); }

Merve Altınışık
public String getLongitude() { return CreatedAnimalLongitude1.getText().toString(); }

Merve Altınışık
public void setAnimalName(String text) { CreatedAnimalName1.setText(text); }

```

Allows users to enter data when creating profiles for animals.

```

2 usages
private RecyclerView rv;
2 usages
private ImageView buttonLogout;
Merve Altınışık *
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);

    rv = findViewById(R.id.petsRV);
    buttonLogout = findViewById(R.id.buttonLogout);
    imageView8 = findViewById(R.id.imageView8);
    buttonAnimalCreate1 = findViewById(R.id.buttonPlus);
    mapView = findViewById(R.id.smallMapView);
    mapView.onCreate(savedInstanceState);
    rootView = findViewById(android.R.id.content);

    Bundle mapViewBundle = null;
    if (savedInstanceState != null) {
        mapViewBundle = savedInstanceState.getBundle( key: "MapViewBundleKey");
    }
    mapView.onCreate(mapViewBundle);
}

Merve Altınışık
imageView8.setOnClickListener(new OnClickListener() {
    Merve Altınışık
    @Override
    public void onClick(View view) {
        Intent intent = new Intent( packageContext: HomeActivity.this, UserProfileActivity.class);
        startActivity(intent);
    }
});

```

This code is for text, buttons etc in Home xml file. It is the code that provides the definition of the java file. For example, in this line, the button that allows the user to log out of his account is defined as buttonLogout in the xml file, we declare it to the java file in this way. buttonLogout = findViewById(R.id.buttonLogout);

The setOnClickListener function is the function that allows to go to the UserProfileActivity, that is, the user page, when the imageView8 element is clicked.


```

Merve Altinçik
buttonLogout.setOnClickListener(new OnClickListener() {
    Merve Altinçik
    public void onClick(View v) {
        FirebaseAuth.getInstance().signOut();
        Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
        startActivity(intent);
        finish();
    }
});

Merve Altinçik
buttonAnimalCreate1.setOnClickListener(new OnClickListener() {
    Merve Altinçik
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), AnimalProfiles.class);
        startActivity(intent);
        finish();
    }
});

```

The buttonLogout function is the function that allows you to go to the LoginActivity, that is, the login page, when the buttonLogout button is clicked.

The buttonAnimalCreate1 function is the function that allows you to go to the AnimalProfiles, that is, the animal profiling page, when the buttonLogout button is clicked.

```

protected void onResume() {
    super.onResume();
    DatabaseReference mDatabase = FirebaseDatabase.getInstance( url: "https://myapplicationanimalproject-default-rtdb.europe-west1.firebaseio.com").getReference();
    mDatabase.child( pathString: "pets").get().addOnSuccessListener(new OnSuccessListener<DataSnapshot>() {
        @Override
        public void onSuccess(DataSnapshot dataSnapshot) {
            if(dataSnapshot.getValue() == null) return;
            ArrayList<PetModel> list = new ArrayList<>();
            HashMap<String, Object> mapList = (HashMap<String, Object>) dataSnapshot.getValue();
            System.out.println("@@@@@@@@II $"+mapList.values());

            for (Object animalProfiles : mapList.values()) {
                if (animalProfiles != null) {
                    HashMap<String, String> map = ((HashMap<String, String>) animalProfiles);
                    if(map != null) {
                        String name = map.get("name");
                        if (name == null)
                            name = map.get("Aname");

                        String pic = map.get("picture_name");
                        if(pic == null)
                            pic = map.get("pictureUrl");
                        String username = map.get("username");
                        if(username == null)
                            username = "No one";

                        String id = map.get("id");
                        if(id == null)
                            id = "";

                        String latitude = map.get("id");
                        if(latitude == null)
                            latitude = "";
                    }
                }
            }
        }
    });
}

```

onResume function that allows displaying the properties of user-created animal profiles

4. Pet Layout

We need an xml file to create the design and frame of the animal profiles on the main page.

