Truth or Dare App Project for PJPB

Faiz Khansa Adrika
18/425516/PA/18408

Project Description

Truth or Dare App is basically an entertainment application adapting from the “truth or dare” game, where challenges thrown to be done.

Like the real game, the app will enable people to challenge anyone to spill a **truth**, or **dare** an action.

“**Game On!**” section where you could get a random challenge.

Project Concept Succession

Truth or Dare App will adapt the social media concept of ***Instagram*** or closer compared to ***TikTok*** with **global user** target.

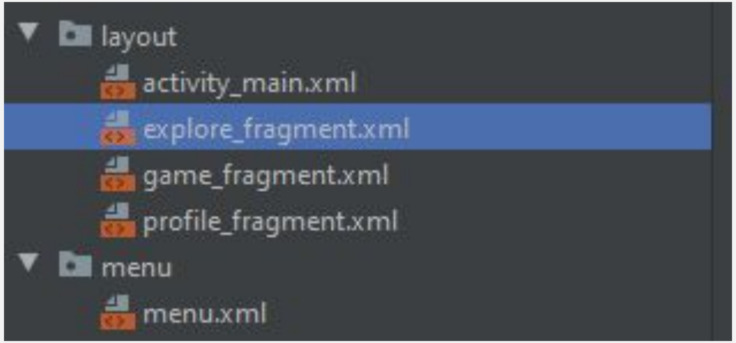
So the special feature of this app is the social media concept, where the challenge will be posted as a photo/video.

But the project haven’t finished yet due to insufficient time, current progress is on **45%** (Layout & Transition only)

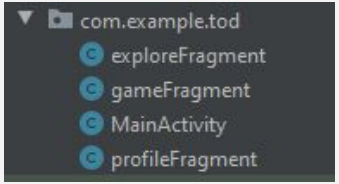
Introduction: Fragments

The app uses fragment method to disect and combine each activity.

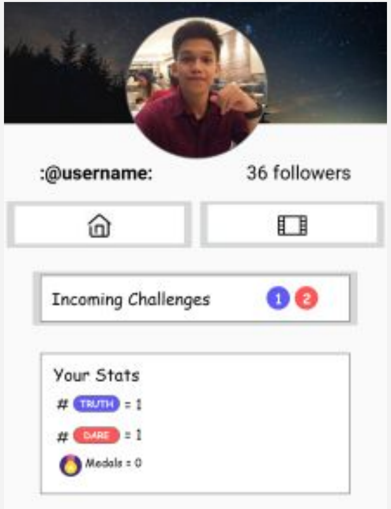
Fragments can be combined as a single activity to build a multi-pane UI and reuse a fragment in multiple activities, so excessive repetitive activities wont exist.

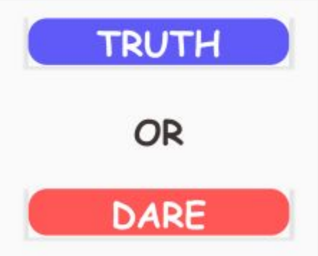


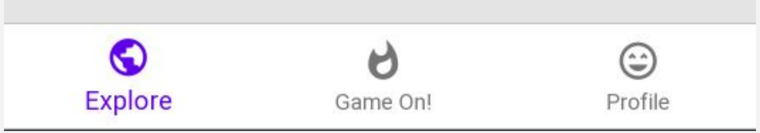
```
▼ layout
    activity_main.xml
    explore_fragment.xml
    game_fragment.xml
    profile_fragment.xml
▼ menu
    menu.xml
```

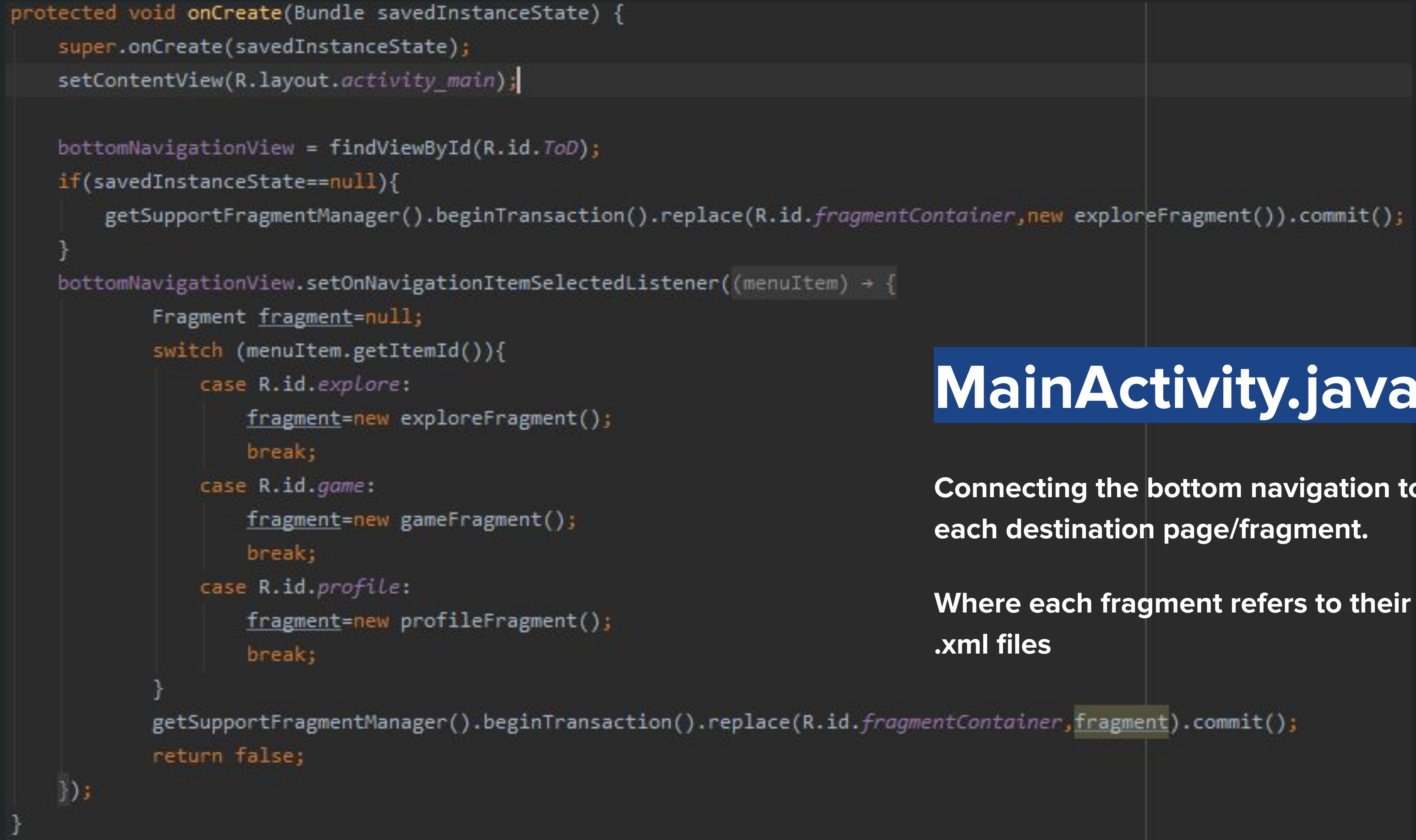


```
▼ com.example.tod
    exploreFragment
    gameFragment
    MainActivity
    profileFragment
```







