**Source Code**

* + 1. **activity\_login**

<?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:id="@+id/relative"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/bg\_pattern"

tools:context=".Activities.LoginActivity">

<RelativeLayout

android:id="@+id/relative1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.CardView

android:id="@+id/cv\_one\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="15dp"

android:layout\_marginTop="20dp"

android:elevation="10dp"

app:cardBackgroundColor="#fcfcfc"

app:cardCornerRadius="10dp"

app:cardElevation="10dp"

app:cardPreventCornerOverlap="false"

app:cardUseCompatPadding="true">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignStart="@+id/cv\_one\_login"

android:orientation="vertical"

android:paddingTop="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_gravity="center\_horizontal"

android:layout\_marginBottom="10dp"

android:contentDescription=""

android:src="@drawable/logoutama"

tools:ignore="ContentDescription" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="20sp"

android:fontFamily="@font/fredokaone\_regular"

android:text="-Login-"

android:textAlignment="center"

android:textAllCaps="false"

android:textColor="@color/colorPrimaryDark"

android:textSize="18sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp">

<EditText

android:id="@+id/login\_email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableStart="@mipmap/icon\_email"

android:drawableTint="#FF4081"

android:hint="@string/hint\_email"

android:inputType="textEmailAddress"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginBottom="50dp"

app:passwordToggleEnabled="true"

app:passwordToggleTint="@color/colorAccent">

<EditText

android:id="@+id/login\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:drawableStart="@mipmap/lock"

android:drawableTint="#FF4081"

android:hint="@string/hint\_pass"

android:inputType="textPassword"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

<ImageButton

android:id="@+id/btn\_login"

android:layout\_width="55dp"

android:layout\_height="55dp"

android:layout\_below="@id/relative1"

android:layout\_centerHorizontal="true"

android:layout\_gravity="center"

android:layout\_marginTop="-43dp"

android:background="@mipmap/check\_circle"

android:elevation="15dp" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/relative1"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="20dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp"

android:fontFamily="@font/fredokaone\_regular"

android:text="OR"

android:textColor="@android:color/white"

android:textStyle="bold" />

<com.google.android.gms.common.SignInButton

android:id="@+id/sign\_in\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_gravity="center\_horizontal" />

<Button

android:id="@+id/btn\_signup"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_margin="20dp"

android:background="@drawable/background\_rounded1"

android:text="Don't Have Acc? Register!"

android:textAllCaps="false"

android:textColor="@android:color/white" />

</RelativeLayout>

<ProgressBar

android:id="@+id/progressBar"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="0dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="0dp"

android:layout\_marginBottom="0dp"

android:indeterminate="true"

android:visibility="invisible"

tools:ignore="MissingConstraints" />

</RelativeLayout>

* + 1. **activity\_register**

<?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="@drawable/bg\_pattern"

tools:context=".Activities.LoginActivity">

<RelativeLayout

android:id="@+id/relative1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.CardView

android:id="@+id/cardv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp"

android:layout\_marginStart="15dp"

android:elevation="10dp"

app:cardBackgroundColor="#fcfcfc"

app:cardCornerRadius="10dp"

app:cardElevation="10dp"

app:cardPreventCornerOverlap="false"

app:cardUseCompatPadding="true">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignStart="@+id/cardv"

android:orientation="vertical"

android:paddingTop="10dp">

<!--<ImageView-->

<!--android:layout\_width="100dp"-->

<!--android:layout\_height="100dp"-->

<!--android:layout\_gravity="center\_horizontal"-->

<!--android:layout\_marginBottom="10dp"-->

<!--android:contentDescription=""-->

<!--android:src="@drawable/logo"-->

<!--tools:ignore="ContentDescription" />-->

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="20sp"

android:fontFamily="@font/fredokaone\_regular"

android:text="-Register-"

android:textAlignment="center"

android:textAllCaps="false"

android:textColor="@color/colorPrimaryDark"

android:textSize="18sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp">

<EditText

android:id="@+id/reg\_email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableStart="@mipmap/icon\_email"

android:drawableTint="#FF4081"

android:hint="@string/hint\_email"

android:inputType="textEmailAddress"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

app:passwordToggleEnabled="true"

app:passwordToggleTint="@color/colorAccent">

<EditText

android:id="@+id/reg\_pass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:drawableStart="@mipmap/lock"

android:drawableTint="#FF4081"

android:hint="@string/hint\_pass"

android:inputType="textPassword"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

app:passwordToggleEnabled="true"

app:passwordToggleTint="@color/colorAccent">

<EditText

android:id="@+id/reg\_confirm\_pass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:drawableStart="@mipmap/lock"

android:drawableTint="#FF4081"

android:hint="Confirm Password"

android:inputType="textPassword"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<CheckBox

android:id="@+id/term"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_marginBottom="50dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="5dp"

android:text="Agree to Our Term and Conditions"/>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

<ImageButton

android:id="@+id/btn\_reg"

android:layout\_width="55dp"

android:layout\_height="55dp"

android:layout\_below="@id/relative1"

android:layout\_centerHorizontal="true"

android:layout\_gravity="center"

android:layout\_marginTop="-43dp"

android:background="@mipmap/check\_circle"

android:elevation="15dp" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/relative1"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="20dp">

<Button

android:id="@+id/back\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:text="Already Have Acc? Login!"

android:textAllCaps="false"

android:background="@drawable/background\_rounded1"

android:textColor="@android:color/white"/>

</RelativeLayout>

<ProgressBar

android:id="@+id/progressBar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="0dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="0dp"

android:layout\_marginBottom="0dp"

android:indeterminate="true"

android:visibility="invisible"

tools:ignore="MissingConstraints"

style="?android:attr/progressBarStyleHorizontal"/>

</RelativeLayout>

* + 1. **activity\_verification\_email**

<?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"

tools:context=".Activities.VerificationEmailActivity">

<RelativeLayout

android:id="@+id/r1"

android:layout\_width="match\_parent"

android:layout\_height="250dp"

android:background="@color/colorPrimaryDark">

<TextView

android:id="@+id/please"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="50dp"

android:fontFamily="@font/comfortaa"

android:gravity="center"

android:text="Please"

android:textColor="@color/putih"

android:textSize="30dp"

android:textStyle="bold" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/please"

android:layout\_centerHorizontal="true"

android:fontFamily="@font/comfortaa"

android:gravity="center"

android:text="Activate Your Email"

android:textColor="@color/putih"

android:textSize="30dp"

android:textStyle="bold" />

</RelativeLayout>

<ImageView

android:id="@+id/icon\_email"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:layout\_below="@+id/r1"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="-75dp"

android:src="@drawable/ic\_email1" />

<TextView

android:id="@+id/tv\_email"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/icon\_email"

android:layout\_centerHorizontal="true"

android:layout\_margin="30dp"

android:fontFamily="@font/comfortaa"

android:gravity="center"

android:text="We've sent a link. Please check your email account and click the link inside"

android:textColor="#000000"

android:textSize="17dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/linear\_veri"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="5dp"

android:text='\*after activate email,click "Next" button below.' />

<LinearLayout

android:id="@+id/linear\_veri"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:orientation="horizontal"

android:weightSum="3">

<Button

android:id="@+id/ext"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="30dp"

android:layout\_weight="1"

android:background="@drawable/background\_rounded1"

android:fontFamily="@font/comfortaa"

android:padding="5dp"

android:text="Resend Link"

android:textAllCaps="false"

android:textColor="@android:color/white" />

<Button

android:id="@+id/edit\_email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="30dp"

android:layout\_weight="1"

android:background="@drawable/background\_rounded1"

android:fontFamily="@font/comfortaa"

android:padding="5dp"

android:text="Edit Email"

android:textAllCaps="false"

android:textColor="@android:color/white" />

<Button

android:id="@+id/btn\_next"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="30dp"

android:layout\_weight="1"

android:background="@drawable/background\_rounded1"

android:fontFamily="@font/comfortaa"

android:padding="5dp"

android:text="Next!"

android:textAllCaps="false"

android:textColor="@android:color/white" />

</LinearLayout>

</RelativeLayout>

* + 1. **activity\_re\_auth\_email**

<?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="@drawable/bg\_login1"

tools:context=".Activities.ReAuthEmailActivity">

<RelativeLayout

android:id="@+id/relative3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true">

<android.support.v7.widget.CardView

android:id="@+id/cardv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="15dp"

android:layout\_marginTop="20dp"

android:elevation="10dp"

app:cardBackgroundColor="#fcfcfc"

app:cardCornerRadius="10dp"

app:cardElevation="10dp"

app:cardPreventCornerOverlap="false"

app:cardUseCompatPadding="true">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignStart="@+id/cardv"

android:layout\_marginBottom="50dp"

android:orientation="vertical"

android:paddingTop="10dp">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="20sp"

android:fontFamily="@font/fredokaone\_regular"

android:text="-Edit Email-"

android:textAlignment="center"

android:textAllCaps="false"

android:textColor="@color/colorPrimaryDark"

android:textSize="18sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp">

<EditText

android:id="@+id/edit\_emailveri"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableStart="@mipmap/icon\_email"

android:drawableTint="#FF4081"

android:hint="@string/hint\_email"

android:inputType="textEmailAddress"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

<ImageButton

android:id="@+id/btn\_edit\_email"

android:layout\_width="55dp"

android:layout\_height="55dp"

android:layout\_below="@id/relative3"

android:layout\_centerHorizontal="true"

android:layout\_gravity="center"

android:layout\_marginTop="-43dp"

android:background="@mipmap/check\_circle"

android:elevation="15dp" />

</RelativeLayout>

* + 1. **activity\_setup**

<?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="@drawable/bg\_pattern"

tools:context=".Activities.LoginActivity">

<RelativeLayout

android:id="@+id/relative1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.CardView

android:id="@+id/cv\_one\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="15dp"

android:layout\_marginTop="10dp"

android:elevation="10dp"

app:cardBackgroundColor="#fcfcfc"

app:cardCornerRadius="10dp"

app:cardElevation="10dp"

app:cardPreventCornerOverlap="false"

app:cardUseCompatPadding="true">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignStart="@+id/cv\_one\_login"

android:orientation="vertical"

android:paddingTop="20dp">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="30sp"

android:fontFamily="@font/fredokaone\_regular"

android:text="-Setup Account-"

android:textAlignment="center"

android:textAllCaps="false"

android:textColor="@color/colorPrimaryDark"

android:textSize="20sp" />

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/setup\_image"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:layout\_gravity="center\_horizontal"

android:layout\_marginBottom="30dp"

android:contentDescription=""

android:src="@drawable/defaultimage"

tools:ignore="ContentDescription" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:visibility="gone">

<EditText

android:id="@+id/setup\_email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableLeft="@mipmap/icon\_email"

android:drawableTint="#FF4081"

android:editable="false"

android:hint="Your Email.."

android:singleLine="true"

android:visibility="gone" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

app:counterEnabled="true"

app:counterMaxLength="15">

<EditText

android:id="@+id/setup\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableLeft="@mipmap/account"

android:drawableTint="#FF4081"

android:hint="Username..."

android:inputType="textCapWords"

android:maxLength="15"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginBottom="50dp"

app:counterEnabled="true"

app:counterMaxLength="13">

<EditText

android:id="@+id/setup\_phone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:drawableLeft="@mipmap/icon\_phone"

android:drawableTint="#FF4081"

android:hint="Phone Number.."

android:inputType="phone"

android:maxLength="13"

android:singleLine="true" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

</RelativeLayout>

<ImageButton

android:id="@+id/btn\_setup"

android:layout\_width="65dp"

android:layout\_height="65dp"

android:layout\_below="@id/relative1"

android:layout\_centerHorizontal="true"

android:layout\_gravity="center"

android:layout\_marginTop="-49dp"

android:background="@mipmap/check\_circle"

android:elevation="15dp" />

</RelativeLayout>

* + 1. **activity\_main**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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"

tools:context=".Activities.MainActivity">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="0dp"

android:background="@color/colorPrimary"

android:theme="@style/ToolbarTheme"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:titleTextColor="@android:color/white">

</android.support.v7.widget.Toolbar>

<FrameLayout

android:id="@+id/fragment\_container"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_margin="0dp"

app:layout\_constraintBottom\_toTopOf="@+id/button\_navigation"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/toolbar">

</FrameLayout>

<android.support.design.widget.BottomNavigationView

android:id="@+id/button\_navigation"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:layout\_marginStart="0dp"

android:layout\_marginEnd="0dp"

android:background="@android:color/white"

app:itemIconTint="@color/colorPrimary"

app:itemTextColor="@color/colorPrimary"

app:labelVisibilityMode="selected"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toLeftOf="parent"

app:menu="@menu/navigation">

</android.support.design.widget.BottomNavigationView>

</android.support.constraint.ConstraintLayout>

* + 1. **activity\_new\_post**

<?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"

tools:context=".Activities.NewPostActivity">

<android.support.v7.widget.Toolbar

android:id="@+id/comment\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorPrimary"

android:minHeight="?attr/actionBarSize"

app:titleTextColor="@android:color/white" />

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@+id/comment\_toolbar"

android:layout\_alignParentStart="true"

android:layout\_marginStart="0dp"

android:layout\_marginTop="0dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageButton

android:id="@+id/post\_image"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:adjustViewBounds="true"

android:background="@android:color/white"

android:scaleType="centerCrop"

android:src="@mipmap/add\_btn">

</ImageButton>

<EditText

android:id="@+id/post\_lati"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/post\_image"

android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

android:background="@drawable/inp\_outline"

android:hint="Latitude"

android:inputType="text"

android:padding="15dp"

android:visibility="gone" />

<EditText

android:id="@+id/post\_longi"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/post\_lati"

android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

android:background="@drawable/inp\_outline"

android:hint="Longitude"

android:inputType="text"

android:padding="15dp"

android:visibility="gone" />

<android.support.design.widget.TextInputLayout

android:id="@+id/txt\_address"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/post\_image"

android:layout\_margin="20dp">

<EditText

android:id="@+id/post\_address"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:enabled="false"

android:hint="Address"

android:maxLength="140"

android:padding="20dp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/txt\_address"

android:layout\_marginStart="20dp"

android:layout\_marginTop="0dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

app:counterEnabled="true"

app:counterMaxLength="140">

<EditText

android:id="@+id/post\_desc"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_post\_desc"

android:maxLength="140"

android:padding="20dp" />

</android.support.design.widget.TextInputLayout>

</RelativeLayout>

</ScrollView>

</RelativeLayout>

* + 1. **fragment\_home**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

tools:context=".Fragment.HomeFragment">

<android.support.v4.widget.SwipeRefreshLayout

android:id="@+id/pull"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.v7.widget.RecyclerView

android:id="@+id/blog\_list\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</android.support.v4.widget.SwipeRefreshLayout>