```

<Button
    android:id="@+id/showProfilesButton"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:layout_marginTop="1dp"
    android:background="#00000000"
    android:contentDescription="Show Profiles Button"
    android:elevation="20dp"
    android:text="Show Profiles"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@id/toolbar"
    app:titleTextColor="#475136" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Know Me"
    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.051"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.407" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

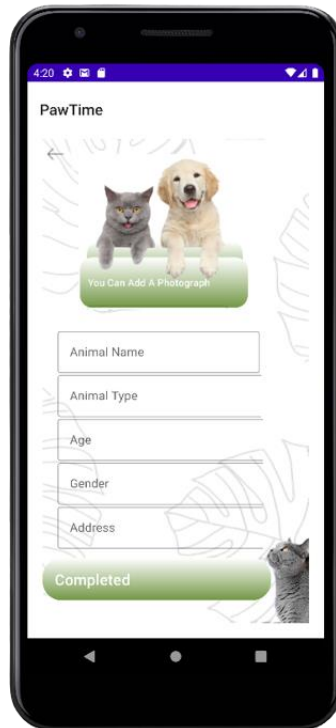
There is a button called showProfilesButton that allows us to open the profile of the animal so that we can see the detailed profile of the animal.

There is an ImageView that allows the image of the animal to be displayed.

There are TextViews in the LinearLayout that allow the animal's properties to be displayed.

5. Create Animal Profile Page

In the middle of the main page there is an animal profile creation button with a plus icon. When the user clicks on this button, the user is directed to the page where the name, species, age, gender, photo of the animal and the address information of the animal are requested. After the user enters this information, the profile of the new animal appears directly on the main page.



xml

```
<com.google.android.material.textfield.TextInputLayout
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="500dp"
    android:layout_marginRight="51dp"
    android:ems="10"
    android:id="@+id/editTextPicture"
    android:layout_width="match_parent"
    android:hint=" "

    android:inputType="text"
    android:paddingLeft="0dp"
    android:paddingTop="0dp"
    android:paddingRight="0dp"
    android:paddingBottom="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"

    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/TextPicture"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/backButton5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="Back" />
```

xml code of the animal profiles page. There is an ImageView for the background. To allow users to enter the characteristics of animals

there are layouts like this

```
<com.google.android.material.textfield.TextInputLayout
```

There is an imageView named imageView5 so that the user can upload images.

```
<androidx.recyclerview.widget.RecyclerView
```

And after the processes are completed, there is a clickable button named buttonAnimal.

.java

```
package com.MyApplication.myapplication;

import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.provider.MediaStore;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import com.MyApplication.LoginActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.Random;
```

Libraries used in Animal Profiles.java

```

public class AnimalProfiles extends AppCompatActivity {
    1 usage
    private static final int PICK_IMAGE_REQUEST = 1;
    1 usage
    private static final int GALLERY_REQUEST_CODE = 123;
    3 usages
    private Bitmap selectedImageBitmap;
    3 usages
    private ImageView uploadImage;

    2 usages
    EditText animalName, type, age, gender, address, picture;
    2 usages
    Button CreateProfileAnimal, backButton4;
    ProgressBar progressBar2;
    1 usage
    DatabaseReference databaseRef;
    4 usages
    DatabaseReference mDatabase;
    10 usages
    String res;

```

Used variable definitions

```

Merve Altıngök
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_animal_profiles);

    animalName = findViewById(R.id.TextName);
    type = findViewById(R.id.TextType);
    age = findViewById(R.id.TextAge);
    gender = findViewById(R.id.TextGender);
    address = findViewById(R.id.TextAddress);
    picture = findViewById(R.id.TextPicture);
    CreateProfileAnimal = findViewById(R.id.buttonAnimal);
    //progressBar2 = findViewById(R.id.progressBar2);

    uploadImage = findViewById(R.id.imageView5);
    backButton4 = findViewById(R.id.backButton5);
    databaseRef = FirebaseDatabase.getInstance().getReference("animal_profiles");
    mDatabase = FirebaseDatabase.getInstance(https://myapplicationanimalproject-default-rtdb-europe-west1.firebaseio.com).getReference();

    uploadImage.setOnClickListener(new View.OnClickListener() {
        Merve Altıngök
        @Override
        public void onClick(View v) { openGallery(); }
    });
}

```

The text in the animal profiles xml file, buttons etc. It is the code that provides the definition of the java file. For example, in this line, the user needs to click on the image to upload the image, and this clicked image is defined as imageView5.