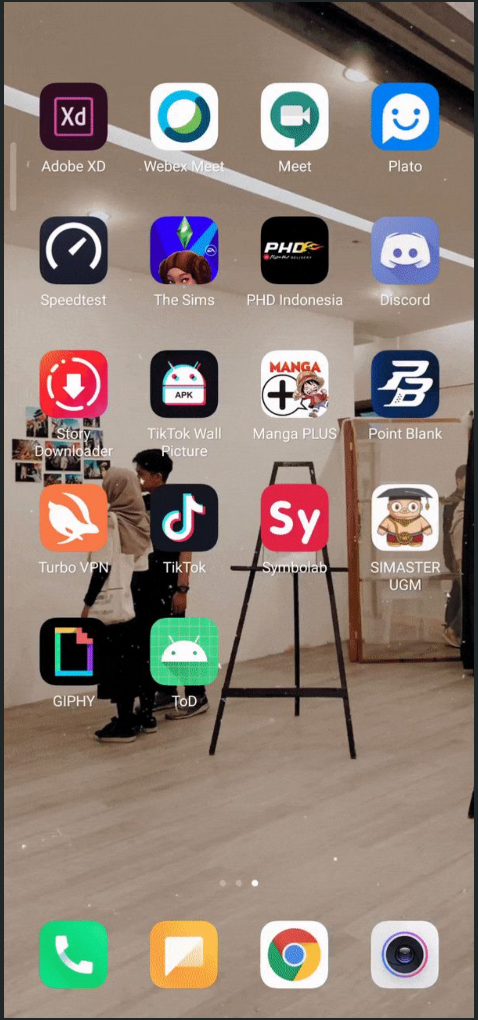


```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    bottomNavigationView = findViewById(R.id.ToD);
    if(savedInstanceState==null){
        getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContainer,new exploreFragment()).commit();
    }
    bottomNavigationView.setOnNavigationItemSelectedListener((menuItem) → {
            Fragment fragment=null;
            switch (menuItem.getItemId()){
                case R.id.explore:
                    fragment=new exploreFragment();
                    break;
                case R.id.game:
                    fragment=new gameFragment();
                    break;
                case R.id.profile:
                    fragment=new profileFragment();
                    break;
            }
            getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContainer,fragment).commit();
            return false;
    });
}
```

MainActivity.java

Connecting the bottom navigation to each destination page/fragment.

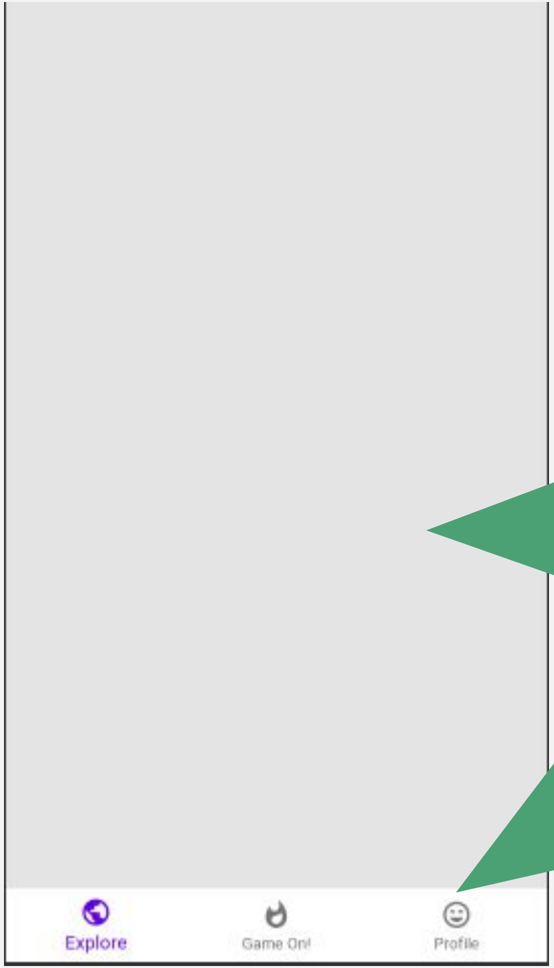
Where each fragment refers to their .xml files



MainActivity.java

Connecting the bottom navigation to each destination page/fragment.

Where each fragment refers to their .xml files



Activity_main.xml

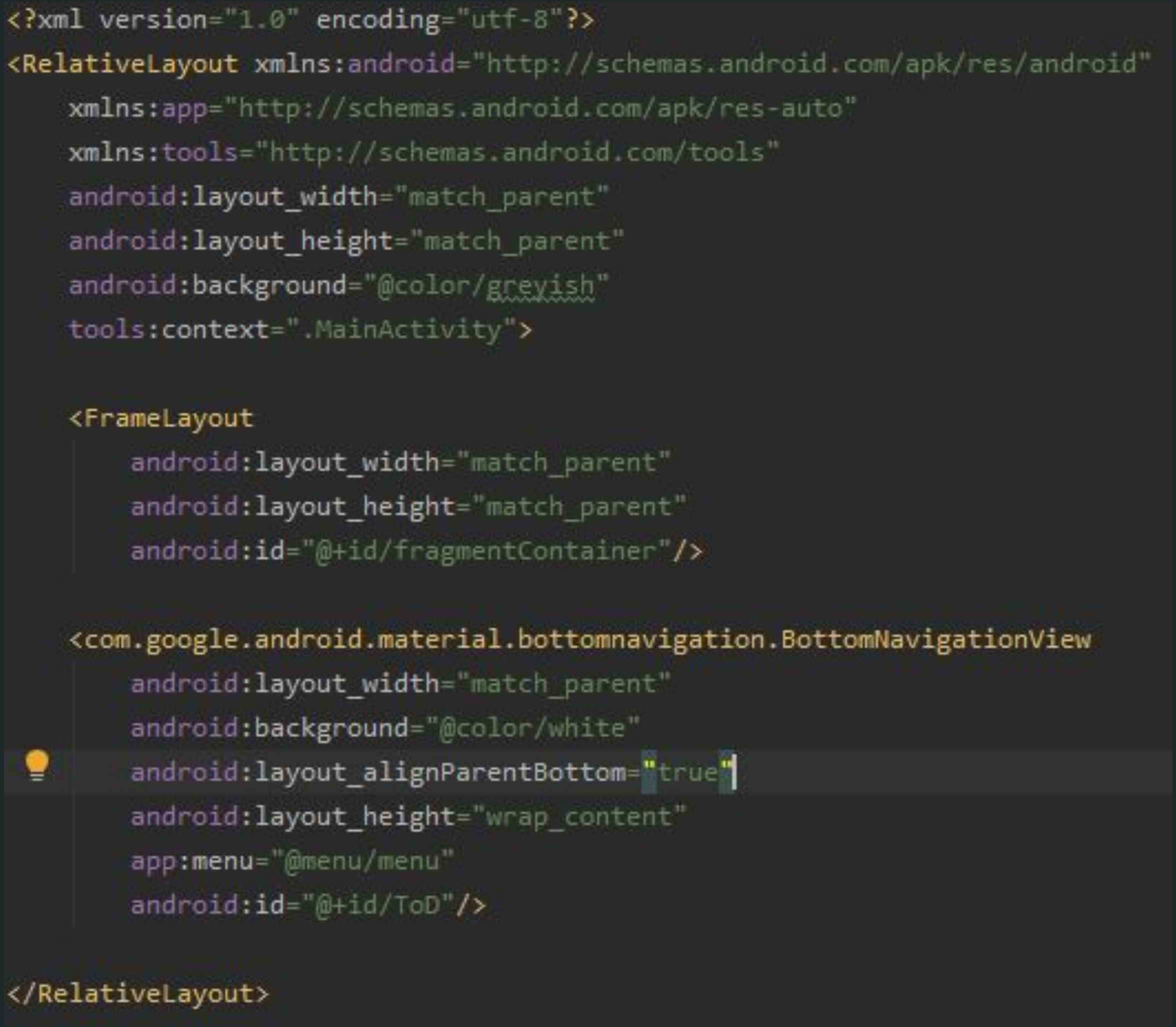
A fragment i created consisting only 2 parts :