</FrameLayout>

* + 1. **blog\_row**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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="wrap\_content">

<android.support.v7.widget.CardView

android:id="@+id/cardview\_home"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="0dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<com.mikhaellopez.circularimageview.CircularImageView

android:id="@+id/blog\_user\_image"

android:layout\_width="65dp"

android:layout\_height="65dp"

android:layout\_marginStart="6dp"

android:layout\_marginTop="6dp"

android:src="@drawable/profile\_placeholder"

app:civ\_border\_color="#EEEEEE"

app:civ\_border\_width="1dp"

app:civ\_shadow="true"

app:civ\_shadow\_color="@color/dasarpink"

app:civ\_shadow\_radius="5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/blog\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:text="..."

android:textSize="18sp"

app:fontFamily="@font/breeserif\_regular"

app:layout\_constraintStart\_toEndOf="@+id/blog\_user\_image"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/blog\_address"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="4dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:ellipsize="end"

android:fontFamily="@font/roboto\_regular"

android:maxLines="2"

android:text=""

android:textColor="@android:color/darker\_gray"

android:textSize="12sp"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_image"

app:layout\_constraintEnd\_toStartOf="@+id/popup\_home"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toEndOf="@+id/blog\_user\_image"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_username"

app:layout\_constraintVertical\_bias="0.0" />

<TextView

android:id="@+id/blog\_date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="16dp"

android:layout\_marginBottom="8dp"

android:fontFamily="@font/roboto\_regular"

android:text="Post Date"

android:textColor="@android:color/darker\_gray"

android:textSize="12sp"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_desc"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_image" />

<ImageView

android:id="@+id/blog\_image"

android:layout\_width="0dp"

android:layout\_height="250dp"

android:layout\_marginTop="6dp"

android:scaleType="centerCrop"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_user\_image"

app:srcCompat="@drawable/image\_placeholder" />

<TextView

android:id="@+id/blog\_desc"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="16dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="16dp"

android:layout\_marginBottom="16dp"

android:text="..."

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/btn\_like"

app:layout\_constraintVertical\_bias="1.0" />

<ImageView

android:id="@+id/btn\_like"

android:layout\_width="28dp"

android:layout\_height="28dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_image"

app:srcCompat="@mipmap/like\_gray" />

<TextView

android:id="@+id/like\_count"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

android:text="0"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_desc"

app:layout\_constraintStart\_toEndOf="@+id/btn\_like"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_image" />

<ImageView

android:id="@+id/btn\_comment"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_desc"

app:layout\_constraintStart\_toEndOf="@+id/like\_count"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_image"

app:srcCompat="@mipmap/comment" />

<TextView

android:id="@+id/comment\_count"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

android:text="0"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_desc"

app:layout\_constraintStart\_toEndOf="@+id/btn\_comment"

app:layout\_constraintTop\_toBottomOf="@+id/blog\_image" />

<ImageView

android:id="@+id/popup\_home"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="16dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/baseline\_more\_vert\_black\_24"

app:layout\_constraintBottom\_toTopOf="@+id/blog\_image"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

<LinearLayout

android:id="@+id/hideitem"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/putih"

android:gravity="center\_vertical|center\_horizontal"

android:orientation="vertical"

android:visibility="invisible">

<ImageView

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_marginTop="-50dp"

android:background="@color/colorAmber\_A400"

android:src="@android:drawable/stat\_sys\_warning" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="20dp"

android:layout\_marginEnd="20dp"

android:background="@android:color/white"

android:fontFamily="@font/comfortaa"

android:gravity="center\_vertical|center\_horizontal"

android:text="Thanks for Reported this post."

android:textColor="@color/blue\_500"

android:textSize="30dp"

android:visibility="visible" />

</LinearLayout>

</android.support.v7.widget.CardView>

</android.support.constraint.ConstraintLayout>

* + 1. **activity\_detail**

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout 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="@android:color/white"

android:fitsSystemWindows="true"

tools:context=".Activities.DetailActivity">

<android.support.design.widget.AppBarLayout

android:id="@+id/appbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/AppTheme.AppBarOverlay">

<android.support.design.widget.CollapsingToolbarLayout

android:id="@+id/collapsingToolbar"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:elevation="20dp"

app:collapsedTitleGravity="start"

app:contentScrim="?attr/colorPrimary"

app:expandedTitleMarginBottom="16dp"

app:expandedTitleMarginEnd="64dp"

app:expandedTitleMarginStart="24dp"

app:layout\_scrollFlags="exitUntilCollapsed|scroll">

<com.flaviofaria.kenburnsview.KenBurnsView

android:id="@+id/detail\_image"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

android:background="@color/grey"

android:maxHeight="350dp"

android:scaleType="fitXY"

android:src="@drawable/rectangle"

app:layout\_collapseMode="parallax"

app:layout\_collapseParallaxMultiplier="0.7" />

<android.support.v7.widget.Toolbar

android:id="@+id/detail\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

app:layout\_collapseMode="pin"

app:popupTheme="@style/AppTheme.PopupOverlay"></android.support.v7.widget.Toolbar>

</android.support.design.widget.CollapsingToolbarLayout>

<RelativeLayout

android:id="@+id/linearLayout2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@android:color/white"

android:paddingStart="10dp"

android:paddingTop="15dp"

android:paddingEnd="10dp"

android:paddingBottom="15dp"

android:showDividers="middle"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/thumb\_userpost"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignParentTop="true"

android:layout\_marginStart="5dp"

android:layout\_marginTop="5dp"

app:srcCompat="@drawable/profile\_placeholder" />

<TextView

android:id="@+id/userpost"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="5dp"

android:layout\_marginBottom="2dp"

android:layout\_toRightOf="@+id/thumb\_userpost"

android:fontFamily="@font/breeserif\_regular"

android:text="..."

android:textColor="@android:color/black" />

<TextView

android:id="@+id/descpost"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/userpost"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/thumb\_userpost"

android:hint="..."

android:textColor="@android:color/black" />

<TextView

android:id="@+id/time\_post"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_alignParentRight="true"

android:layout\_marginTop="5dp"

android:fontFamily="@font/roboto\_regular"

android:text="..ago"

android:textColor="@android:color/darker\_gray"

android:textSize="12sp" />

</RelativeLayout>

</android.support.design.widget.AppBarLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/list\_comment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:background="@android:color/white"

android:padding="5dp">

<EditText

android:id="@+id/isi\_komen"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_toLeftOf="@+id/kirim\_komen"

android:background="@null"

android:hint="Comment here...."

android:inputType="textImeMultiLine"

android:padding="10dp">

<requestFocus />

</EditText>

<ImageView

android:id="@+id/kirim\_komen"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/isi\_komen"

android:layout\_alignBottom="@+id/isi\_komen"

android:layout\_alignParentRight="true"

android:layout\_marginTop="0dp"

android:layout\_marginRight="1dp"

android:layout\_marginBottom="0dp"

android:paddingRight="10dp"

app:srcCompat="@drawable/baseline\_send\_24" />

</RelativeLayout>

</android.support.design.widget.CoordinatorLayout>

* + 1. **comment\_list\_item**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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="wrap\_content"

tools:layout\_editor\_absoluteY="25dp">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/comment\_image"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_marginStart="15dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

android:src="@drawable/profile\_placeholder"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/comment\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:fontFamily="@font/breeserif\_regular"

android:hint="..."

android:textColor="@android:color/black"

app:layout\_constraintStart\_toEndOf="@+id/comment\_image"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/comment\_message"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:text="Loading Comment..."

android:textAllCaps="false"

android:textColor="#5F5959"

app:layout\_constraintStart\_toEndOf="@+id/comment\_image"

app:layout\_constraintTop\_toBottomOf="@+id/comment\_username" />

</android.support.constraint.ConstraintLayout>

* + 1. **popup\_detail\_profile**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="350dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:background="@color/colorPrimaryDark"

android:orientation="vertical"

android:padding="5dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<com.mikhaellopez.circularimageview.CircularImageView

android:id="@+id/popup\_user\_image"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_gravity="center"

android:layout\_marginTop="25dp"

android:src="@drawable/profile\_placeholder"

app:civ\_border\_color="@color/dasarpink"

app:civ\_border\_width="2dp" />

<TextView

android:id="@+id/popup\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="10dp"

android:text="..."

android:textColor="@android:color/white"

android:textSize="21dp"

android:textStyle="bold" />

<TextView

android:id="@+id/popup\_user\_email"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="5dp"

android:fontFamily="@font/roboto\_light"

android:text="..."

android:textColor="@android:color/white" />

<TextView

android:id="@+id/popup\_user\_phone"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="5dp"

android:fontFamily="@font/roboto\_light"

android:text="..."

android:textColor="@android:color/white" />

</LinearLayout>

<Button

android:id="@+id/btnviewprofile"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="25dp"

android:layout\_marginBottom="20dp"

android:background="@drawable/background\_rounded2"

android:text="View Profile"

android:textAllCaps="false"

android:textColor="@android:color/white" />

</LinearLayout>

* + 1. **fragment\_bottom\_sheet\_lapor**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/fragment\_history\_menu\_bottom"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:theme="@style/AppBottomSheetDialogTheme"

app:layout\_behavior="android.support.design.widget.BottomSheetBehavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:gravity="center"

android:orientation="horizontal"

android:padding="16dp">

<View

android:layout\_width="50dp"

android:layout\_height="5dp"

android:background="@drawable/background\_rounded1" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_report"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_highlight\_off\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="Report this Post" />

</LinearLayout>

</LinearLayout>

* + 1. **fragment\_near**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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:id="@+id/containerNear"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Fragment.NearFragment">

<android.support.v4.widget.SwipeRefreshLayout

android:id="@+id/pull"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.v7.widget.RecyclerView

android:id="@+id/near\_list\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

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"></android.support.v7.widget.RecyclerView>

<ImageView

android:id="@+id/icongps"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:layout\_marginStart="8dp"

android:layout\_marginEnd="8dp"

android:visibility="visible"

app:layout\_constraintEnd\_toEndOf="@+id/near\_list\_view"

app:layout\_constraintStart\_toStartOf="@+id/near\_list\_view"

app:layout\_constraintTop\_toBottomOf="@+id/empty\_post"

app:srcCompat="@drawable/pinmu1" />

<TextView

android:id="@+id/empty\_post"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginStart="32dp"

android:layout\_marginTop="100dp"

android:layout\_marginEnd="32dp"

android:fontFamily="@font/comfortaa"

android:gravity="center\_vertical|center\_horizontal"

android:paddingLeft="50dp"

android:paddingRight="50dp"

android:text="can't get your location or GPS IS DISABLE. refresh or CLICK THIS ICON"

android:textAllCaps="true"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="@+id/near\_list\_view"

app:layout\_constraintStart\_toStartOf="@+id/near\_list\_view"

app:layout\_constraintTop\_toTopOf="@+id/near\_list\_view" />

</android.support.constraint.ConstraintLayout>

</android.support.v4.widget.SwipeRefreshLayout>

</FrameLayout>

* + 1. **near\_row**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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="wrap\_content"

android:layout\_marginBottom="8dp">

<android.support.v7.widget.CardView

android:id="@+id/card\_near"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingTop="10dp"

android:paddingBottom="10dp">

<android.support.constraint.ConstraintLayout

android:id="@+id/linearLayout3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<TextView

android:id="@+id/near\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginTop="0dp"

android:layout\_marginBottom="0dp"

android:hint="..."

android:textColor="@android:color/black"

android:textSize="16sp"

android:textStyle="bold"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/textView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="4dp"

android:text="@string/filled\_bullet"

android:textSize="8sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/near\_username"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/near\_time"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="4dp"

android:fontFamily="@font/roboto\_light"

android:gravity="center"

android:text="..."

android:textSize="12sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/textView5"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.6" />

</android.support.constraint.ConstraintLayout>

<RelativeLayout

android:id="@+id/relativeLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:orientation="horizontal"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/linearLayout3">

<TextView

android:id="@+id/near\_desc"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_toLeftOf="@+id/near\_user\_image"

android:inputType="textMultiLine"

android:paddingTop="8dp"

android:paddingBottom="8dp"

android:text="..."

android:textSize="14sp" />

<ImageView

android:id="@+id/near\_user\_image"

android:layout\_width="90dp"

android:layout\_height="90dp"

android:layout\_alignParentEnd="true"

android:layout\_marginEnd="10dp"

android:src="@drawable/profile\_placeholder" />

</RelativeLayout>

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:layout\_marginBottom="4dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout">

<ImageView

android:id="@+id/see\_map"

android:layout\_width="30dp"

android:layout\_height="30dp"

android:layout\_marginStart="8dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/mapicon" />

<TextView

android:id="@+id/jarak"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_toRightOf="@+id/see\_map"

android:fontFamily="@font/pacifico\_regular"

android:hint="..."

android:textColor="@android:color/black"

android:textSize="14sp"

android:textStyle="bold"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/see\_map" />

<TextView

android:id="@+id/view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:text="|"

android:textSize="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/jarak" />