This is how we define it in the java file. `uploadImage = findViewById(R.id.imageView5);`

The `backButton4` function is the function that allows you to go to the HomeActivity, that is, the home page, when you click the `backButton4` button.

```

Merve Altınışık
CreateProfileAnimal.setOnClickListener(new View.OnClickListener() {
    Merve Altınışık
    @Override
    public void onClick(View v) {
        Merve Altınışık
        FirebaseDatabase.getInstance().getReference().child("pets").orderByKey().limitToLast(1).get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {
            Merve Altınışık
            @Override
            public void onComplete(@NonNull Task<DataSnapshot> task) {
                if (!task.isSuccessful()) {
                    Log.e("firebase", "Error getting data", task.getException());
                }
                else {
                    Log.d("firebase", String.valueOf(task.getResult()));
                    Object val = task.getResult().getValue();
                    Random r = new Random();
                    int low = 10;
                    int high = 100000;
                    int result = r.nextInt( (bound: high-low) + low);
                    if(val == null)
                        val = result;
                    res = val.toString();
                    if(res.contains("=")){
                        res = res.substring(1, res.indexOf('='));
                        res = "" + (Integer.parseInt(res) + 1);
                        Log.d("out", res);
                    }
                }
            }
        });
    }
}

```

By connecting to the firebase database, it creates a table in the realtime database and receives the created data.

```

Merve Altınışık *
mDatabase.child("user").addValueEventListener(new ValueEventListener() {
    2 usages Merve Altınışık *
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.exists()) {
            for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                TextView textViewName = findViewById(R.id.textViewName);
                TextView textViewAge = findViewById(R.id.textViewAge);
                TextView textViewGender = findViewById(R.id.textViewGender);
                TextView textViewType = findViewById(R.id.textViewType);
                TextView textViewAddress = findViewById(R.id.textViewAddress);
                ImageView profileImage = findViewById(R.id.profileImage);

                String name = snapshot.child("name").getValue(String.class);
                String age = snapshot.child("age").getValue(String.class);
                String gender = snapshot.child("gender").getValue(String.class);
                String type = snapshot.child("type").getValue(String.class);
                String address = snapshot.child("address").getValue(String.class);
                String pictureUrl = snapshot.child("pictureUrl").getValue(String.class);

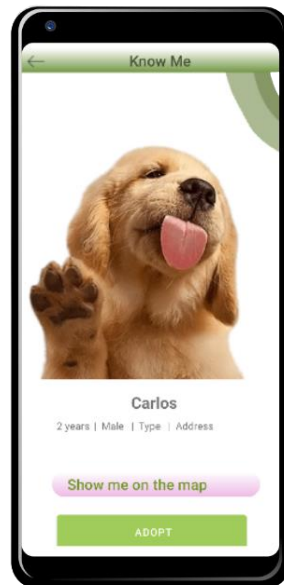
                textViewName.setText(name);
                textViewAge.setText(age);
                textViewGender.setText(gender);
                textViewType.setText(type);
                textViewAddress.setText(address);
                Picasso.get().load(pictureUrl).into(profileImage);
            }
        }
    }
}

```

It allows the user to view the data entered and also allows the user to upload an image for the animal.

6. Animal Profiles:

The main page displays the profiles of the animals that users have entered. These profiles include a photo, the animal's name, type, location and age. Users can browse the profiles by clicking on the selected animal's profile and the detailed profile of the animal will open. There is an adopt button on this page for those who want to adopt the animal. When they click on the button, the profile is deleted from the main page because the animal has been adopted. As a second option instead of adoption, help can be provided. By clicking on the location button, you can view the location of the animal on a real map, so you can go to the animal and help it with food, water, medical and anything else it needs.



xml

```
<LinearLayout
    android:layout_width="270dp"
    android:layout_height="58dp"
    android:layout_marginTop="8dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewName">

    <TextView
        android:id="@+id/textViewAge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="6dp"
        android:text="2 years"
        android:textSize="14sp" />

    <TextView
        android:id="@+id/dizayn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="| "
        android:textSize="14sp" />

    <TextView
        android:id="@+id/textViewGender"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="6dp"
        android:text="Male"
        android:textSize="14sp" />

    <Space
        android:layout_width="4dp"
        android:layout_height="wrap_content" />
```

The xml code of the pet profiles page. There is an ImageView for the background.

ImageView feature is available to display the picture of the animal.

There are TextViews in LinearLayout for ordering the animal's properties.