1. **Frame Layout** for fragment background.
2. **Bottom Navigation**, for menu navigation

*fully functioning

Activity_main.xml

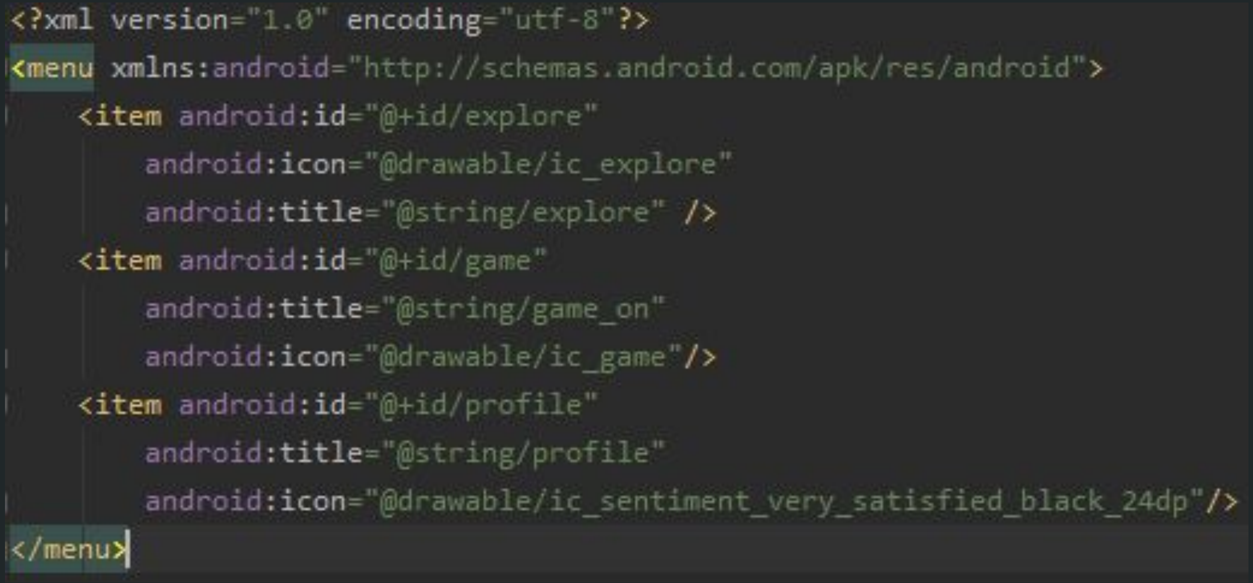
The bottom navigation view is inhabited by menu.xml



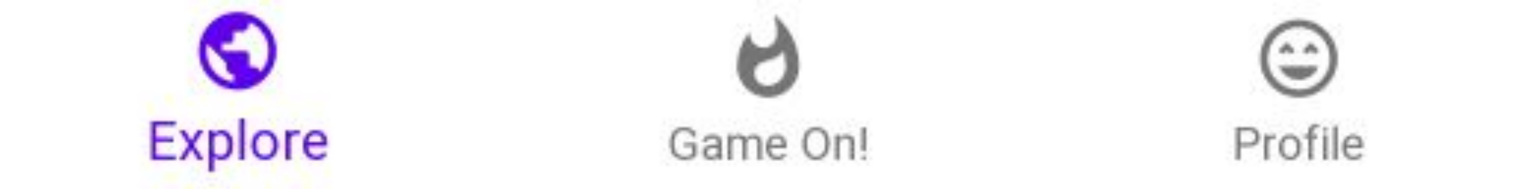
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/greyish"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/fragmentContainer"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:layout_width="match_parent"
        android:background="@color/white"
        android:layout_alignParentBottom="true"
        android:layout_height="wrap_content"
        app:menu="@menu/menu"
        android:id="@+id/ToD"/>

</RelativeLayout>
```

Menu.xml

Storing the attributes (icon & text) for bottom navigation

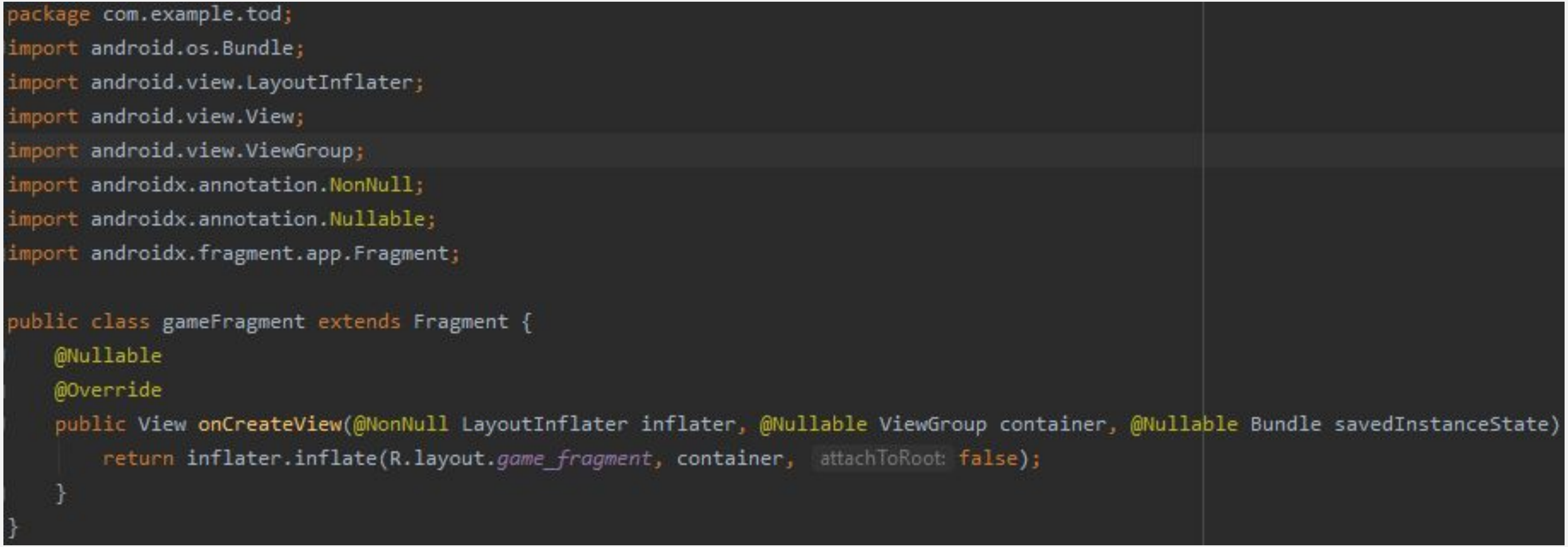