<TextView

android:id="@+id/near\_address"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginEnd="4dp"

android:ellipsize="end"

android:maxLines="1"

android:text="..."

android:textSize="12sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/view" />

</android.support.constraint.ConstraintLayout>

</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

</android.support.constraint.ConstraintLayout>

* + 1. **activity\_maps**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical"

tools:context=".Activities.MapsActivity">

<fragment

android:id="@+id/mapNearBy"

android:name="com.google.android.gms.maps.MapFragment"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1" />

<Button

android:id="@+id/btnGetDirection"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/selectableItemBackground"

android:text="Get Direction" />

</LinearLayout>

* + 1. **fragment\_chats**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

tools:context=".Fragment.ChatsFragment">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.design.widget.TabLayout

android:id="@+id/tablay"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorPrimary"

app:tabIndicatorColor="@color/putih"

app:tabSelectedTextColor="@color/putih"

app:tabTextColor="@color/putih">

</android.support.design.widget.TabLayout>

<android.support.v4.view.ViewPager

android:id="@+id/view\_pager"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

</android.support.v4.view.ViewPager>

</LinearLayout>

</FrameLayout>

* + 1. **fragment\_vpchat**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

tools:context=".Fragment.VPChatsFragment">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_vpchat"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

</android.support.v7.widget.RecyclerView>

<TextView

android:id="@+id/no\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginTop="32dp"

android:fontFamily="@font/comfortaa"

android:gravity="center\_vertical|center\_horizontal"

android:paddingLeft="50dp"

android:paddingRight="50dp"

android:text="No Messages Yet"

android:textAllCaps="true"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<ImageView

android:id="@+id/no\_msg\_img"

android:layout\_width="wrap\_content"

android:layout\_height="100dp"

android:layout\_centerInParent="true"

android:src="@drawable/nothing2"

android:textColor="@color/colorAccent"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/no\_msg" />

</android.support.constraint.ConstraintLayout>

</FrameLayout>

* + 1. **fragment\_vpchat\_user**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

tools:context=".Fragment.VPUsersChatFragment">

<android.support.v4.widget.SwipeRefreshLayout

android:id="@+id/pull1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<EditText

android:id="@+id/src\_users"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="5dp"

android:background="@drawable/background\_seach"

android:drawableLeft="@mipmap/icon\_search"

android:hint="Search..." />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_chatsusers"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@+id/src\_users">

</android.support.v7.widget.RecyclerView>

</RelativeLayout>

</android.support.v4.widget.SwipeRefreshLayout>

</FrameLayout>

* + 1. **userchats\_row**

<?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"

android:id="@+id/rl\_userchatrow"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/chat\_userimage"

android:layout\_width="50dp"

android:layout\_height="50dp"

android:src="@drawable/profile\_placeholder" />

<TextView

android:id="@+id/chat\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_toEndOf="@+id/chat\_userimage"

android:layout\_toRightOf="@id/chat\_userimage"

android:maxLines="1"

android:text="username"

android:textSize="18dp" />

<TextView

android:id="@+id/chat\_timestamp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_marginEnd="5dp"

android:fontFamily="@font/aref\_ruqaa"

android:maxLines="1"

android:text="--"

android:textSize="13dp" />

<TextView

android:id="@+id/last\_msg"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/chat\_username"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/chat\_userimage"

android:maxLines="1"

android:paddingTop="5dp"

android:textSize="12sp" />

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/img\_on"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_below="@+id/chat\_username"

android:layout\_marginLeft="-15dp"

android:layout\_marginTop="10dp"

android:layout\_toEndOf="@+id/chat\_userimage"

android:layout\_toRightOf="@+id/chat\_userimage"

android:src="@drawable/profile\_placeholder"

android:visibility="gone"

app:civ\_border\_color="@color/colorGreen\_A400"

app:civ\_border\_width="10dp" />

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/img\_off"

android:layout\_width="15dp"

android:layout\_height="15dp"

android:layout\_below="@+id/chat\_username"

android:layout\_marginLeft="-15dp"

android:layout\_marginTop="10dp"

android:layout\_toEndOf="@+id/chat\_userimage"

android:layout\_toRightOf="@+id/chat\_userimage"

android:src="@drawable/profile\_placeholder"

android:visibility="gone"

app:civ\_border\_color="#eaeaea"

app:civ\_border\_width="10dp" />

<View

android:layout\_width="match\_parent"

android:layout\_height="1dp"

android:layout\_below="@+id/chat\_userimage"

android:layout\_marginTop="5dp"

android:layout\_toRightOf="@+id/chat\_userimage"

android:background="@color/grey" />

</RelativeLayout>

* + 1. **activity\_message**

<?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"

tools:context=".Activities.MessageActivity">

<android.support.design.widget.AppBarLayout

android:id="@+id/bar\_lay"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar\_msg"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorPrimary"

android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"

app:popupTheme="@style/MenuStyle">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/image\_profile\_msg"

android:layout\_width="35dp"

android:layout\_height="35dp" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="35dp"

android:layout\_marginStart="25dp">

<TextView

android:id="@+id/username\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@android:color/white"

android:textSize="16sp" />

<TextView

android:id="@+id/user\_online"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/username\_msg"

android:layout\_alignParentBottom="true"

android:text="online"

android:textSize="10sp"

android:visibility="gone" />

</RelativeLayout>

</android.support.v7.widget.Toolbar>

</android.support.design.widget.AppBarLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_msg"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_above="@+id/rela"

android:layout\_below="@+id/bar\_lay">

</android.support.v7.widget.RecyclerView>

<RelativeLayout

android:id="@+id/rela"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:background="@android:color/white"

android:padding="5dp">

<EditText

android:id="@+id/text\_send"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_toLeftOf="@+id/send\_msg"

android:background="@android:color/transparent"

android:hint="Type a message..."

android:padding="10dp" />

<ImageButton

android:id="@+id/send\_msg"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_alignParentEnd="true"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:background="@android:color/transparent"

android:src="@drawable/baseline\_send\_24" />

</RelativeLayout>

</RelativeLayout>

* + 1. **chat\_item\_left**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:padding="8dp">

<TextView

android:id="@+id/show\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginVertical="3dp"

android:layout\_marginStart="5dp"

android:layout\_toRightOf="@+id/profile\_chat\_img"

android:background="@drawable/background\_left"

android:padding="8dp"

android:text="helo"

android:textSize="18sp" />

<TextView

android:id="@+id/time\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/show\_msg"

android:layout\_marginStart="7dp"

android:text="00:00"

android:textColor="#a0a0a0"

android:textSize="12sp" />

<TextView

android:id="@+id/txt\_seen"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/show\_msg"

android:visibility="gone" />

</RelativeLayout>

* + 1. **chat\_item\_right**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="8dp">

<RelativeLayout

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentRight="true">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/profile\_chat\_img"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:visibility="gone" />

<TextView

android:id="@+id/show\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:background="@drawable/background\_right"

android:padding="8dp"

android:text="hello"

android:textColor="@android:color/white"

android:textSize="18sp" />

<TextView

android:id="@+id/time\_msg"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/show\_msg"

android:layout\_alignParentEnd="true"

android:layout\_marginRight="6dp"

android:text="00:00"

android:textColor="#a0a0a0"

android:textSize="12sp" />

<TextView

android:id="@+id/txt\_seen"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/show\_msg"

android:layout\_alignParentRight="true" />

</RelativeLayout>

</RelativeLayout>

* + 1. **fragment\_account**

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout 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"

tools:context=".Fragment.AccountFragment">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.constraint.ConstraintLayout

android:id="@+id/constraintLayout2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/grey"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<View

android:id="@+id/lini"

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:background="@drawable/drawable\_gradient\_animation\_list"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<com.mikhaellopez.circularimageview.CircularImageView

android:id="@+id/profile\_image\_view"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_marginStart="8dp"

android:elevation="15dp"

android:src="@drawable/profile\_placeholder"

app:civ\_border\_color="@android:color/white"

app:civ\_border\_width="1dp"

app:civ\_shadow="true"

app:civ\_shadow\_color="@color/colorPrimary"

app:layout\_constraintBottom\_toBottomOf="@+id/lini"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/lini" />

<Button

android:id="@+id/options"

android:layout\_width="63dp"

android:layout\_height="22dp"

android:layout\_below="@+id/lini"

android:layout\_alignParentEnd="true"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="10dp"

android:background="@drawable/background\_rounded2"

android:text="Options"

android:textAllCaps="false"

android:textColor="@color/colorPrimary"

android:textSize="12sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/lini" />

<TextView

android:id="@+id/username\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:hint="..."

android:text="---"

android:textColor="@android:color/black"

android:textSize="21sp"

android:textStyle="bold"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/profile\_image\_view" />

<TextView

android:id="@+id/email\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:fontFamily="@font/roboto\_light"

android:hint="..."

android:textColor="@android:color/black"

app:layout\_constraintStart\_toEndOf="@+id/imageView2"

app:layout\_constraintTop\_toBottomOf="@+id/username\_view" />

<TextView

android:id="@+id/phone\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:fontFamily="monospace"

android:hint="..."

android:text="..."

android:textColor="@android:color/black"

android:typeface="sans"

app:layout\_constraintStart\_toEndOf="@+id/imageView3"

app:layout\_constraintTop\_toBottomOf="@+id/email\_view" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:src="@mipmap/icon\_email"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/username\_view" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

android:src="@mipmap/icon\_phone"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView2" />

</android.support.constraint.ConstraintLayout>

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/grey"

android:gravity="center\_horizontal"

android:padding="5dp"

android:text="POSTS"

android:textColor="@color/colorPrimary"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout2" />

<TextView

android:id="@+id/empty"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginTop="16dp"

android:fontFamily="@font/comfortaa"

android:text="No Post Yet"

android:textColor="@android:color/black"

android:textSize="30sp"

android:textStyle="bold"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<ImageView

android:id="@+id/emptyimg"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_centerInParent="true"

android:layout\_marginTop="8dp"

android:src="@drawable/nothing2"

android:textColor="@color/colorAccent"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/empty" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_images"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

</android.support.constraint.ConstraintLayout>

</FrameLayout>

* + 1. **activity\_about\_us**

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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"

android:background="@drawable/bg\_pattern"

tools:context=".Activities.AboutUsActivity">

<android.support.v7.widget.CardView

android:id="@+id/cv\_one\_login"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="4dp"

android:layout\_marginEnd="4dp"

android:elevation="10dp"

app:cardBackgroundColor="#fcfcfc"

app:cardCornerRadius="10dp"

app:cardElevation="10dp"

app:cardPreventCornerOverlap="false"

app:cardUseCompatPadding="true"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

android:id="@+id/imageView"

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="4dp"

android:layout\_marginEnd="8dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.454"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/logoutama" />

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:orientation="horizontal"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView">

<RelativeLayout

android:id="@+id/c1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.5">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/foto\_aris"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:layout\_centerHorizontal="true"

android:src="@drawable/profile\_placeholder" />

<TextView

android:id="@+id/textView2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/foto\_aris"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:gravity="center\_horizontal"

android:text="Dibuat Oleh:"

android:textColor="@android:color/black"

android:textSize="18sp" />

<TextView

android:id="@+id/textView3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/textView2"

android:layout\_marginStart="8dp"

android:layout\_marginTop="4dp"

android:layout\_marginEnd="8dp"

android:gravity="center\_horizontal"

android:text="Aris Bimas Nugroho"

android:textColor="@android:color/black"

android:textSize="18sp" />

<TextView

android:id="@+id/textView4"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/textView3"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="8dp"

android:gravity="center\_horizontal"

android:text="arisbimas2@gmail.com"

android:textColor="@color/colorPrimaryDark"

android:textSize="14sp" />

<Button

android:id="@+id/btnOpenWhatsapp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:background="@drawable/background\_rounded\_green"

android:drawableLeft="@drawable/icons\_wa"

android:padding="2dp"

android:text="Contact Person"

android:textAllCaps="false"

android:textColor="@android:color/black"

android:textSize="12sp"

android:layout\_below="@+id/textView4"

android:layout\_centerHorizontal="true"/>

</RelativeLayout>

<View

android:layout\_width="2dp"

android:layout\_height="match\_parent"

android:background="@color/colorPrimary"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<RelativeLayout

android:id="@+id/c2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.5">

<TextView

android:id="@+id/kegunaan"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/comfortaa"

android:gravity="left"

android:padding="10dp"

android:text="@string/tujuanapp"

android:textColor="@android:color/black"

android:textSize="16sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/showKentuntuan"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="32dp"

android:layout\_marginBottom="16dp"

android:background="@drawable/background\_rounded\_green"

android:padding="2dp"

android:text="Ketentuan Aplikasi"

android:textAllCaps="false"

android:textColor="@android:color/black"

android:textSize="12sp"

android:layout\_below="@+id/kegunaan"

android:layout\_centerHorizontal="true"/>

</RelativeLayout>

</LinearLayout>

</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

</android.support.constraint.ConstraintLayout>

* + 1. **activity\_detail\_profile**

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout 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="@android:color/white"

android:fitsSystemWindows="true"

tools:context=".Activities.DetailProfileActivity">

<android.support.design.widget.AppBarLayout

android:id="@+id/appbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/AppTheme.AppBarOverlay">

<android.support.design.widget.CollapsingToolbarLayout

android:id="@+id/collapsingToolbar"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:elevation="20dp"

app:collapsedTitleGravity="start"

app:contentScrim="?attr/colorPrimary"

app:expandedTitleMarginBottom="16dp"

app:expandedTitleMarginEnd="64dp"

app:expandedTitleMarginStart="24dp"

app:layout\_scrollFlags="exitUntilCollapsed|scroll">

<android.support.constraint.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@android:color/white">

<android.support.constraint.ConstraintLayout

android:id="@+id/constraintLayout2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/grey"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<View

android:id="@+id/lini"