There is a button that opens the map when clicked, and there is also an adopt button that allows the user to adopt the animal.

.java

```
package com.MyApplication.myapplication;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Objects;
7 usages  ⚡ Merve Altınçik *
public class PetProfileActivity extends AppCompatActivity {
    2 usages
    Button adoptButton,buttonBackPetProfile1;
    2 usages
    Button locationButton;
    2 usages
    ImageView profileImage;
    2 usages
    TextView textViewName;
    2 usages
    TextView textViewAge;
    2 usages
    TextView textViewGender;
    2 usages
}
```

Libraries used in Pet Profiles java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_pet_profile);
    adoptButton = findViewById(R.id.adoptButton);
    locationButton = findViewById(R.id.locationButton);
    profileImage = findViewById(R.id.profileImage);
    textViewName = findViewById(R.id.textViewName);
    textViewAge = findViewById(R.id.textViewAge);
    textViewGender = findViewById(R.id.textViewGender);
    textViewType = findViewById(R.id.textViewType);
    textViewAddress = findViewById(R.id.textViewAddress);
    adoptedUser = findViewById(R.id.adoptedUser);
    buttonBackPetProfile1 = findViewById(R.id.buttonBackPetProfile1);
    separatorView = findViewById(R.id.separatorView);

    ⚡ Merve Altınçik *
    buttonBackPetProfile1.setOnClickListener(new View.OnClickListener() {
        ⚡ Merve Altınçik *
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: PetProfileActivity.this, HomeActivity.class);
            startActivity(intent);
            if (textViewAddress.getText().length() > 50) {
                separatorView.setVisibility(View.VISIBLE);
            } else {
                separatorView.setVisibility(View.GONE);
            }
        }
    });
};
```

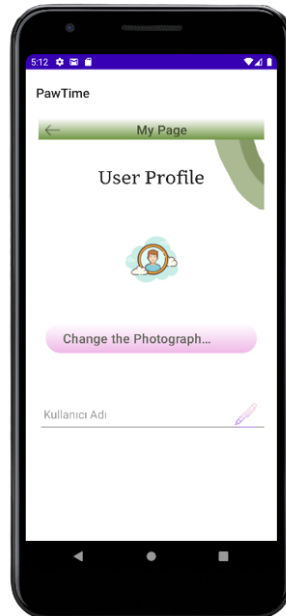
this code is for text, buttons etc in pet profile xml file. It is the code that provides the definition of the java file. For example, this line contains a button that allows the user to return from the page. It is defined as buttonBackPetProfile1 in the xml file, this is how we declare it to the java file.

buttonBackPetProfile1 = findViewById(R.id.buttonBackPetProfile1);

buttonBackPetProfile1 has setOnClickListener function. This function is the function that allows to go to the previous page when the buttonBackPetProfile1 button is clicked.

7. User Profile :

There is a user profile button with a human icon in the lower corner of the homepage. When the user clicks this button, he displays his profile. If the user wants to edit their profile, they can. They can edit their name, choose, and upload a photo from their phone's gallery.



xml

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="44dp"
    android:layout_height="37dp"
    android:elevation="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/back5" />
```

```
<Button
    android:id="@+id/buttonBackPetProfile2"
    android:layout_width="76dp"
    android:layout_height="80dp"
    android:layout_marginEnd="308dp"
    android:background="@android:color/transparent"
    android:contentDescription=""
    android:elevation="22dp"
    android:text=""
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar2"
    android:layout_width="411dp"
    android:layout_height="33dp"
    android:background="@drawable/greenbackground"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />
```

androidx.constraintlayout.widget.ConstraintLayout

The xml code of the user profiles page. There is an ImageView for the background.

ImageView feature is available for displaying the user's picture.

There is a button for the user to change the picture.