```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/explore"
        android:icon="@drawable/ic_explore"
        android:title="@string/explore" />
    <item android:id="@+id/game"
        android:title="@string/game_on"
        android:icon="@drawable/ic_game"/>
    <item android:id="@+id/profile"
        android:title="@string/profile"
        android:icon="@drawable/ic_sentiment_very_satisfied_black_24dp"/>
</menu>
```



gameFragment.java

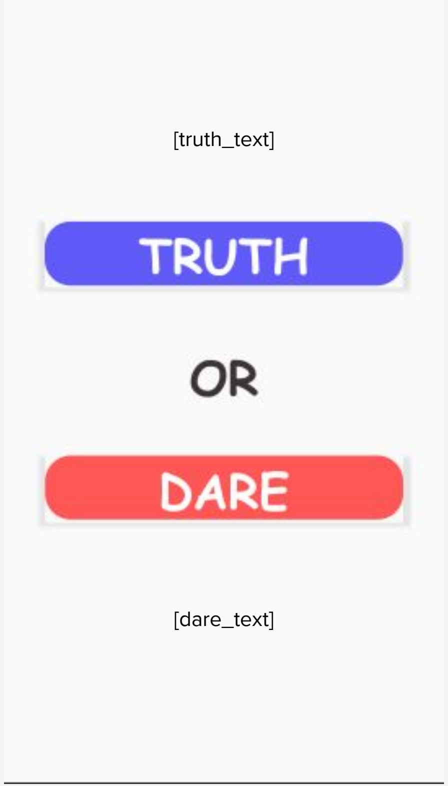
OnCreateView return value to the game_fragment.xml to be shown when selected from MainActivity.java



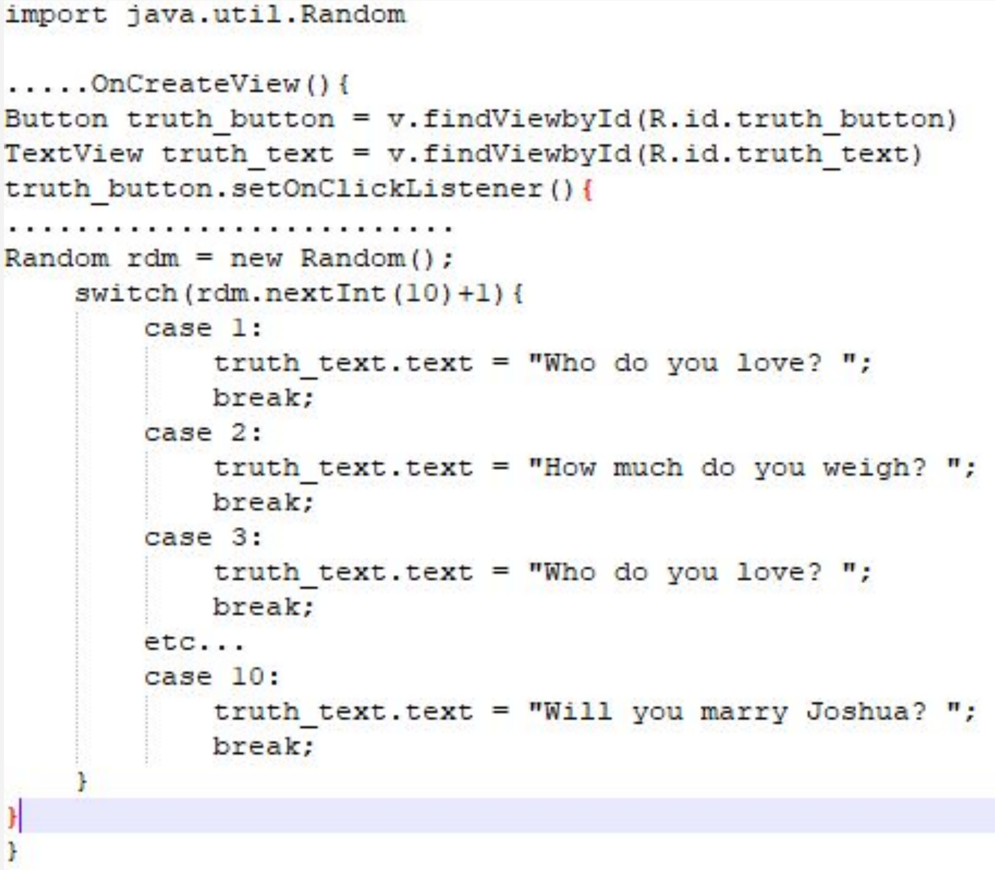
```
package com.example.tod;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class gameFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
        return inflater.inflate(R.layout.game_fragment, container,  attachToRoot: false);
    }
}
```

gameFragment.java

Haven’t been implemented :



Method to show random text :



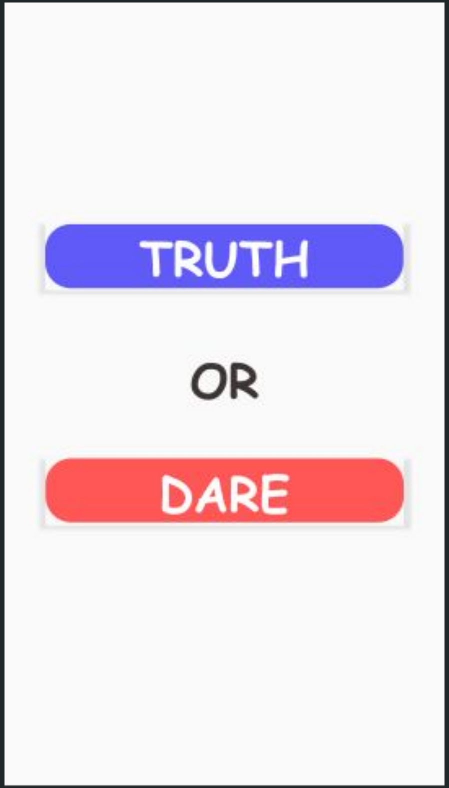
```
import java.util.Random