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:background="@drawable/drawable\_gradient\_animation\_list"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<com.mikhaellopez.circularimageview.CircularImageView

android:id="@+id/profile\_image\_view"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_marginStart="8dp"

android:elevation="15dp"

android:src="@drawable/profile\_placeholder"

app:civ\_border\_color="@android:color/white"

app:civ\_border\_width="1dp"

app:civ\_shadow="true"

app:civ\_shadow\_color="@color/colorPrimary"

app:layout\_constraintBottom\_toBottomOf="@+id/lini"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/lini" />

<TextView

android:id="@+id/username\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:hint="..."

android:text="---"

android:textColor="@android:color/black"

android:textSize="21sp"

android:textStyle="bold"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/profile\_image\_view" />

<TextView

android:id="@+id/email\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:fontFamily="@font/roboto\_light"

android:hint="..."

android:textColor="@android:color/black"

app:layout\_constraintStart\_toEndOf="@+id/imageView2"

app:layout\_constraintTop\_toBottomOf="@+id/username\_view" />

<TextView

android:id="@+id/phone\_view"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:fontFamily="monospace"

android:hint="..."

android:text="..."

android:textColor="@android:color/black"

android:typeface="sans"

app:layout\_constraintStart\_toEndOf="@+id/imageView3"

app:layout\_constraintTop\_toBottomOf="@+id/email\_view" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:src="@mipmap/icon\_email"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/username\_view" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_marginStart="12dp"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="8dp"

android:src="@mipmap/icon\_phone"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/imageView2" />

</android.support.constraint.ConstraintLayout>

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/grey"

android:gravity="center\_horizontal"

android:padding="5dp"

android:text="POSTS"

android:textColor="@color/colorPrimary"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/constraintLayout2" />

<TextView

android:id="@+id/empty"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginTop="16dp"

android:fontFamily="@font/amaticsc\_egular"

android:text="No Post Yet"

android:textColor="@android:color/black"

android:textSize="30sp"

android:textStyle="bold"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<ImageView

android:id="@+id/emptyimg"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_centerInParent="true"

android:layout\_marginTop="8dp"

android:src="@drawable/nothing2"

android:textColor="@color/colorAccent"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/empty" />

</android.support.constraint.ConstraintLayout>

<android.support.v7.widget.Toolbar

android:id="@+id/detail\_profile\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

app:layout\_collapseMode="pin"

app:popupTheme="@style/AppTheme.PopupOverlay"></android.support.v7.widget.Toolbar>

</android.support.design.widget.CollapsingToolbarLayout>

</android.support.design.widget.AppBarLayout>

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_images"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:scrollbars="vertical"

app:layout\_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior" />

</android.support.design.widget.CoordinatorLayout>

* + 1. **activity\_edit\_post**

<?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"

tools:context=".Activities.NewPostActivity">

<android.support.v7.widget.Toolbar

android:id="@+id/edit\_post\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorPrimary"

android:minHeight="?attr/actionBarSize"

app:titleTextColor="@android:color/white" />

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@+id/edit\_post\_toolbar"

android:layout\_alignParentStart="true"

android:layout\_marginStart="0dp"

android:layout\_marginTop="0dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageButton

android:id="@+id/post\_image"

android:layout\_width="match\_parent"

android:layout\_height="250dp"

android:layout\_alignParentStart="true"

android:layout\_alignParentTop="true"

android:layout\_marginStart="0dp"

android:layout\_marginTop="-5dp"

android:adjustViewBounds="true"

android:background="@android:color/white"

android:scaleType="centerCrop"

android:src="@mipmap/add\_btn">

</ImageButton>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/post\_image"

android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

app:counterEnabled="true"

app:counterMaxLength="140">

<EditText

android:id="@+id/post\_desc"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_post\_desc"

android:maxLength="140"

android:padding="20dp" />

</android.support.design.widget.TextInputLayout>

</RelativeLayout>

</ScrollView>

</RelativeLayout>

* + 1. **fragment\_bottom\_sheet\_options**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/fragment\_history\_menu\_bottom"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:theme="@style/AppBottomSheetDialogTheme"

app:layout\_behavior="android.support.design.widget.BottomSheetBehavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:gravity="center"

android:orientation="horizontal"

android:padding="16dp">

<View

android:layout\_width="50dp"

android:layout\_height="5dp"

android:background="@drawable/background\_rounded1" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_edit\_profile"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_edit\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="Edit Profile" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_logout"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_highlight\_off\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="Log Out" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_about"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_developer\_mode\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="About App" />

</LinearLayout>

</LinearLayout>

* + 1. **fragment\_bottom\_sheet\_account**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/fragment\_history\_menu\_bottom"

android:layout\_width="match\_parent"

android:layout\_height="180dp"

android:orientation="vertical"

android:theme="@style/AppBottomSheetDialogTheme"

app:layout\_behavior="android.support.design.widget.BottomSheetBehavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:gravity="center"

android:orientation="horizontal"

android:padding="16dp">

<View

android:layout\_width="50dp"

android:layout\_height="5dp"

android:background="@drawable/background\_rounded1" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_edit\_post"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_edit\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="Edit Post" />

</LinearLayout>

<LinearLayout

android:id="@+id/bottom\_sheet\_delete\_post"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

android:orientation="horizontal"

android:padding="16dp">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

app:srcCompat="@drawable/baseline\_delete\_black\_24" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginLeft="16dp"

android:text="Delete Post" />

</LinearLayout>

</LinearLayout>

* + 1. **popup\_about \_app**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="350dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:background="@color/grey"

android:orientation="vertical"

android:padding="5dp">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:text="@string/ketentuan1"

android:textColor="@android:color/black" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:text="@string/ketentuan2"

android:textColor="@android:color/black" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:text="@string/ketentuan3"

android:textColor="@android:color/black" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:text="@string/ketentuan4"

android:textColor="@android:color/black" />

</LinearLayout>

</ScrollView>

</LinearLayout>

* + 1. **SplashScreenActivity**

package com.aris.crowdreporting.Activities;

import android.graphics.Color;

import android.graphics.Typeface;

import android.support.v4.content.res.ResourcesCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import com.aris.crowdreporting.BuildConfig;

import com.aris.crowdreporting.R;

import gr.net.maroulis.library.EasySplashScreen;

public class SplashScreenActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

String versionName = BuildConfig.VERSION\_NAME;

EasySplashScreen easySplashScreen = new EasySplashScreen(SplashScreenActivity.this)

.withFullScreen()

.withTargetActivity(MainActivity.class)

.withSplashTimeOut(3000)

.withBackgroundResource(R.drawable.gradient3)

.withLogo(R.drawable.logoutama)

.withFooterText("Version "+versionName)

.withAfterLogoText("By: Aris Bimas Nugroho");

//add custom font

Typeface typeface = ResourcesCompat.getFont(this, R.font.comfortaa);

easySplashScreen.getAfterLogoTextView().setTypeface(typeface);

easySplashScreen.getFooterTextView().setTypeface(typeface);

easySplashScreen.getAfterLogoTextView().setTextColor(Color.BLACK);

easySplashScreen.getFooterTextView().setTextColor(Color.WHITE);

easySplashScreen.getLogo().setMaxHeight(550);

easySplashScreen.getLogo().setMaxWidth(550);

View view = easySplashScreen.create();

setContentView(view);

}

}

* + 1. **LoginActivity**

package com.aris.crowdreporting.Activities;

import android.app.ProgressDialog;

import android.content.Intent;

import android.graphics.drawable.AnimationDrawable;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.RelativeLayout;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.google.android.gms.auth.api.signin.GoogleSignIn;

import com.google.android.gms.auth.api.signin.GoogleSignInAccount;

import com.google.android.gms.auth.api.signin.GoogleSignInClient;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.gms.common.SignInButton;

import com.google.android.gms.common.api.ApiException;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthCredential;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.GoogleAuthProvider;

public class LoginActivity extends AppCompatActivity {

private EditText mNama, mEmail, mPass;

private ImageButton mSignIn;

private Button mRegis;

private ProgressDialog mDialog;

private ProgressBar progressBar;

private ProgressDialog pDialog;

private FirebaseAuth mAuth;

private GoogleSignInClient mGoogleSignInClient;

private static final int RC\_SIGN\_IN = 9001;

SignInButton signInButton;

AnimationDrawable animationDrawable;

RelativeLayout frameLayout;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

mAuth = FirebaseAuth.getInstance();

mEmail = (EditText) findViewById(R.id.login\_email);

mPass = (EditText) findViewById(R.id.login\_password);

mSignIn = (ImageButton) findViewById(R.id.btn\_login);

mRegis = (Button) findViewById(R.id.btn\_signup);

progressBar = (ProgressBar) findViewById(R.id.progressBar);

pDialog = new ProgressDialog(LoginActivity.this);

//Get Firebase auth instance

mAuth = FirebaseAuth.getInstance();

mRegis.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent pindah = new Intent(LoginActivity.this, RegisterActivity.class);

pindah.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(pindah);

}

});

mSignIn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String email = mEmail.getText().toString();

final String password = mPass.getText().toString();

if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)){

progressBar.setVisibility(View.VISIBLE);

mAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

sendMain();

} else {

String errMasg = task.getException().getMessage();

Toast.makeText(LoginActivity.this, "Error " + errMasg, Toast.LENGTH\_SHORT).show();

}

progressBar.setVisibility(View.INVISIBLE);

}

});

}

else if (TextUtils.isEmpty(email)) {

Toast.makeText(getApplicationContext(), "Enter email address!", Toast.LENGTH\_SHORT).show();

return;

}else if (TextUtils.isEmpty(password)) {

Toast.makeText(getApplicationContext(), "Enter password!", Toast.LENGTH\_SHORT).show();

return;

}

}

});

// Configure Google Sign In

GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)

.requestIdToken(getString(R.string.default\_web\_client\_id))

.requestEmail()

.build();

mGoogleSignInClient = GoogleSignIn.getClient(this,gso);

signInButton = (SignInButton)findViewById(R.id.sign\_in\_button);

signInButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

signIn();

}

});

}

private void displayProgressDialog() {

pDialog.setMessage("Logging In.. Please wait...");

pDialog.setIndeterminate(false);

pDialog.setCancelable(false);

pDialog.show();

}

private void signIn() {

Intent signInIntent = mGoogleSignInClient.getSignInIntent();

startActivityForResult(signInIntent, RC\_SIGN\_IN);

displayProgressDialog();

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

// Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);

if (requestCode == RC\_SIGN\_IN) {

Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);

try {

// Google Sign In was successful, authenticate with Firebase

GoogleSignInAccount account = task.getResult(ApiException.class);

firebaseAuthWithGoogle(account);

} catch (ApiException e) {

// Google Sign In failed, update UI appropriately

// ...

}

}

}

private void firebaseAuthWithGoogle(GoogleSignInAccount account) {

displayProgressDialog();

AuthCredential credential = GoogleAuthProvider.getCredential(account.getIdToken(), null);

mAuth.signInWithCredential(credential)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

// Sign in success, update UI with the signed-in user's information

// Log.d(TAG, "signInWithCredential:success");

FirebaseUser user = mAuth.getCurrentUser();

sendMain();

Toast.makeText(LoginActivity.this, "Logged in Successfully", Toast.LENGTH\_SHORT).show();

} else {

// If sign in fails, display a message to the user.

Toast.makeText(LoginActivity.this, "Could not Login User", Toast.LENGTH\_SHORT).show();

}

// ...

}

});

}

@Override

protected void onStart() {

super.onStart();

if (mAuth.getCurrentUser() != null) {

sendMain();

}

}

private void sendMain() {

Intent intent = new Intent(LoginActivity.this, MainActivity.class);

startActivity(intent);

finish();

}

}

* + 1. **RegisterActivity**

package com.aris.crowdreporting.Activities;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.Toast;

import com.aris.crowdreporting.R;

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.FirebaseUser;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import dmax.dialog.SpotsDialog;

public class RegisterActivity extends AppCompatActivity {

private FirebaseAuth.AuthStateListener mAuthStateListener;

private FirebaseAuth mAuth;

private DatabaseReference mDatabase;

private CheckBox term;

private ProgressBar mProgress;

private EditText mName, mEmail, mPass, mPassConfirm;

private Button mAlready;

private ImageButton mBtnRegis;

private SpotsDialog dialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

mAuth = FirebaseAuth.getInstance();

mDatabase = FirebaseDatabase.getInstance().getReference().child("Users");

mEmail = (EditText)findViewById(R.id.reg\_email);

mPass = (EditText)findViewById(R.id.reg\_pass);

mPassConfirm = (EditText)findViewById(R.id.reg\_confirm\_pass);

mBtnRegis = (ImageButton) findViewById(R.id.btn\_reg);

mAlready = (Button) findViewById(R.id.back\_login);

term = (CheckBox)findViewById(R.id.term);

mProgress = (ProgressBar) findViewById(R.id.progressBar);

mAlready.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

mBtnRegis.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String email = mEmail.getText().toString();

String pass = mPass.getText().toString();

String confirmpass = mPassConfirm.getText().toString();

if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(pass)

&& !TextUtils.isEmpty(confirmpass) && term.isChecked()){

if (pass.equals(confirmpass)){

// mProgress.setVisibility(View.VISIBLE);

dialog = new SpotsDialog(RegisterActivity.this, "Please Wait.");

dialog.show();

mAuth.createUserWithEmailAndPassword(email, pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()){

mAuth.getCurrentUser().sendEmailVerification()

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Intent setupIntent = new Intent(RegisterActivity.this, VerificationEmailActivity.class);

startActivity(setupIntent);

finish();

}

}

});

} else {

String errMsg = task.getException().getMessage();

Toast.makeText(RegisterActivity.this, "Error ! " + errMsg, Toast.LENGTH\_SHORT).show();

}

// mProgress.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

});

} else {

Toast.makeText(RegisterActivity.this, "Password and Confirm Password not Match!", Toast.LENGTH\_SHORT).show();