<androidx.constraintlayout.widget.ConstraintLayout is available, which when clicked allows the user's name to be edited

```

buttonBackPetProfile2 = findViewById(R.id.buttonBackPetProfile2);
name.setText(usernameFromShared);
imageViewProfile = findViewById(R.id.imageViewProfile);

Merve Altınçik
buttonBackPetProfile2.setOnClickListener(new View.OnClickListener() {
    Merve Altınçik
    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: UserProfileActivity.this, HomeActivity.class);
        startActivity(intent);
    }
});

Merve Altınçik
imageViewProfile.setOnClickListener(new View.OnClickListener() {
    Merve Altınçik
    @Override
    public void onClick(View v) { openGallery(); }
});

Merve Altınçik
updatePictureButton.setOnClickListener(new View.OnClickListener() {
    Merve Altınçik
    @Override
    public void onClick(View view) { openGallery(); }
});

Merve Altınçik
save.setOnClickListener(new View.OnClickListener() {
    Merve Altınçik
    @Override
    public void onClick(View view) {
        String nname = name.getText().toString().trim();
        if (!nname.isEmpty()) {
            prefs.edit().putString("username", nname).apply();
        }
    }
});

```

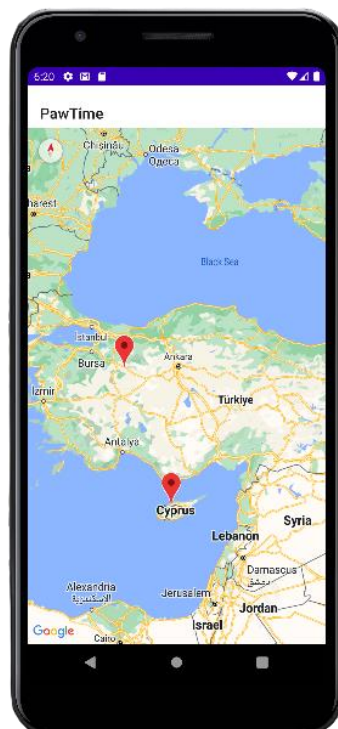
There is the buttonBackPetProfile2.setOnClickListener function that allows you to return to the previous page when you click the buttonBackPetProfile2 button.

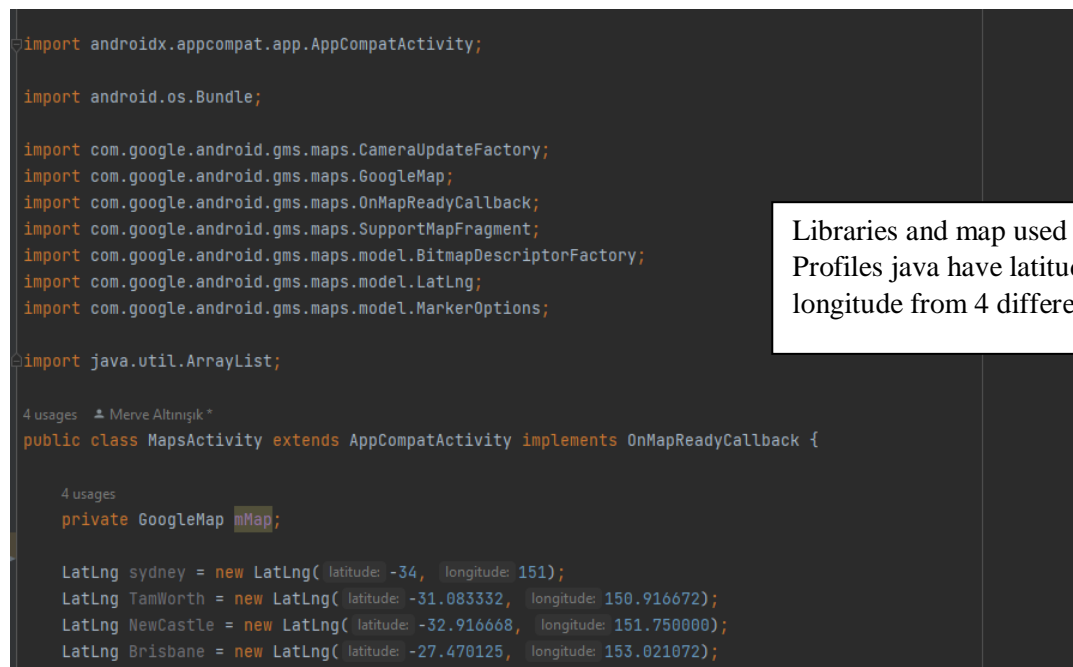
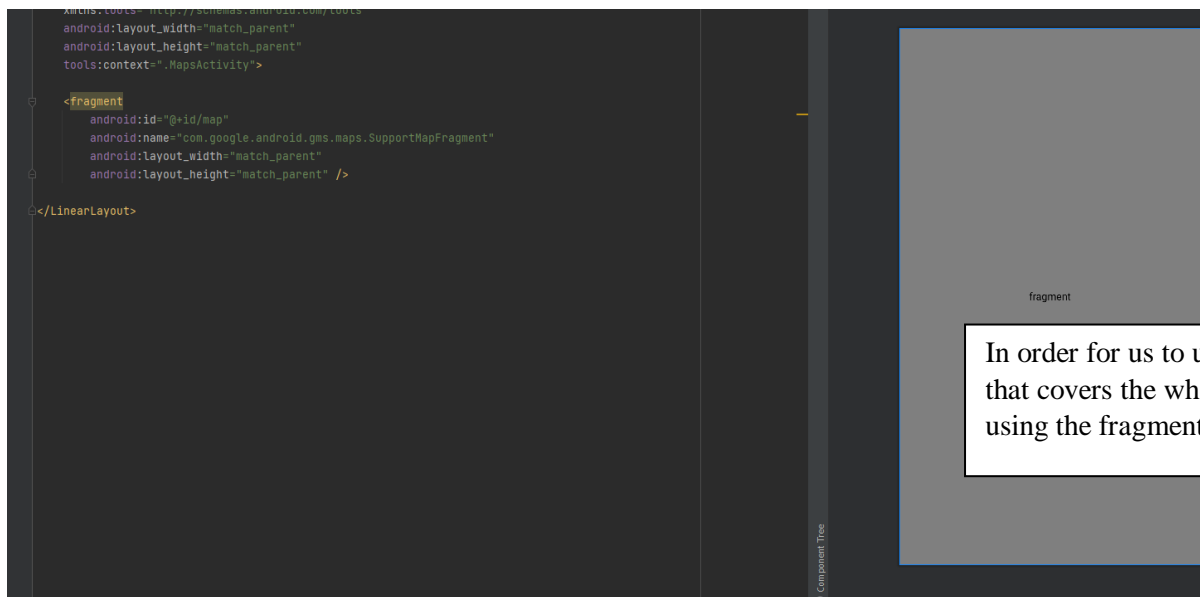
There is the imageViewProfile.setOnClickListener function that allows it to select a photo from the gallery.