.....OnCreateView(){
Button truth_button = v.findViewbyId(R.id.truth_button)
TextView truth_text = v.findViewbyId(R.id.truth_text)
truth_button.setOnClickListener(){
..........................
Random rdm = new Random();
    switch(rdm.nextInt(10)+1){
        case 1:
            truth_text.text = "Who do you love? ";
            break;
        case 2:
            truth_text.text = "How much do you weigh? ";
            break;
        case 3:
            truth_text.text = "Who do you love? ";
            break;
        etc...
        case 10:
            truth_text.text = "Will you marry Joshua? ";
            break;
    }
}
}
```

game_fragment.xml

The main game page, consist of :

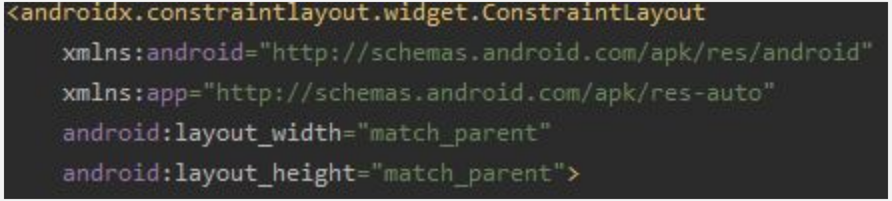
1. **Truth Button Widget** for ‘Truth’ game.
2. **Dare Button Widget** for ‘Dare’ game.
3. **TextView**, the “OR”

*buttons not yet functioning

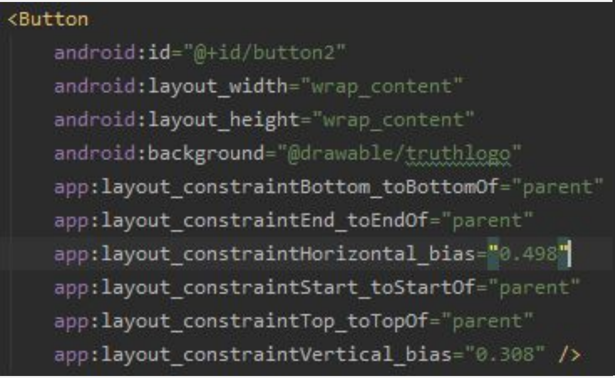


Game_fragment.xml

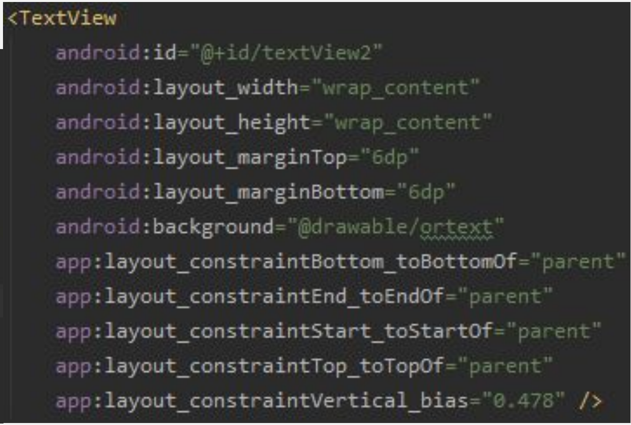
Using Constraint Layout for responsive UI and easier margin placement.



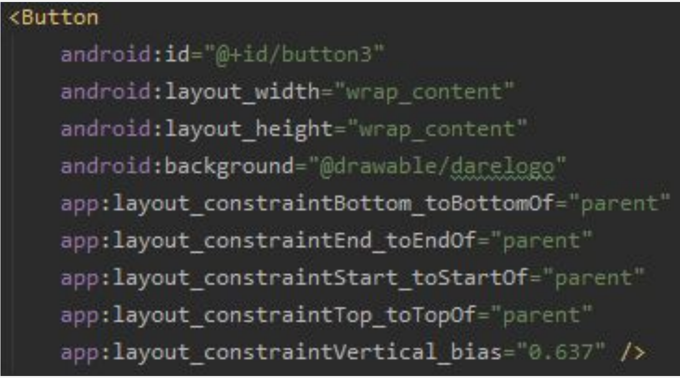
```xml
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```



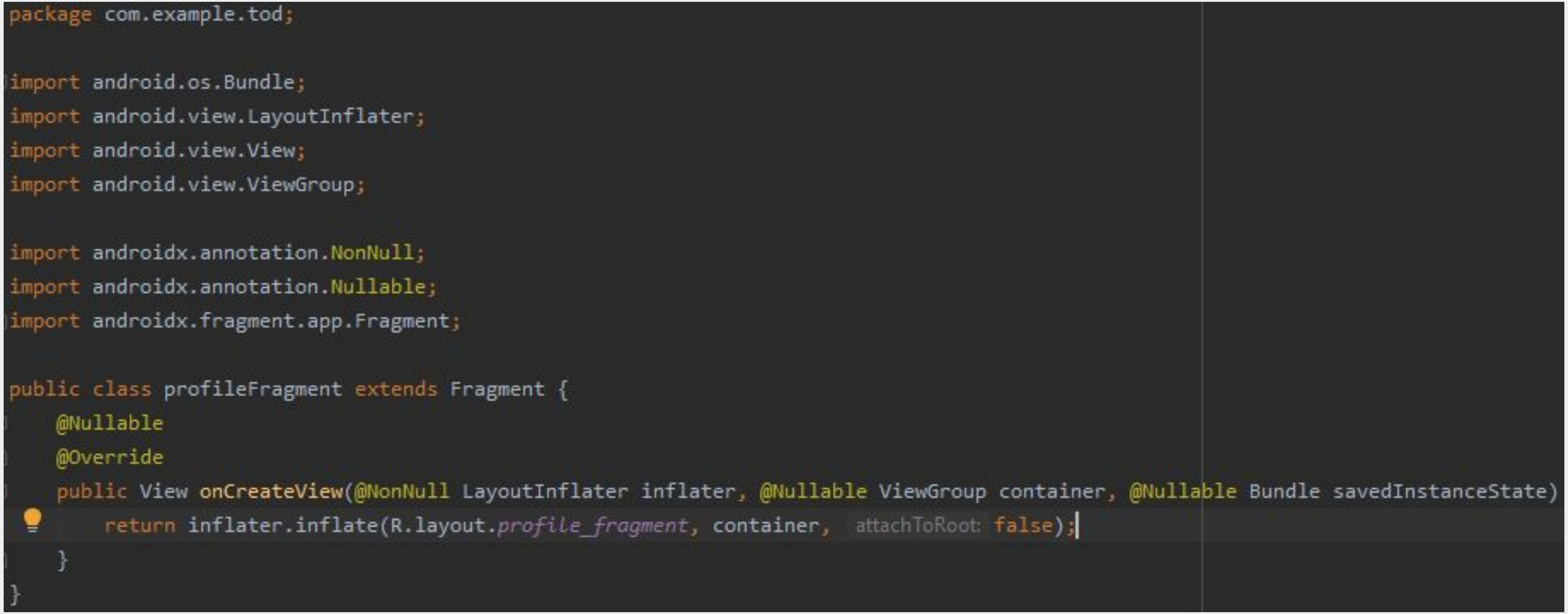
```xml
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/truthlogo"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.308" />
```



```xml
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="6dp"
    android:layout_marginBottom="6dp"
    android:background="@drawable/ortext"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.478" />
```



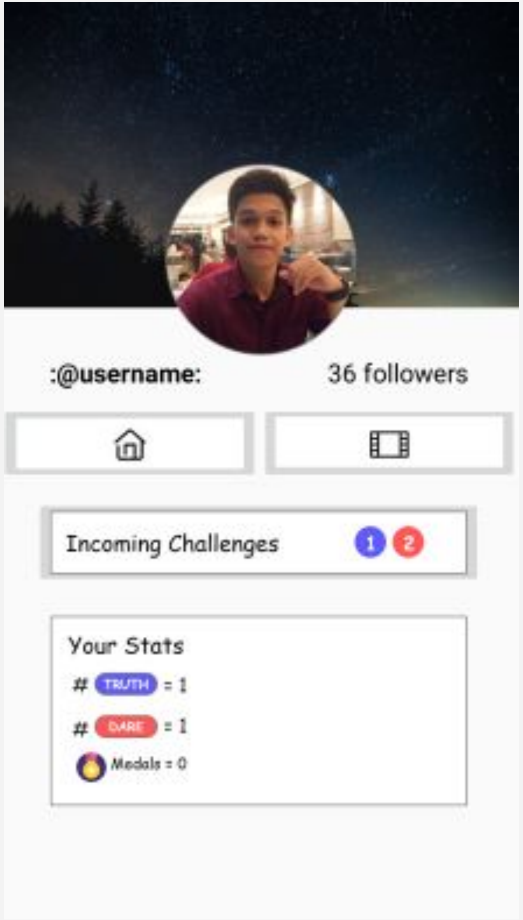
```xml
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/darelogo"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.637" />
```



```java
package com.example.tod;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class profileFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
        return inflater.inflate(R.layout.profile_fragment, container,  attachToRoot: false);
    }
}
```

profileFragment.java

OnCreateView return value to the profile_fragment.xml to be shown when selected from MainActivity.java



profile_fragment.xml

The profile page for personal information, consist of :