}

}else if (TextUtils.isEmpty(email)) {

Toast.makeText(getApplicationContext(), "Enter email address!", Toast.LENGTH\_SHORT).show();

return;

}else if (TextUtils.isEmpty(pass)) {

Toast.makeText(getApplicationContext(), "Enter password!", Toast.LENGTH\_SHORT).show();

return;

} else if (!term.isChecked()){

Toast.makeText(RegisterActivity.this, "Term and Conditions is not Checked", Toast.LENGTH\_SHORT).show();

return;

}

}

});

}

@Override

protected void onStart() {

super.onStart();

FirebaseUser currentUser = mAuth.getCurrentUser();

if (currentUser != null){

sendMain();

}

}

private void sendMain() {

Intent intent = new Intent(RegisterActivity.this, MainActivity.class);

startActivity(intent);

finish();

}

}

* + 1. **ReAuthEmailActivity**

package com.aris.crowdreporting.Activities;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import dmax.dialog.SpotsDialog;

public class ReAuthEmailActivity extends AppCompatActivity {

private EditText editEmail;

private ImageButton btnEditEmail;

private SpotsDialog spotsDialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_re\_auth\_email);

editEmail = findViewById(R.id.edit\_emailveri);

btnEditEmail = findViewById(R.id.btn\_edit\_email);

btnEditEmail.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

spotsDialog = new SpotsDialog(ReAuthEmailActivity.this, "Please Wait.");

spotsDialog.show();

String email = editEmail.getText().toString();

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

user.updateEmail(email)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(ReAuthEmailActivity.this, "User email address updated.", Toast.LENGTH\_SHORT).show();

Intent setupIntent = new Intent(ReAuthEmailActivity.this, VerificationEmailActivity.class);

startActivity(setupIntent);

finish();

spotsDialog.dismiss();

}

}

});

}

});

}

}

* + 1. **VerificationEmailActivity**

package com.aris.crowdreporting.Activities;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.Html;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import dmax.dialog.SpotsDialog;

public class VerificationEmailActivity extends AppCompatActivity {

private FirebaseAuth mAuth;

private FirebaseUser currentUser;

private Button nextBtn, editEmail, exit;

private TextView tvEmail;

private SpotsDialog dialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_verification\_email);

mAuth = FirebaseAuth.getInstance();

currentUser = mAuth.getCurrentUser();

nextBtn = findViewById(R.id.btn\_next);

tvEmail = findViewById(R.id.tv\_email);

editEmail = findViewById(R.id.edit\_email);

exit = findViewById(R.id.ext);

String emailUser = "We've sent a link. Please check your email account for " + "<b>" + currentUser.getEmail() + "</b>" + " and click the link inside.";

tvEmail.setText(Html.fromHtml(emailUser));

editEmail.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(VerificationEmailActivity.this, ReAuthEmailActivity.class));

}

});

nextBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Toast.makeText(VerificationEmailActivity.this, "" + mAuth.getCurrentUser().getEmail(), Toast.LENGTH\_SHORT).show();

dialog = new SpotsDialog(VerificationEmailActivity.this, "Please Wait.");

dialog.show();

mAuth.getCurrentUser().reload().addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

if (mAuth.getCurrentUser().isEmailVerified()) {

Toast.makeText(VerificationEmailActivity.this, "" + currentUser.getEmail(), Toast.LENGTH\_SHORT).show();

Intent setupIntent = new Intent(VerificationEmailActivity.this, SetupActivity.class);

startActivity(setupIntent);

finish();

dialog.dismiss();

} else {

Toast.makeText(VerificationEmailActivity.this, "Please Verified Your Email.", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

}

});

}

});

exit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dialog = new SpotsDialog(VerificationEmailActivity.this, "Please Wait");

dialog.show();

FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();

FirebaseUser user = firebaseAuth.getCurrentUser();

user.reload();

if (!user.isEmailVerified()) {

user.sendEmailVerification()

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(VerificationEmailActivity.this, "Link Sent!", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

}

});

} else {

Toast.makeText(VerificationEmailActivity.this, "Your Email is Verified, Click Next Button.", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

}

});

}

}

* + 1. **SetupActivity**

package com.aris.crowdreporting.Activities;

import android.Manifest;

import android.app.AlertDialog;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.net.Uri;

import android.os.Build;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.Gravity;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.tasks.Continuation;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import com.theartofdev.edmodo.cropper.CropImage;

import com.theartofdev.edmodo.cropper.CropImageView;

import java.util.HashMap;

import java.util.Map;

import de.hdodenhof.circleimageview.CircleImageView;

import dmax.dialog.SpotsDialog;

public class SetupActivity extends AppCompatActivity {

private CircleImageView setupImage;

private Uri mainImageUri;

private EditText setupName, setupEmail, setupPhone;

private ImageButton setupBtn;

private String user\_id;

private String emailuser\_id;

private boolean is\_Changed = false;

private StorageReference storageReference;

private FirebaseAuth firebaseAuth;

private FirebaseFirestore firebaseFirestore;

private SpotsDialog dialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_setup);

firebaseAuth = FirebaseAuth.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

storageReference = FirebaseStorage.getInstance().getReference();

user\_id = firebaseAuth.getCurrentUser().getUid();

emailuser\_id = firebaseAuth.getCurrentUser().getEmail();

setupImage = findViewById(R.id.setup\_image);

setupName = (EditText)findViewById(R.id.setup\_name);

setupEmail = (EditText)findViewById(R.id.setup\_email);

setupPhone = (EditText)findViewById(R.id.setup\_phone);

setupBtn = (ImageButton) findViewById(R.id.btn\_setup);

// setupProgress = (ProgressBar)findViewById(R.id.progressBar);

// setupProgress.setVisibility(View.VISIBLE);

dialog = new SpotsDialog(SetupActivity.this);

dialog.show();

setupBtn.setEnabled(false);

firebaseFirestore.collection("Users").document(user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()){

if (task.getResult().exists()){

String name = task.getResult().getString("name");

String phone = task.getResult().getString("phone");

String image = task.getResult().getString("image");

mainImageUri = Uri.parse(image);

setupName.setText(name);

setupEmail.setText(emailuser\_id);

setupPhone.setText(phone);

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.defaultimage);

Glide.with(SetupActivity.this).setDefaultRequestOptions(placeholderReq).load(image).into(setupImage);

}

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(SetupActivity.this, ""+ errMSg, Toast.LENGTH\_SHORT).show();

}

// setupProgress.setVisibility(View.INVISIBLE);

dialog.dismiss();

setupBtn.setEnabled(true);

}

});

setupBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

final String username = setupName.getText().toString();

final String phone = setupPhone.getText().toString();

if (!TextUtils.isEmpty(username) && !TextUtils.isEmpty(phone) && mainImageUri != null) {

// setupProgress.setVisibility(View.VISIBLE);

dialog.show();

if(is\_Changed) {

user\_id = firebaseAuth.getCurrentUser().getUid();

StorageReference image\_path = storageReference.child("profile\_images").child(user\_id + ".jpg");

UploadTask uploadTask = image\_path.putFile(mainImageUri);

Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {

@Override

public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {

if (!task.isSuccessful()) {

throw task.getException();

}

// Continue with the task to get the download URL

return image\_path.getDownloadUrl();

}

}).addOnCompleteListener(new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> task) {

if (task.isSuccessful()) {

storeFirestore(task, user\_id, username, emailuser\_id, phone);

} else {

String errMsg = task.getException().getMessage();

Toast.makeText(SetupActivity.this, "Image Error " + errMsg, Toast.LENGTH\_SHORT).show();

// setupProgress.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

}

});

}else {

storeFirestore(null, user\_id, username, emailuser\_id, phone);

}

} else if (mainImageUri == null){

Toast toast = Toast.makeText(SetupActivity.this, "Please select photo", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

} else if (TextUtils.isEmpty(username)){

Toast toast = Toast.makeText(SetupActivity.this, "Please enter username", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

} else if (TextUtils.isEmpty(phone)){

Toast toast = Toast.makeText(SetupActivity.this, "Please enter phone number", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

}

}

});

setupImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M){

if(ContextCompat.checkSelfPermission(SetupActivity.this, Manifest.permission.READ\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED){

Toast.makeText(SetupActivity.this, "Permission Ok", Toast.LENGTH\_LONG).show();

ActivityCompat.requestPermissions(SetupActivity.this, new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE}, 1);

} else {

BringImagePicker();

}

} else {

BringImagePicker();

}

}

});

}

private void storeFirestore(@NonNull Task<Uri> task,String user\_id, String username, String emailuser\_id, String phone) {

SpotsDialog dialog = new SpotsDialog(SetupActivity.this);

Uri downloadUri;

if (task != null) {

downloadUri = task.getResult();

} else {

downloadUri = mainImageUri;

}

Map<String, String> userMap = new HashMap<>();

userMap.put("user\_id" , user\_id);

userMap.put("name" , username);

userMap.put("email" , emailuser\_id);

userMap.put("phone" , phone);

userMap.put("image", downloadUri.toString());

userMap.put("status", "offline");

firebaseFirestore.collection("Users").document(user\_id).set(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()){

// setupProgress.setVisibility(View.INVISIBLE);

dialog.show();

Toast.makeText(SetupActivity.this, "The User Setting are Updated", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(SetupActivity.this, MainActivity.class);

startActivity(intent);

finish();

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(SetupActivity.this, ""+ errMSg, Toast.LENGTH\_SHORT).show();

}

// setupProgress.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

});

}

private void BringImagePicker() {

// start picker to get image for cropping and then use the image in cropping activity

CropImage.activity()

.setGuidelines(CropImageView.Guidelines.ON)

.setAspectRatio(1, 1)

.start(SetupActivity.this);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == CropImage.CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE) {

CropImage.ActivityResult result = CropImage.getActivityResult(data);

if (resultCode == RESULT\_OK) {

mainImageUri = result.getUri();

setupImage.setImageURI(mainImageUri);

is\_Changed = true;

} else if (resultCode == CropImage.CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE) {

Exception error = result.getError();

}

}

}

}

* + 1. **MainActivity**

package com.aris.crowdreporting.Activities;

import android.Manifest;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.net.Uri;

import android.provider.Settings;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.design.widget.BottomNavigationView;

import android.support.v4.app.Fragment;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.Toolbar;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.Toast;

import com.aris.crowdreporting.Fragment.AccountFragment;

import com.aris.crowdreporting.Fragment.HomeFragment;

import com.aris.crowdreporting.Fragment.NearFragment;

import com.aris.crowdreporting.Fragment.ChatsFragment;

import com.aris.crowdreporting.Fragment.NearNewFragment;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.R;

import com.firebase.client.Firebase;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.karumi.dexter.Dexter;

import com.karumi.dexter.MultiplePermissionsReport;

import com.karumi.dexter.PermissionToken;

import com.karumi.dexter.listener.DexterError;

import com.karumi.dexter.listener.PermissionRequest;

import com.karumi.dexter.listener.PermissionRequestErrorListener;

import com.karumi.dexter.listener.multi.MultiplePermissionsListener;

import java.util.HashMap;

import java.util.List;

public class MainActivity extends AppCompatActivity implements BottomNavigationView.OnNavigationItemSelectedListener {

private static final String TAG = "MainActivity TAG";

private FirebaseAuth.AuthStateListener mAuthStateListener;

private FirebaseAuth mAuth;

private FirebaseFirestore firebaseFirestore;

private Toolbar main\_toolbar;

private String current\_user\_id;

private Status status;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

requestMultiplePermissions();

main\_toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(main\_toolbar);

getSupportActionBar().setTitle("Crowdsourcing");

Firebase.setAndroidContext(this);

mAuth = FirebaseAuth.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

if (mAuth.getCurrentUser() != null){

// kita set default nya Home Fragment

loadFragment(new HomeFragment());

// inisialisasi BottomNavigaionView

BottomNavigationView bottomNavigationView = findViewById(R.id.button\_navigation);

// beri listener pada saat item/menu bottomnavigation terpilih

bottomNavigationView.setOnNavigationItemSelectedListener(this);

// BottomNavigationViewEx bottomNavigationView = (BottomNavigationViewEx) findViewById(R.id.bottomnav);

// beri listener pada saat item/menu bottomnavigation terpilih

// bottomNavigationView.setOnNavigationItemSelectedListener(this);

// bottomNavigationView.enableAnimation(true);

// bottomNavigationView.setTextSize(10);

// bottomNavigationView.setItemHeight(100);

}

}

@Override

protected void onStart() {

super.onStart();

FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();

if (currentUser == null){

sendToLogin();

} else {

if (currentUser.isEmailVerified()){

Log.d(TAG, "email terverifikasi");

current\_user\_id = mAuth.getCurrentUser().getUid();

String usr = "Users";

firebaseFirestore.collection(usr).document(current\_user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()){

if (!task.getResult().exists()){

Intent setupIntent = new Intent(MainActivity.this, SetupActivity.class);

startActivity(setupIntent);

finish();

}

} else {

String errMsg = task.getException().getMessage();

Toast.makeText(MainActivity.this, ""+ errMsg, Toast.LENGTH\_SHORT).show();

}

}

});

} else {

Intent emailVerifIntent = new Intent(MainActivity.this, VerificationEmailActivity.class);

startActivity(emailVerifIntent);

finish();

}

}

}

private void sendToLogin() {

Intent intent = new Intent(MainActivity.this, LoginActivity.class);

startActivity(intent);

finish();

}

private boolean loadFragment(Fragment fragment){

if (fragment != null) {

getSupportFragmentManager().beginTransaction()

.replace(R.id.fragment\_container, fragment)

.commit();

return true;

}

return false;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.action\_bar, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.act\_add) {

Intent intent = new Intent(MainActivity.this, NewPostActivity.class);

startActivity(intent);

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

Fragment fragment = null;

switch (menuItem.getItemId()){

case R.id.nav\_home:

fragment = new HomeFragment();

break;

case R.id.nav\_near:

fragment = new NearFragment();

break;

case R.id.nav\_account:

fragment = new AccountFragment();

break;

case R.id.nav\_chats:

fragment = new ChatsFragment();

break;