There is a save.setOnClickListener function that allows the user to edit their name

8. Map view:

When you enter the animal profiles on the home page, when you click on the location button, the map opens and the location of the animal appears on this map, so users can identify animals close to their location and go to them and help them with food, water, medical and anything else they need.





6. Risk Analysis

- **Technical Difficulties:**

The uploaded photo for the user profile and the animal profile should be uploaded in a secure way that can handle the large volume. Correctly adjust and position the uploaded photo.

- **Privacy And Security:**

User information and location data may be accessed or misused, including photos selected and uploaded by the user from the gallery when creating their own profile page and animal profile. Confidentiality and security measures are in place to protect user data.

- **Legal And Ethical Issues:**

There may be legal and ethical issues that need to be considered in advance to avoid legal problems. Permission is obtained from people before their photos are uploaded to the application, and the application does not use user data for illegal or unethical purposes.

- **Capacity**

Firebase's automatic scaling feature saves this situation in case of any problems that may arise due to increased demand and new users. Firebase automatically scales according to the traffic density in the application and responds to user demands. This ensures that your app runs smoothly even during peak traffic periods.

- **Technology Compatibility:**

Android Studio environment, Java language, integration with firebase database

- **reliability**

It has Login and Authentication feature. Firebase allows users to easily sign up for the app and provides authentication features that allow secure and fast login with their social media account or email. This feature simplifies user management and security.

- **Fast Integration And Data Update**

Firebase can be easily integrated with Android Studio in a few simple steps. Firebase has Realtime Database feature. Firebase provides instant updating of data in your app with real-time database. It allows users to establish instant data connection and other users to see changes instantly. Thanks to this feature, users can quickly send their own data. At the same time, new animal profiles can be created quickly thanks to this feature.

- **Third-party Library Dependencies:**

Relying on third-party platforms like Android Studio may have risks such as potential compatibility issues, license restrictions, and lack of support.

For this, check the compatibility of the license terms. Make regular updates. Have a backup contingency plan in case the unexpected happens.

- **User Preference And Interaction:**

The risk that the application is not liked or used by users.

by users, individuals, which may limit its effectiveness in helping animals. The application is designed to encourage and support the participation and interaction of all animal-loving users who love and are interested in animals.