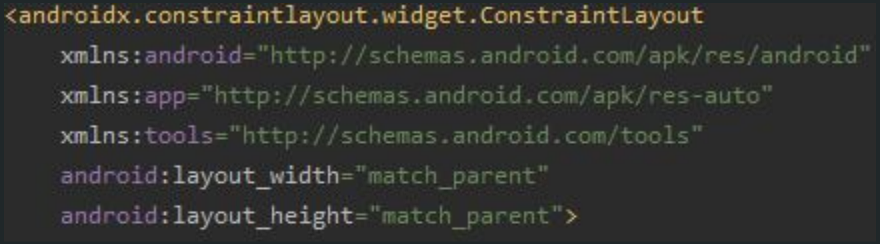
1. **ImageView** for profile, header, and stats.
2. **Image Button**, for menu navigation; e.g. home, video, and challenge button

*buttons not yet functioning

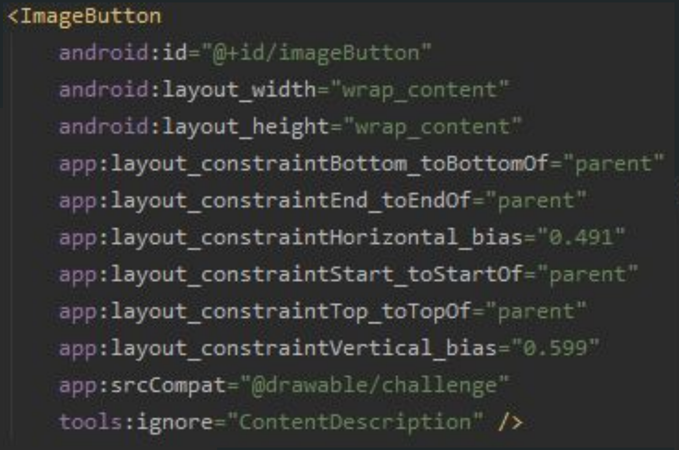
profile_fragment.xml

Using constraintLayout as well. Consist of 2 ImageView, 3 ImageButton, 2 Textview.

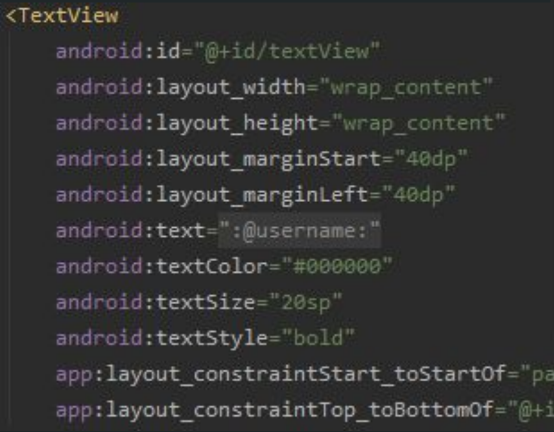
The widget just as representation, not all are attached.



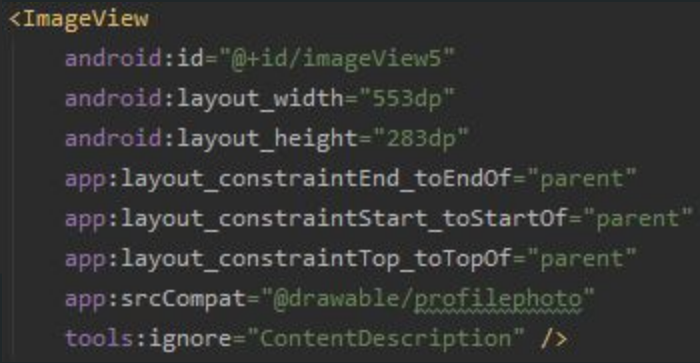
```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```



```
<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.491"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.599"
    app:srcCompat="@drawable/challenge"
    tools:ignore="ContentDescription" />
```



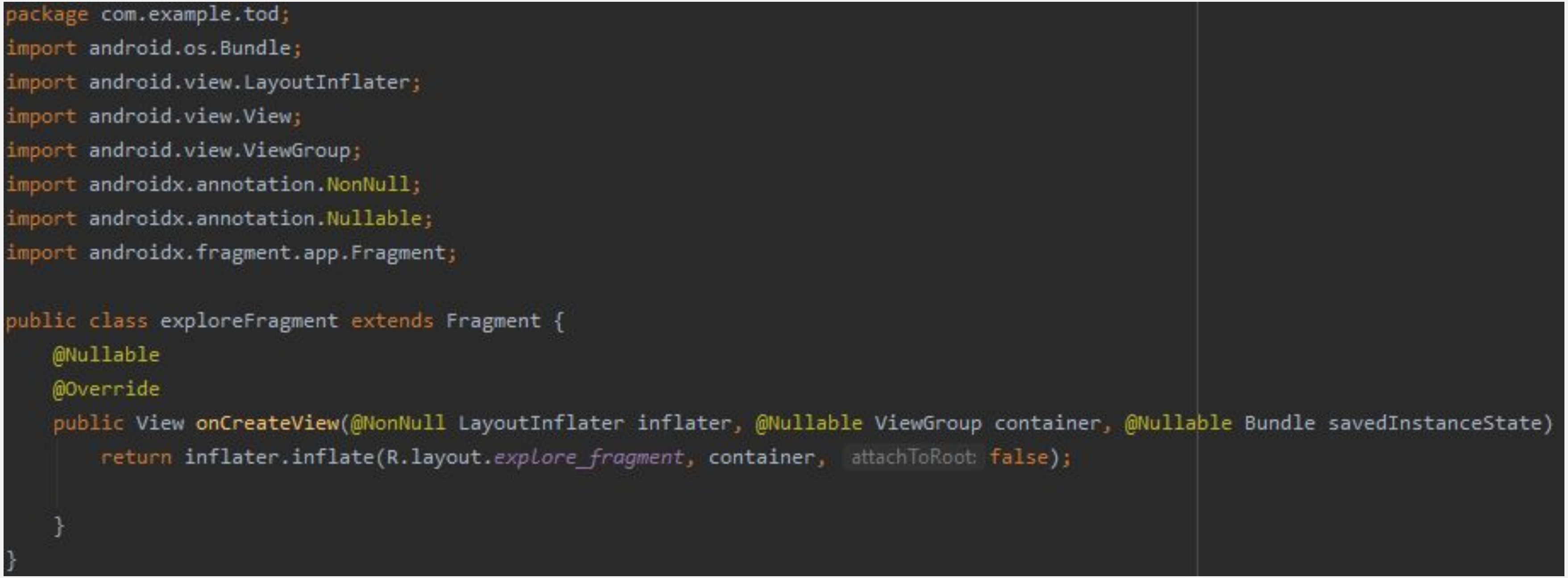
```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginLeft="40dp"
    android:text=":@username:"
    android:textColor="#000000"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView5" />
```