}

return loadFragment(fragment);

}

//runtime permission method

private void requestMultiplePermissions(){

Dexter.withActivity(this)

.withPermissions(

Manifest.permission.ACCESS\_FINE\_LOCATION,

Manifest.permission.ACCESS\_COARSE\_LOCATION )

.withListener(new MultiplePermissionsListener() {

@Override

public void onPermissionsChecked(MultiplePermissionsReport report) {

// check if all permissions are granted

if (report.areAllPermissionsGranted()) {

// Toast.makeText(getApplicationContext(), "All permissions are granted by user!", Toast.LENGTH\_SHORT).show();

}

// check for permanent denial of any permission

if (report.isAnyPermissionPermanentlyDenied()) {

// show alert dialog navigating to Settings

openSettingsDialog();

}

}

@Override

public void onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions, PermissionToken token) {

token.continuePermissionRequest();

}

}).

withErrorListener(new PermissionRequestErrorListener() {

@Override

public void onError(DexterError error) {

Toast.makeText(getApplicationContext(), "Some Error! ", Toast.LENGTH\_SHORT).show();

}

})

.onSameThread()

.check();

}

private void openSettingsDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());

builder.setTitle("Required Permissions");

builder.setMessage("This app require permission to use awesome feature. Grant them in app settings.");

builder.setPositiveButton("Take Me To SETTINGS", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

Intent intent = new Intent(Settings.ACTION\_APPLICATION\_DETAILS\_SETTINGS);

Uri uri = Uri.fromParts("package", getPackageName(), null);

intent.setData(uri);

startActivityForResult(intent, 101);

}

});

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

builder.show();

}

}

* + 1. **NewPostActivity**

package com.aris.crowdreporting.Activities;

import android.Manifest;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Intent;

import android.content.IntentSender;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.PorterDuff;

import android.location.Address;

import android.location.Geocoder;

import android.location.Location;

import com.aris.crowdreporting.R;

import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationCallback;

import com.google.android.gms.location.LocationListener;

import android.net.Uri;

import android.os.Looper;

import android.support.annotation.NonNull;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.Toolbar;

import android.text.TextUtils;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.Toast;

import com.developers.imagezipper.ImageZipper;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.common.api.PendingResult;

import com.google.android.gms.common.api.ResultCallback;

import com.google.android.gms.common.api.Status;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationResult;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.location.LocationSettingsRequest;

import com.google.android.gms.location.LocationSettingsResult;

import com.google.android.gms.location.LocationSettingsStatusCodes;

import com.google.android.gms.tasks.Continuation;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import com.theartofdev.edmodo.cropper.CropImage;

import com.theartofdev.edmodo.cropper.CropImageView;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Locale;

import java.util.Map;

import java.util.UUID;

import dmax.dialog.SpotsDialog;

public class NewPostActivity extends AppCompatActivity implements GoogleApiClient.ConnectionCallbacks,

GoogleApiClient.OnConnectionFailedListener,

LocationListener {

private ImageButton mSelectImage;

private EditText mLati, mLongi, mAddress;

private EditText mDesc;

private Button mSubmit;

private Toolbar newPostToolbar;

private ProgressBar progressBar;

private Uri postImageUri = null;

private StorageReference storageReference;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private String current\_user\_id;

private Bitmap imageBitmap;

private Location mylocation;

private GoogleApiClient googleApiClient;

private final static int REQUEST\_CHECK\_SETTINGS\_GPS = 0x1;

private final static int REQUEST\_ID\_MULTIPLE\_PERMISSIONS = 0x2;

private FusedLocationProviderClient fusedLocationClient;

private SpotsDialog dialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_new\_post);

newPostToolbar = findViewById(R.id.comment\_toolbar);

setSupportActionBar(newPostToolbar);

getSupportActionBar().setTitle("Add New Post");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

newPostToolbar.getNavigationIcon().setColorFilter(getResources().getColor(R.color.putih), PorterDuff.Mode.SRC\_ATOP);

storageReference = FirebaseStorage.getInstance().getReference();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

current\_user\_id = firebaseAuth.getCurrentUser().getUid();

mSelectImage = (ImageButton) findViewById(R.id.post\_image);

mLati = (EditText) findViewById(R.id.post\_lati);

mLongi = (EditText) findViewById(R.id.post\_longi);

mDesc = (EditText) findViewById(R.id.post\_desc);

mAddress = (EditText) findViewById(R.id.post\_address);

dialog = new SpotsDialog(NewPostActivity.this, "Track Your Location");

mSelectImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

CropImage.activity()

.setGuidelines(CropImageView.Guidelines.ON)

.setMinCropResultSize(512, 512)

.setAspectRatio(1, 1)

.start(NewPostActivity.this);

}

});

setUpGClient();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.post, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

AlertDialog dialog = new SpotsDialog(NewPostActivity.this, "Uploading...");

//noinspection SimplifiableIfStatement

if (id == R.id.postit) {

String latitude = mLati.getText().toString();

String longitude = mLongi.getText().toString();

String address = mAddress.getText().toString();

String desc = mDesc.getText().toString().toLowerCase();

if (!TextUtils.isEmpty(latitude) && !TextUtils.isEmpty(longitude) && !TextUtils.isEmpty(desc) && postImageUri != null) {

// progressBar.setVisibility(View.VISIBLE);

dialog.show();

String randomName = UUID.randomUUID().toString();

StorageReference file\_path = storageReference.child("post\_image").child(randomName + ".jpg");

UploadTask uploadTask = file\_path.putFile(postImageUri);

Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {

@Override

public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {

if (!task.isSuccessful()) {

throw task.getException();

}

// Continue with the task to get the download URL

return file\_path.getDownloadUrl();

}

}).addOnCompleteListener(new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> task) {

if (task.isSuccessful()) {

File newImageFile = new File(postImageUri.getPath());

try {

File imageZipperFile = new ImageZipper(NewPostActivity.this)

.setQuality(10)

.setMaxWidth(200)

.setMaxHeight(200)

.compressToFile(newImageFile);

} catch (IOException e) {

e.printStackTrace();

}

try {

imageBitmap = new ImageZipper(NewPostActivity.this).compressToBitmap(newImageFile);

} catch (IOException e) {

e.printStackTrace();

}

ByteArrayOutputStream baos = new ByteArrayOutputStream();

imageBitmap.compress(Bitmap.CompressFormat.JPEG, 80, baos);

byte[] thumbData = baos.toByteArray();

final StorageReference ref = storageReference.child("post\_image/thumbs").child(randomName + ".jpg");

UploadTask uploadTask = ref.putBytes(thumbData);

Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {

@Override

public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {

if (!task.isSuccessful()) {

throw task.getException();

}

// Continue with the task to get the download URL

return ref.getDownloadUrl();

}

}).addOnCompleteListener(new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> taskThumb) {

if (taskThumb.isSuccessful()) {

Uri downloadUri = task.getResult();

Uri downloadUriThumb = taskThumb.getResult();

String imageUri = downloadUri.toString();

String imageUriThumb = downloadUriThumb.toString();

String lati = latitude.toString();

String longi = longitude.toString();

String addrs = address;

String des = desc.toString();

Map<String, Object> postMap = new HashMap<>();

postMap.put("image\_uri", imageUri);

postMap.put("image\_thumb", imageUriThumb);

postMap.put("latitude", lati);

postMap.put("longitude", longi);

postMap.put("address", addrs);

postMap.put("desc", des);

postMap.put("user\_id", current\_user\_id);

postMap.put("timestamp", FieldValue.serverTimestamp());

postMap.put("reports", "false");

firebaseFirestore.collection("Posts").add(postMap).addOnCompleteListener(new OnCompleteListener<DocumentReference>() {

@Override

public void onComplete(@NonNull Task<DocumentReference> task) {

if (task.isSuccessful()) {

Toast.makeText(NewPostActivity.this, "Post was Added", Toast.LENGTH\_SHORT).show();

Intent mainIntent = new Intent(NewPostActivity.this, MainActivity.class);

startActivity(mainIntent);

finish();

} else {

}

// progressBar.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

});

} else {

// Handle failures

// ...

}

}

});

} else {

String errMsg = task.getException().getMessage();

Toast.makeText(NewPostActivity.this, "Image Error " + errMsg, Toast.LENGTH\_SHORT).show();

// progressBar.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

}

});

} else if (TextUtils.isEmpty(latitude)) {

Toast.makeText(NewPostActivity.this, "Lokasi Tidak Terditeksi", Toast.LENGTH\_SHORT).show();

} else if (TextUtils.isEmpty(longitude)) {

Toast.makeText(NewPostActivity.this, "Lokasi Tidak Terditeksi", Toast.LENGTH\_SHORT).show();

} else if (TextUtils.isEmpty(desc)) {

Toast.makeText(NewPostActivity.this, "Deskripsi harus diisi!", Toast.LENGTH\_SHORT).show();

} else if (postImageUri == null) {

Toast.makeText(NewPostActivity.this, "Harus disertakan Foto", Toast.LENGTH\_SHORT).show();

}

return true;

}

return super.onOptionsItemSelected(item);

}

private synchronized void setUpGClient() {

googleApiClient = new GoogleApiClient.Builder(this)

.enableAutoManage(this, 0, this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

googleApiClient.connect();

dialog.show();

}

@Override

public void onLocationChanged(Location location) {

mylocation = location;

if (mylocation != null) {

Double latitude = mylocation.getLatitude();

Double longitude = mylocation.getLongitude();

mLati.setText("" + latitude);

mLongi.setText("" + longitude);

//Or Do whatever you want with your location

dialog.dismiss();

Geocoder geocoder = new Geocoder(NewPostActivity.this, Locale.getDefault());

String myCity="";

try {

List<Address> addresses = geocoder.getFromLocation(latitude, longitude,1);

String address = addresses.get(0).getAddressLine(0);

// myCity = addresses.get(0).getLocality();

// Toast.makeText(this, ""+address, Toast.LENGTH\_SHORT).show();

mAddress.getText().clear();

mAddress.setText(address);

} catch (IOException e) {

mAddress.setText(" ");

e.printStackTrace();

}

}

}

@Override

public void onConnected(Bundle bundle) {

checkPermissions();

}

@Override

public void onConnectionSuspended(int i) {

//Do whatever you need

//You can display a message here

Toast.makeText(this, "Connection Failed", Toast.LENGTH\_SHORT).show();

}

@Override

public void onConnectionFailed(ConnectionResult connectionResult) {

//You can display a message here

Toast.makeText(this, "Connection Failed", Toast.LENGTH\_SHORT).show();

}

private void getMyLocation() {

if (googleApiClient != null) {

if (googleApiClient.isConnected()) {

int permissionLocation = ContextCompat.checkSelfPermission(NewPostActivity.this,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

mylocation = LocationServices.FusedLocationApi.getLastLocation(googleApiClient);

LocationRequest locationRequest = new LocationRequest();

locationRequest.setInterval(3000);

locationRequest.setFastestInterval(3000);

locationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()

.addLocationRequest(locationRequest);

builder.setAlwaysShow(true);

LocationServices.getFusedLocationProviderClient(this).requestLocationUpdates(locationRequest, new LocationCallback() {

@Override

public void onLocationResult(LocationResult locationResult) {

// do work here

onLocationChanged(locationResult.getLastLocation());

}

}, Looper.myLooper());

PendingResult<LocationSettingsResult> result =

LocationServices.SettingsApi

.checkLocationSettings(googleApiClient, builder.build());

result.setResultCallback(new ResultCallback<LocationSettingsResult>() {

@Override

public void onResult(LocationSettingsResult result) {

final Status status = result.getStatus();

switch (status.getStatusCode()) {

case LocationSettingsStatusCodes.SUCCESS:

// All location settings are satisfied.

// You can initialize location requests here.

int permissionLocation = ContextCompat

.checkSelfPermission(NewPostActivity.this,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

mylocation = LocationServices.FusedLocationApi.getLastLocation(googleApiClient);

}

break;

case LocationSettingsStatusCodes.RESOLUTION\_REQUIRED:

// Location settings are not satisfied.

// But could be fixed by showing the user a dialog.

try {

// Show the dialog by calling startResolutionForResult(),

// and check the result in onActivityResult().

// Ask to turn on GPS automatically

status.startResolutionForResult(NewPostActivity.this,

REQUEST\_CHECK\_SETTINGS\_GPS);

} catch (IntentSender.SendIntentException e) {

// Ignore the error.

}

break;

case LocationSettingsStatusCodes.SETTINGS\_CHANGE\_UNAVAILABLE:

// Location settings are not satisfied.

// However, we have no way

// to fix the

// settings so we won't show the dialog.