7. Ethics

- Ensuring animal welfare and protecting animal rights:

It is important to ensure that the use of the app does not harm animals in any way. Animals should not be harmed in any way by the use of the app. Any actions taken as a result of the use of the app must protect the interests and rights of animals.

- Protection of the privacy of individuals:

In the app, users use their own photos for use on their profile page, and creating an animal profile involves uploading photos of animals. It is important to consider the privacy of the people who may appear in the photos. They must give their consent before using their images to obtain the relevant photos.

- Ensuring the accuracy of the information:

It is important to ensure the accuracy of the information provided for security. It must be verified through the application that it is accurate, reliable and up-to-date.

- Privacy and data protection:

To ensure that the data of users using the application is protected. Explicit consent is obtained from individuals for the collection and processing of users' data and their data is used in accordance with applicable data protection regulations.

- Respect the rights of others:

The rights of people who may own the land where the animals seen on the street and wanted to be implemented or the rights of these people should be respected.

- User Experience and Ease of Use:

A simple, understandable, user-friendly interface is provided to the users using the application.

- System Reliability:

In case of a possible error, necessary tests have been carried out and security measures have been taken to quickly correct the detected problems.

8. Conclusion

After completing the PawTime project, a bridge of love will be formed between animals in need and animal-loving people. Thus, animals in need will be identified more easily and more help will be provided

8.1 Benefits

a. Benefits to users :

- Increasing awareness and benefit:

Thanks to the application, awareness about the existence of animals in need of help increases. In this way, the aid to these animals increases and the animals are protected more.

- Increase in rescue potential:

Thanks to the application, information about the location of the animals is provided, helping to facilitate the access of any organization or individual to the animals that need to be rescued and in need of care. Thus, they can help animals and provide resources such as food and water.

- Improving animal welfare:

The app helps to improve the overall welfare of animals.

b. Benefits to me :

You should explain the benefits to you, what will you learn, etc.

- skill development:

I made the project using Java language. I created it on Android Studio. I have connected to Firebase database and used it. Thanks to this project, I gained experience in making mobile applications.

- project management:

While creating the project, I learned and experienced planning and organization.

Why did I choose this project?

One of the reasons I chose this project is my love of animals. My desire to contribute to the world of animals is one of the reasons I created this project. Thanks to this project, it creates more awareness about animals and thus enables people to reach animals in need more easily, provides the opportunity to rescue and care for them.

Another reason why I chose such a project was my interest in mobile applications. I had an environment to realize this and in this way I could have experience in mobile application.

8.2 Future Works

When I continue the project after graduating from school, there are a few issues and issues that I can develop about the project.

There are potential improvements that can be made.

Extending the scope of the project: It may be possible to expand the scope of the project.

partnership with additional organizations or adding new features that allow users to connect animal welfare-related resources or services.

Adding new features:

It may be possible to add new features as project usage increases and evolves. The needs and wishes of the users are determined and in line with these requests and needs,

Features can be added to increase its usefulness.

For example, map features can be developed. The markings on the map and the information it contains can be increased.

Features that allow it to report emergencies can be recognized.

Recognizable features that allow users to track and report their progress over time.

Interface improvements can be made. A user-friendly design can be designed, and features that make it easier for users to access and use the application can be added.

The mobile application project can be moved to the web. Web Development version can be made.

References

How to Analyze Risk in Project Management? (tarih yok).

<https://www.ntaskmanager.com/blog/analyze-risk-in-project-management/> adresinden alındı

Importance of ethics and ethical issues in Project Management: 2023. (tarih yok).

<https://www.henryharvin.com/blog/ethics-to-consider-in-project-management/> adresinden alındı

Project module add-on: "Project Module" Requirements & Screen Designs. (tarih yok).

<https://moodle.org/mod/forum/discuss.php?d=22083> adresinden alındı

Project risk assessment: an example with a risk matrix template. (tarih yok).

<https://bigpicture.one/blog/project-risk-assessment-examples/> adresinden alındı

What's the Code of Conduct for Project Managers? (tarih yok). <https://onlinepmcourses.com/ethics-code-conduct-project-managers/> adresinden alındı

Writing a Literature Review. (tarih yok). <https://writingcenter.uagc.edu/writing-literature-review> adresinden alındı