```
<ImageView
    android:id="@+id/imageView5"
    android:layout_width="553dp"
    android:layout_height="283dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/profilephoto"
    tools:ignore="ContentDescription" />
```

exploreFragment.java

OnCreateView return value to the explore_fragment.xml to be shown when selected from MainActivity.java



```
package com.example.tod;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class exploreFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState)
        return inflater.inflate(R.layout.explore_fragment, container,  attachToRoot: false);

    }
}
```

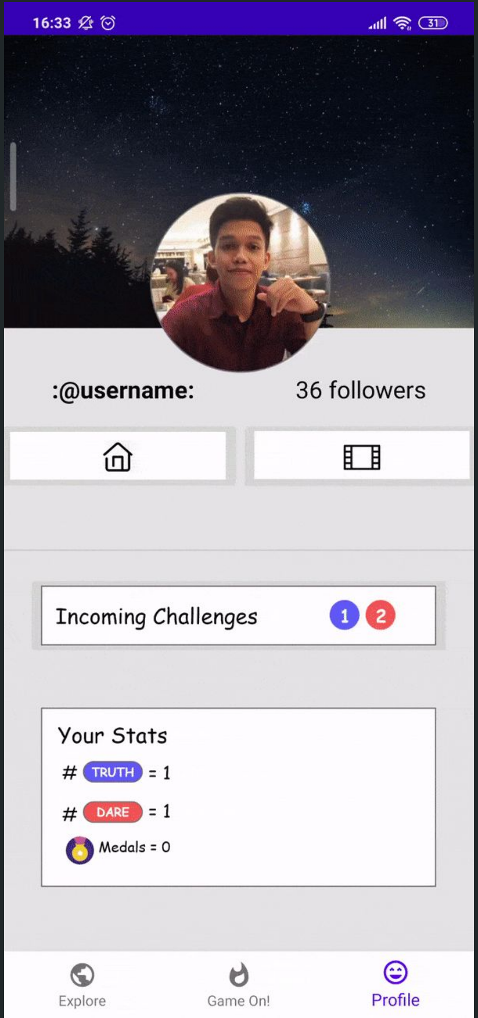
explore_fragment.xml

The explore feed to see scrollable photos page, consist of :

1. **ImageView** for the photos.
2. **TextView** for photo caption.

*fully functioning

*but lack of medal icon



16:33

:@username:

36 followers

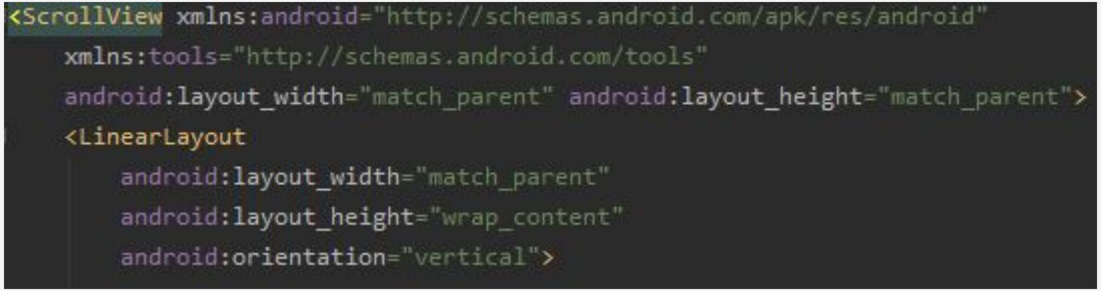
Incoming Challenges 1 2

Your Stats

TRUTH = 1

DARE = 1

Medals = 0

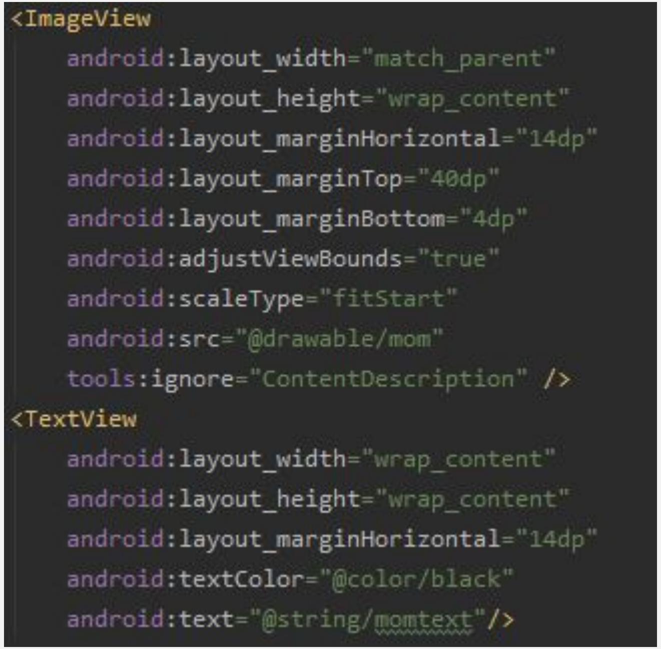


```xml
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
```

explore_fragment.xml

Using ScrollView and LinearLayout so the fragment can be scrolled easily.

For multiple post, add multiple syntax of this ➔ (with different sources, of course)

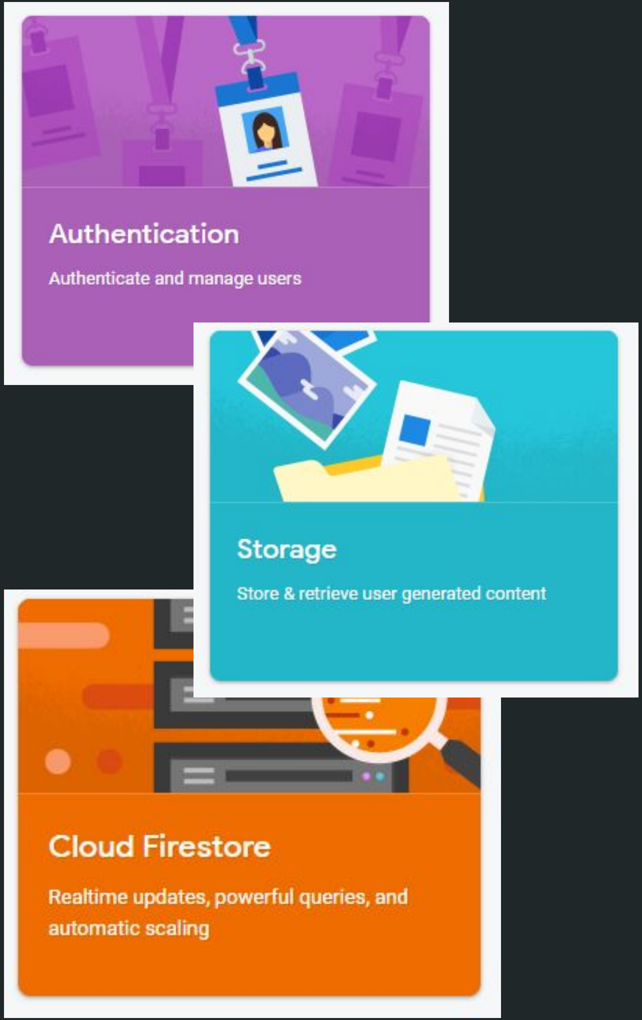