// finish();

break;

}

}

});

}

}

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

switch (requestCode) {

case REQUEST\_CHECK\_SETTINGS\_GPS:

switch (resultCode) {

case Activity.RESULT\_OK:

getMyLocation();

break;

case Activity.RESULT\_CANCELED:

finish();

break;

}

break;

}

if (requestCode == CropImage.CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE) {

CropImage.ActivityResult result = CropImage.getActivityResult(data);

if (resultCode == RESULT\_OK) {

postImageUri = result.getUri();

mSelectImage.setImageURI(postImageUri);

} else if (resultCode == CropImage.CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE) {

Exception error = result.getError();

}

}

}

private void checkPermissions() {

int permissionLocation = ContextCompat.checkSelfPermission(NewPostActivity.this,

android.Manifest.permission.ACCESS\_FINE\_LOCATION);

List<String> listPermissionsNeeded = new ArrayList<>();

if (permissionLocation != PackageManager.PERMISSION\_GRANTED) {

listPermissionsNeeded.add(android.Manifest.permission.ACCESS\_FINE\_LOCATION);

if (!listPermissionsNeeded.isEmpty()) {

ActivityCompat.requestPermissions(this,

listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), REQUEST\_ID\_MULTIPLE\_PERMISSIONS);

}

} else {

getMyLocation();

}

}

@Override

public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

int permissionLocation = ContextCompat.checkSelfPermission(NewPostActivity.this,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

getMyLocation();

}

}

@Override

public void onStop() {

super.onStop();

if (googleApiClient.isConnected()) {

googleApiClient.disconnect();

}

}

@Override

public void onPause() {

super.onPause();

googleApiClient.stopAutoManage(this);

googleApiClient.disconnect();

}

}

* + 1. **HomeFragment**

package com.aris.crowdreporting.Fragment;

import android.app.ProgressDialog;

import android.content.Context;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.support.v4.view.MenuItemCompat;

import android.support.v4.widget.SwipeRefreshLayout;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.support.v7.widget.SearchView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.MainActivity;

import com.aris.crowdreporting.Activities.NewPostActivity;

import com.aris.crowdreporting.Adapters.ChatsUsersAdapter;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.Adapters.BlogRecyclerAdapter;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.R;

import com.aris.crowdreporting.HelperClasses.User;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.HashMap;

import java.util.List;

import javax.annotation.Nullable;

import dmax.dialog.SpotsDialog;

import static android.support.constraint.Constraints.TAG;

public class HomeFragment extends Fragment implements SearchView.OnQueryTextListener{

SwipeRefreshLayout pullToRefresh;

private RecyclerView blog\_list\_view;

private List<Blog> blog\_list;

private List<User> user\_list;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private BlogRecyclerAdapter blogRecyclerAdapter;

private DocumentSnapshot lastVisible;

private Boolean isFirstPageFirstLoad = true;

private SpotsDialog dialog;

private String user\_id;

private Status status;

private Context mContext;

public HomeFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, final ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_home, container, false);

//nampilin tambahan menu di actionbar pada fragment

setHasOptionsMenu(true);

mContext = getContext();

blog\_list = new ArrayList<>();

user\_list = new ArrayList<>();

blog\_list\_view = view.findViewById(R.id.blog\_list\_view);

pullToRefresh = view.findViewById(R.id.pull);

pullToRefresh.setColorSchemeColors(Color.CYAN, Color.YELLOW, Color.MAGENTA);

firebaseAuth = FirebaseAuth.getInstance();

blogRecyclerAdapter = new BlogRecyclerAdapter(blog\_list);

blog\_list\_view.setLayoutManager(new LinearLayoutManager(container.getContext()));

blog\_list\_view.setAdapter(blogRecyclerAdapter);

blog\_list\_view.setHasFixedSize(true);

dialog = new SpotsDialog(mContext, "Loading..");

dialog.show();

if(firebaseAuth.getCurrentUser() != null) {

firebaseFirestore = FirebaseFirestore.getInstance();

user\_id = firebaseAuth.getCurrentUser().getUid();

blog\_list\_view.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

Boolean reachedBottom = !recyclerView.canScrollVertically(1);

if(reachedBottom){

// loadMorePost();

}

}

});

firstQ();

pullToRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

@Override

public void onRefresh() {

blogRecyclerAdapter.notifyDataSetChanged();

pullToRefresh.setRefreshing(false);

}

});

}

// Inflate the layout for this fragment

return view;

}

private void firstQ() {

CollectionReference collectionReference = firebaseFirestore.collection("Posts");

Query firstQuery = collectionReference.whereEqualTo("reports", "false");

firstQuery.addSnapshotListener(getActivity(), new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

if (isFirstPageFirstLoad) {

lastVisible = documentSnapshots.getDocuments().get(documentSnapshots.size() - 1);

blog\_list.clear();

}

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String blogPostId = doc.getDocument().getId();

Blog blogPost = doc.getDocument().toObject(Blog.class).withId(blogPostId);

if (isFirstPageFirstLoad) {

if (blogPost.getDesc().toLowerCase().contains("bekasi") || blogPost.getDesc().toLowerCase().contains("bks") || blogPost.getDesc().toLowerCase().contains("bksi")

|| blogPost.getAddress().toLowerCase().contains("bekasi") || blogPost.getAddress().toLowerCase().contains("bks")){

blog\_list.add(blogPost);

Collections.sort(blog\_list, new Comparator<Blog>() {

@Override

public int compare(Blog o1, Blog o2) {

return o2.getTimestamp().compareTo(o1.getTimestamp());

}

});

blogRecyclerAdapter.notifyItemInserted(blog\_list.size());

blogRecyclerAdapter.notifyDataSetChanged();

}

} else {

if (blogPost.getDesc().toLowerCase().contains("bekasi") || blogPost.getDesc().toLowerCase().contains("bks") || blogPost.getDesc().toLowerCase().contains("bksi")

|| blogPost.getAddress().toLowerCase().contains("bekasi") || blogPost.getAddress().toLowerCase().contains("bks")) {

blog\_list.add(0, blogPost);

Collections.sort(blog\_list, new Comparator<Blog>() {

@Override

public int compare(Blog o1, Blog o2) {

return o2.getTimestamp().compareTo(o1.getTimestamp());

}

});

blogRecyclerAdapter.notifyItemInserted(0);

blogRecyclerAdapter.notifyDataSetChanged();

}

}

dialog.dismiss();

}

}

isFirstPageFirstLoad = false;

}

}

});

}

@Override

public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {

inflater.inflate(R.menu.action\_bar\_search, menu);

MenuItem menuItem = menu.findItem(R.id.menu\_search);

SearchView searchView = (SearchView)menuItem.getActionView();

searchView.setOnQueryTextListener(this);

menuItem.setOnActionExpandListener(new MenuItem.OnActionExpandListener() {

@Override

public boolean onMenuItemActionExpand(MenuItem menuItem) {

return true;

}

@Override

public boolean onMenuItemActionCollapse(MenuItem menuItem) {

blog\_list.clear();

firstQ();

return true;

}

});

super.onCreateOptionsMenu(menu, inflater);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.menu\_search) {

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public boolean onQueryTextSubmit(String s) {

FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();

Query query = firebaseFirestore.collection("Posts").orderBy("timestamp", Query.Direction.DESCENDING);

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

blog\_list.clear();

for (DocumentChange doc : queryDocumentSnapshots.getDocumentChanges()){

String blogPostId = doc.getDocument().getId();

Blog blogPost = doc.getDocument().toObject(Blog.class).withId(blogPostId);

if (blogPost.getDesc().toLowerCase().contains(s)){

blog\_list.add(blogPost);

}

}

blogRecyclerAdapter = new BlogRecyclerAdapter(blog\_list);

blog\_list\_view.setAdapter(blogRecyclerAdapter);

}

});

return false;

}

@Override

public boolean onQueryTextChange(String s) {

return false;

}

@Override

public void onResume() {

super.onResume();

status = new Status("online");

}

@Override

public void onPause() {

super.onPause();

status = new Status("offline");

}

@Override

public void onStop() {

super.onStop();

status = new Status("offline");

}

}

* + 1. **BlogRecyclerAdapter**

package com.aris.crowdreporting.Adapters;

import android.app.Activity;

import android.app.Dialog;

import android.content.Context;

import android.content.Intent;

import android.graphics.Color;

import android.graphics.drawable.ColorDrawable;

import android.support.annotation.NonNull;

import android.support.v7.widget.RecyclerView;

import android.text.format.DateFormat;

import android.util.Log;

import android.view.Gravity;

import android.view.LayoutInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.PopupMenu;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.DetailProfileActivity;

import com.aris.crowdreporting.Activities.DetailActivity;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.HelperUtils.TimeAgo;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

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.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.firestore.SetOptions;

import com.mikhaellopez.circularimageview.CircularImageView;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.Nullable;

import dmax.dialog.SpotsDialog;

import static android.support.constraint.Constraints.TAG;

public class BlogRecyclerAdapter extends RecyclerView.Adapter<BlogRecyclerAdapter.ViewHolder> {

public List<Blog> blog\_list;

public Context context;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

public BlogRecyclerAdapter(List<Blog> blog\_list){

this.blog\_list = blog\_list;

}

@Override

public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.blog\_row, parent, false);

context = parent.getContext();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(final ViewHolder holder, int position) {

holder.setIsRecyclable(false);

final String blogPostId = blog\_list.get(position).BlogPostId;

final String blogUserId = blog\_list.get(position).getUser\_id();

final String currentUserId = firebaseAuth.getCurrentUser().getUid();

String desc\_data = blog\_list.get(position).getDesc();

holder.setDescText(desc\_data);

String address\_data = blog\_list.get(position).getAddress();

holder.setAddress(address\_data);

String image\_url = blog\_list.get(position).getImage\_uri();

String thumbUri = blog\_list.get(position).getImage\_thumb();

holder.setBlogImage(image\_url, thumbUri);

String user\_id = blog\_list.get(position).getUser\_id();

//User Data will be retrieved here...

firebaseFirestore.collection("Users").document(user\_id)

.addSnapshotListener((Activity) context,new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(DocumentSnapshot documentSnapshot, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (documentSnapshot.exists()) {

String userImageUrl = documentSnapshot.get("image").toString();

String userNameData = documentSnapshot.get("name").toString();

if (userNameData.isEmpty()) {

userNameData = "No name";

}

holder.blogUserName.setText(userNameData);

// GlideLoadImage.loadSmallImage(context,holder.blogUserImage,userImageUrl,userImageUrl);

RequestOptions placeholderOption = new RequestOptions();

placeholderOption.placeholder(R.drawable.profile\_placeholder);

Glide.with(context).applyDefaultRequestOptions(placeholderOption).load(userImageUrl).into(holder.blogUserImage);

} else {

Log.w("USER\_DETAIL", "Empty DOC");

}

}

});

long millisecond = blog\_list.get(position).getTimestamp().getTime();

String dateString = DateFormat.format("MM/dd/yyyy", new Date(millisecond)).toString();

String timeAgo = TimeAgo.getTimeAgo(millisecond);

// Toast.makeText(context, ""+timeAgo, Toast.LENGTH\_SHORT).show();

holder.setTime(timeAgo);

//Get Likes Count

firebaseFirestore.collection("Posts/" + blogPostId + "/Likes").addSnapshotListener((Activity) context, new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if(!documentSnapshots.isEmpty()){

int count = documentSnapshots.size();

holder.updateLikesCount(count);

} else {

holder.updateLikesCount(0);

}

}

});

//Get Likes

firebaseFirestore.collection("Posts/" + blogPostId + "/Likes").document(currentUserId).addSnapshotListener((Activity) context, new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(DocumentSnapshot documentSnapshot, FirebaseFirestoreException e) {

if(documentSnapshot.exists()){

holder.blogLikeBtn.setImageDrawable(context.getDrawable(R.mipmap.dilike));

} else {

holder.blogLikeBtn.setImageDrawable(context.getDrawable(R.mipmap.like\_gray));

}

}

});

//Likes Feature

holder.blogLikeBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

final Animation likeAnim = AnimationUtils.loadAnimation(context,R.anim.like\_animation);

holder.blogLikeBtn.startAnimation(likeAnim);

firebaseFirestore.collection("Posts/" + blogPostId + "/Likes").document(currentUserId).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if(task.isSuccessful()){

if (!task.getResult().exists()) {

Map<String, Object> likesMap = new HashMap<>();

likesMap.put("timestamp", FieldValue.serverTimestamp());

firebaseFirestore.collection("Posts/" + blogPostId + "/Likes").document(currentUserId).set(likesMap);

} else {

firebaseFirestore.collection("Posts/" + blogPostId + "/Likes").document(currentUserId).delete();

//coba

}

} else {

Log.i("LikeError", task.getException().getMessage());

Toast.makeText(context, "Please check your connection", Toast.LENGTH\_SHORT).show();

}

}

});

}

});

holder.blogCommentBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context, DetailActivity.class);

intent.putExtra("blog\_id",blogPostId);

intent.putExtra("user\_id",blogUserId);

intent.putExtra("imurl",image\_url);

intent.putExtra("desc",desc\_data);

intent.putExtra("time\_post",timeAgo);

// intent.putExtra("address",address\_data);

context.startActivity(intent);

}

});

//Get Comments Count

firebaseFirestore.collection("Posts/" + blogPostId + "/Comments").addSnapshotListener((Activity) context, new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if(!documentSnapshots.isEmpty()){

int count = documentSnapshots.size();

holder.updateCommentsCount(count);

} else {

holder.updateCommentsCount(0);

}

}

});

//POPUP

holder.popUpHome.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

PopupMenu popupMenu = new PopupMenu(context, holder.popUpHome, Gravity.CENTER, 0, R.style.PopupMenuMoreCentralized);

popupMenu.setGravity(Gravity.CENTER);

popupMenu.getMenuInflater().inflate(R.menu.popup\_menu\_home, popupMenu.getMenu());

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {

@Override

public boolean onMenuItemClick(MenuItem item) {

switch (item.getItemId()) {

case R.id.report:

holder.spotsDialog = new SpotsDialog(context, "Please Wait.");

holder.spotsDialog.show();

//Lapor Feature

CollectionReference collectionReference = firebaseFirestore.collection("Posts");

collectionReference.document(blogPostId + "/PostHoax/" + currentUserId).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.getResult().exists()){

holder.spotsDialog.dismiss();

Toast.makeText(context, "You has been reported this post!", Toast.LENGTH\_SHORT).show();

} else {

Map<String, Object> reportsMap = new HashMap<>();

reportsMap.put("reports", "true");

collectionReference.document(blogPostId).collection("PostHoax").document(currentUserId)

.set(reportsMap).addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

holder.spotsDialog.dismiss();

Toast.makeText(context, "Thanks for reported this post..", Toast.LENGTH\_SHORT).show();

holder.hideItem.setVisibility(View.VISIBLE);

blog\_list.remove(position);

Query query = collectionReference.document(blogPostId).collection("PostHoax");

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

if (queryDocumentSnapshots.size() > 3){

firebaseFirestore.collection("Posts").document(blogPostId).set(reportsMap, SetOptions.merge());