```xml
<ImageView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="14dp"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="4dp"
    android:adjustViewBounds="true"
    android:scaleType="fitStart"
    android:src="@drawable/mom"
    tools:ignore="ContentDescription" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="14dp"
    android:textColor="@color/black"
    android:text="@string/momtext"/>
```

Security and Data Management

The usage of **Firebase** as the project's technical (Data and Security) manager was planned.

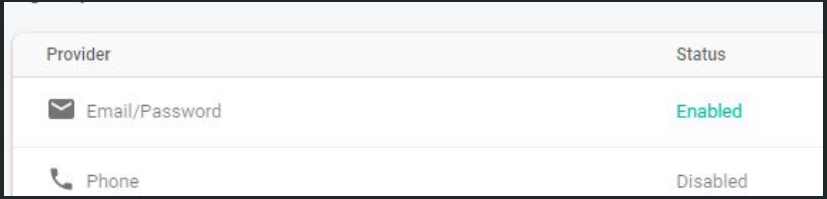
But still it is incomplete because of some issues.


Authentication
Authenticate and manage users
Storage
Store & retrieve user generated content
Cloud Firestore
Realtime updates, powerful queries, and automatic scaling

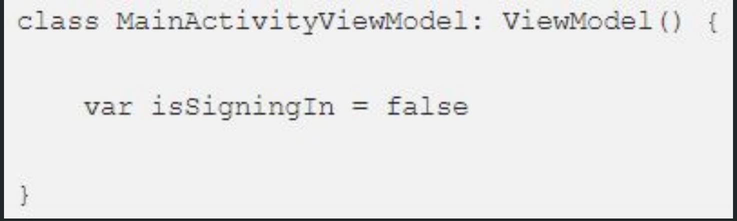
Login Authentication

Planned with **Firebase**’s Authentication. Added the ViewModel and code the syntax and util for email verification.

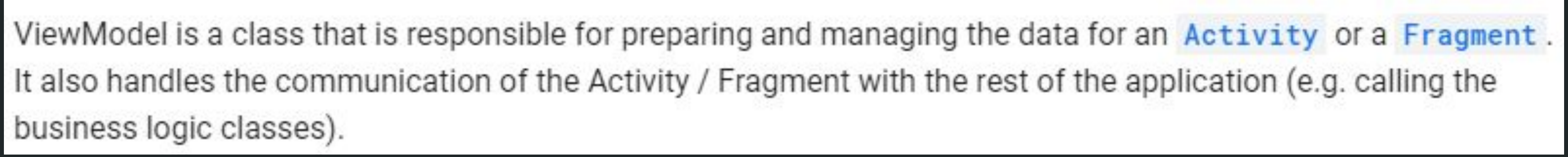
But the program still wont work, and the issue persists. Therefore it cannot be done **just yet**.



Provider	Status
✉ Email/Password	Enabled
☎ Phone	Disabled



```
class MainActivityViewModel: ViewModel() {

    var isSigningIn = false

}
```

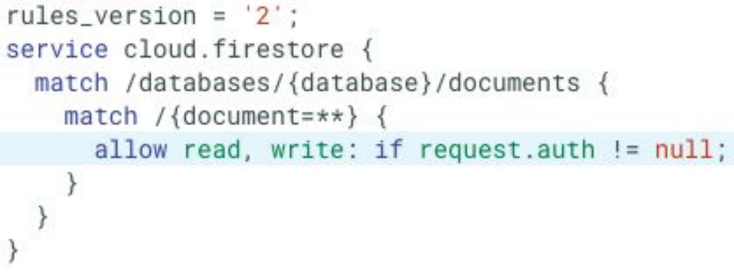


ViewModel is a class that is responsible for preparing and managing the data for an `Activity` or a `Fragment`. It also handles the communication of the Activity / Fragment with the rest of the application (e.g. calling the business logic classes).

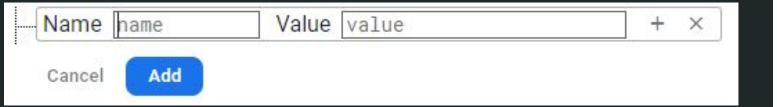
Data Management

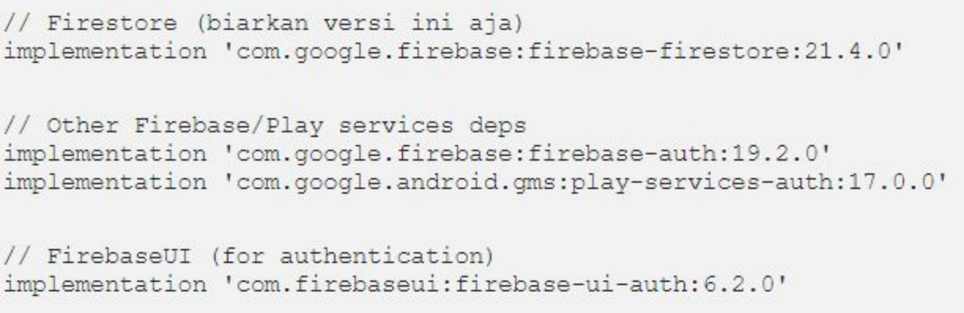
Planned with **Firebase**’s Firestore. Already added util; i also have set the rules for Firestore Database to allow read and write in database

But nothing more, not yet implemented to the android program.

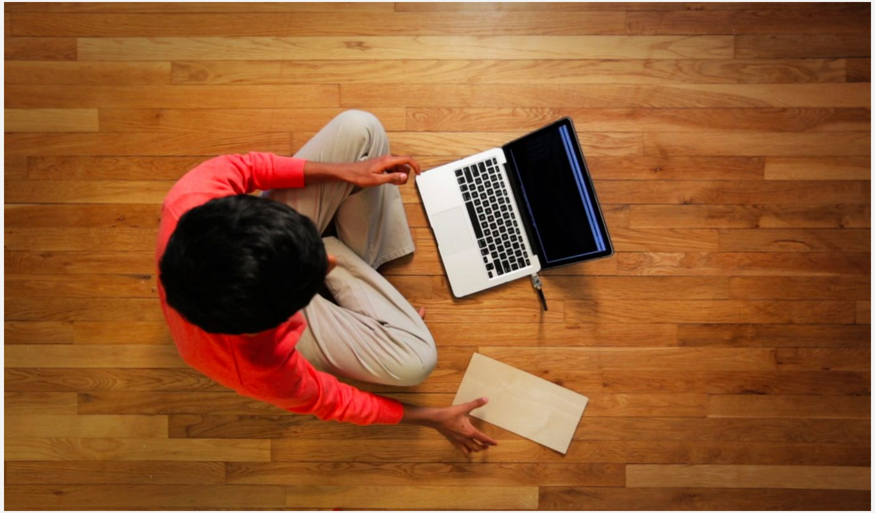


```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```





```
// Firestore (biarkan versi ini aja)
implementation 'com.google.firebase:firebase-firestore:21.4.0'

// Other Firebase/Play services deps
implementation 'com.google.firebase:firebase-auth:19.2.0'
implementation 'com.google.android.gms:play-services-auth:17.0.0'

// FirebaseUI (for authentication)
implementation 'com.firebaseui:firebase-ui-auth:6.2.0'
```



Unfinished work :

❑ Functioning the buttons, add a few activities/fragments

❑ Login authentication remedy

❑ Firestore & database implementation

❑ Enabling Camera API

❑ Connect to the internet

That’s all for the

final progress.

Thank You!

MITT LIV
Uppsala universitet 1852–1916
Det goda universitetet. Festskrift till Anders Hallberg
Studenten, Staden och Sanningen
UPPSALA UNIVERSITE