}

}

});

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

holder.spotsDialog.dismiss();

Toast.makeText(context, ""+e.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

}

}

});

return true;

default:

return false;

}

}

});

popupMenu.show();

}

});

holder.blogImageView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context, DetailActivity.class);

intent.putExtra("blog\_id",blogPostId);

intent.putExtra("user\_id",blogUserId);

intent.putExtra("imurl",image\_url);

intent.putExtra("desc",desc\_data);

intent.putExtra("time\_post",timeAgo);

context.startActivity(intent);

}

});

holder.blogUserImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

holder.showDataPopUpProfile(blog\_list.get(position).getUser\_id());

holder.mDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));

holder.mDialog.show();

}

});

holder.blogUserName.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

holder.showDataPopUpProfile(blog\_list.get(position).getUser\_id());

holder.mDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));

holder.mDialog.show();

}

});

}

@Override

public int getItemCount() {

return blog\_list.size();

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View mView;

private TextView descView;

private ImageView blogImageView;

private CircularImageView popUpUserImage;

private TextView blogDate, blogAddress;

private TextView blogUserName, popUpUserName, popUpUserEmail, popUpUserPhone;

private CircularImageView blogUserImage;

private ImageView blogLikeBtn, popUpHome;

private TextView blogLikeCount;

private LinearLayout hideItem;

private ImageView blogCommentBtn;

private Button deleteBtn, viewProfileBtn;

private Dialog mDialog;

private SpotsDialog spotsDialog;

public ViewHolder(View itemView) {

super(itemView);

mView = itemView;

hideItem = mView.findViewById(R.id.hideitem);

blogLikeBtn = mView.findViewById(R.id.btn\_like);

blogCommentBtn = mView.findViewById(R.id.btn\_comment);

// deleteBtn = mView.findViewById(R.id.btn\_delete);

popUpHome = mView.findViewById(R.id.popup\_home);

blogUserName = mView.findViewById(R.id.blog\_username);

blogUserImage = mView.findViewById(R.id.blog\_user\_image);

mDialog = new Dialog(context);

mDialog.setContentView(R.layout.popup\_detail\_profile);

}

public void setDescText(String descText){

descView = mView.findViewById(R.id.blog\_desc);

String upperString = descText.substring(0,1).toUpperCase() + descText.substring(1);

descView.setText(upperString);

}

public void setBlogImage(String downloadUri, String thumbUri){

blogImageView = mView.findViewById(R.id.blog\_image);

RequestOptions requestOptions = new RequestOptions();

requestOptions.placeholder(R.drawable.image\_placeholder);

Glide.with(context).applyDefaultRequestOptions(requestOptions).load(thumbUri).thumbnail(

Glide.with(context).load(thumbUri)

).into(blogImageView);

}

public void setTime(String date) {

blogDate = mView.findViewById(R.id.blog\_date);

blogDate.setText(date);

}

public void updateLikesCount(int count){

blogLikeCount = mView.findViewById(R.id.like\_count);

blogLikeCount.setText(count +"");

}

public void updateCommentsCount(int count){

blogLikeCount = mView.findViewById(R.id.comment\_count);

blogLikeCount.setText(count +"");

}

public void showDataPopUpProfile(String popupName){

popUpUserName = mDialog.findViewById(R.id.popup\_username);

popUpUserEmail = mDialog.findViewById(R.id.popup\_user\_email);

popUpUserPhone = mDialog.findViewById(R.id.popup\_user\_phone);

popUpUserImage = mDialog.findViewById(R.id.popup\_user\_image);

viewProfileBtn = mDialog.findViewById(R.id.btnviewprofile);

firebaseFirestore.collection("Users").document(popupName).addSnapshotListener(new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(@Nullable DocumentSnapshot documentSnapshot, @Nullable FirebaseFirestoreException e) {

if (documentSnapshot.exists()){

String popupusername = documentSnapshot.get("name").toString();

String popupuseremail = documentSnapshot.get("email").toString();

String popupuserphone = documentSnapshot.get("phone").toString();

String popupuserimg = documentSnapshot.get("image").toString();

popUpUserName.setText(popupusername);

popUpUserEmail.setText(popupuseremail);

popUpUserPhone.setText(popupuserphone);

Glide.with(context).load(popupuserimg).into(popUpUserImage);

viewProfileBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent seeProfile = new Intent(context, DetailProfileActivity.class);

seeProfile.putExtra("user\_id", popupName);

context.startActivity(seeProfile);

}

});

}

}

});

}

public void setAddress(String address\_data) {

blogAddress = mView.findViewById(R.id.blog\_address);

blogAddress.setText(address\_data);

}

}

}

* + 1. **DetailActivity**

package com.aris.crowdreporting.Activities;

import android.app.AlertDialog;

import android.support.annotation.NonNull;

import android.support.design.widget.AppBarLayout;

import android.support.design.widget.CollapsingToolbarLayout;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.text.TextUtils;

import android.view.MenuItem;

import android.view.View;

import android.view.WindowManager;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import android.support.v7.widget.Toolbar;

import com.aris.crowdreporting.Adapters.CommentsRecyclerAdapter;

import com.aris.crowdreporting.HelperClasses.Comments;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.flaviofaria.kenburnsview.KenBurnsView;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.gson.annotations.Until;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import dmax.dialog.SpotsDialog;

public class DetailActivity extends AppCompatActivity {

private static final String TAG = "DetailsActivity";

//private android.support.v7.widget.Toolbar mToolbar;

private KenBurnsView blogImage;

private AppBarLayout appBarLayout;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private Toolbar mToolbar;

private CollapsingToolbarLayout collapsingToolbarLayout;

private String blogTime;

private String mTitle = "";

private Toolbar commentToolbar;

private EditText comment\_field;

private ImageView comment\_post\_btn, thumbUserPost;

private TextView descPost, userPost, timeAgo;

private RecyclerView comment\_list;

private CommentsRecyclerAdapter commentsRecyclerAdapter;

private List<Comments> commentsList;

private String currentUserId;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_detail);

this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN);

mToolbar = findViewById(R.id.detail\_toolbar);

setSupportActionBar(mToolbar);

final android.support.v7.app.ActionBar actionBar = getSupportActionBar();

actionBar.setTitle("");

collapsingToolbarLayout = findViewById(R.id.collapsingToolbar);

collapsingToolbarLayout.setElevation(10.0f);

collapsingToolbarLayout.setExpandedTitleColor(ContextCompat.getColor(DetailActivity.this,R.color.putih));

appBarLayout = findViewById(R.id.appbar);

appBarLayout.addOnOffsetChangedListener(new AppBarLayout.OnOffsetChangedListener() {

@Override

public void onOffsetChanged(AppBarLayout appBarLayout, int verticalOffset) {

if (Math.abs(verticalOffset) == appBarLayout.getTotalScrollRange()) {

// Collapsed

actionBar.setDisplayHomeAsUpEnabled(true);

actionBar.setTitle("Detail");

} else if (verticalOffset == 0) {

// Expanded

actionBar.setDisplayHomeAsUpEnabled(false);

actionBar.setTitle("");

collapsingToolbarLayout.setTitle("");

} else {

actionBar.setDisplayHomeAsUpEnabled(false);

actionBar.setTitle("");

collapsingToolbarLayout.setTitle("");

// Somewhere in between

}

}

});

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

currentUserId = firebaseAuth.getCurrentUser().getUid();

//GET EXTRA FROM BLOGRECVIEW

final String blogPostId = getIntent().getStringExtra("blog\_id");

final String blogUserId = getIntent().getStringExtra("user\_id");

final String imgUrl = getIntent().getStringExtra("imurl");

final String time = getIntent().getStringExtra("time\_post");

final String desc = getIntent().getStringExtra("desc");

final String address\_data = getIntent().getStringExtra("address");

blogImage = findViewById(R.id.detail\_image);

userPost = findViewById(R.id.userpost);

descPost = findViewById(R.id.descpost);

thumbUserPost = findViewById(R.id.thumb\_userpost);

timeAgo = findViewById(R.id.time\_post);

comment\_field = findViewById(R.id.isi\_komen);

comment\_post\_btn = findViewById(R.id.kirim\_komen);

comment\_list = findViewById(R.id.list\_comment);

//RecyclerView Firebase List

commentsList = new ArrayList<>();

commentsRecyclerAdapter = new CommentsRecyclerAdapter(commentsList);

comment\_list.setHasFixedSize(true);

comment\_list.setLayoutManager(new LinearLayoutManager(this));

comment\_list.setAdapter(commentsRecyclerAdapter);

//SHOW COMMENTS

firebaseFirestore.collection("Posts/" + blogPostId + "/Comments")

.orderBy("timestamp", Query.Direction.DESCENDING)

.addSnapshotListener(DetailActivity.this, new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (!documentSnapshots.isEmpty()) {

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

Comments comments = doc.getDocument().toObject(Comments.class);

commentsList.add(comments);

commentsRecyclerAdapter.notifyItemChanged(0);

commentsRecyclerAdapter.notifyDataSetChanged();

}

}

}

}

});

//Tekan Btn Comment

comment\_post\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String comment\_message = comment\_field.getText().toString();

if (TextUtils.isEmpty(comment\_message)){

Toast.makeText(DetailActivity.this, "Give some message.", Toast.LENGTH\_SHORT).show();

} else {

AlertDialog dialog = new SpotsDialog(DetailActivity.this);

dialog.show();

Map<String, Object> commentsMap = new HashMap<>();

commentsMap.put("message", comment\_message);

commentsMap.put("user\_id", currentUserId);

commentsMap.put("timestamp", FieldValue.serverTimestamp());

firebaseFirestore.collection("Posts/" + blogPostId + "/Comments").add(commentsMap).addOnCompleteListener(new OnCompleteListener<DocumentReference>() {

@Override

public void onComplete(@NonNull Task<DocumentReference> task) {

if(!task.isSuccessful()){

dialog.dismiss();

Toast.makeText(DetailActivity.this, "Error Posting Comment : " + task.getException().getMessage(), Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(DetailActivity.this, "Comment posted", Toast.LENGTH\_SHORT).show();

comment\_field.setText("");

dialog.dismiss();

}

}

});

}

}

});

//SHOW DETAIL POS

//set Desc Post

descPost.setText(desc);

//set time post

timeAgo.setText(time);

//Set Image Post

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.rectangle);

Glide.with(getApplicationContext()).applyDefaultRequestOptions(placeholderReq).load(imgUrl).into(blogImage);

//Set Image User Post

firebaseFirestore.collection("Users").document(blogUserId).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if(task.isSuccessful()){

String userName = task.getResult().getString("name");

String userImage = task.getResult().getString("image");

Glide.with(getApplicationContext()).load(userImage).into(thumbUserPost);

userPost.setText(userName);

} else {

//Firebase Exception

}

}

});

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == android.R.id.home) {

onBackPressed();

return true;

}

return super.onOptionsItemSelected(item);

}

private Status status;

@Override

public void onResume() {

super.onResume();

status = new Status("online");

}

@Override

public void onPause() {

super.onPause();

status = new Status("offline");

}

@Override

public void onStop() {

super.onStop();

status = new Status("offline");

}

}

* + 1. **CommentsRecyclerAdapter**

package com.aris.crowdreporting.Adapters;

import android.content.Context;

import android.support.annotation.NonNull;

import android.support.v7.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import com.aris.crowdreporting.HelperClasses.Comments;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;

public class CommentsRecyclerAdapter extends RecyclerView.Adapter<CommentsRecyclerAdapter.ViewHolder> {

public List<Comments> commentsList;

public Context context;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

public CommentsRecyclerAdapter(List<Comments> commentsList){

this.commentsList = commentsList;

}

@Override

public CommentsRecyclerAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.comment\_list\_item, parent, false);

context = parent.getContext();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

return new CommentsRecyclerAdapter.ViewHolder(view);

}

@Override

public void onBindViewHolder(final CommentsRecyclerAdapter.ViewHolder holder, int position) {

holder.setIsRecyclable(false);

String commentMessage = commentsList.get(position).getMessage();

holder.setComment\_message(commentMessage);

String user\_id = commentsList.get(position).getUser\_id();

firebaseFirestore.collection("Users").document(user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if(task.isSuccessful()){

String userName = task.getResult().getString("name");

String userImage = task.getResult().getString("image");

holder.setUserData(userName, userImage);

} else {

//Firebase Exception

}

}

});

}

@Override

public int getItemCount() {

if(commentsList != null) {

return commentsList.size();

} else {

return 0;

}

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View mView;

private TextView comment\_message;

private CircleImageView blogUserImage;

private TextView blogUserName;

public ViewHolder(View itemView) {

super(itemView);

mView = itemView;

}

public void setComment\_message(String message){

comment\_message = mView.findViewById(R.id.comment\_message);

comment\_message.setText(message);

}

public void setUserData(String name, String image){

blogUserImage = mView.findViewById(R.id.comment\_image);

blogUserName = mView.findViewById(R.id.comment\_username);

blogUserName.setText(name);

RequestOptions placeholderOption = new RequestOptions();

placeholderOption.placeholder(R.drawable.profile\_placeholder);

Glide.with(context).applyDefaultRequestOptions(placeholderOption).load(image).into(blogUserImage);

}

public void setImageDescPost(){

}

}

}

* + 1. **DetailProfileActivity**

package com.aris.crowdreporting.Activities;

import android.graphics.PorterDuff;

import android.graphics.drawable.AnimationDrawable;

import android.net.Uri;

import android.support.annotation.NonNull;

import android.support.design.widget.AppBarLayout;

import android.support.design.widget.CollapsingToolbarLayout;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.GridLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.support.v7.widget.Toolbar;

import android.util.Log;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Adapters.MyPhotoRecyclerAdapter;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.mikhaellopez.circularimageview.CircularImageView;

import java.util.ArrayList;

import java.util.List;

public class DetailProfileActivity extends AppCompatActivity {

private final static String TAG = "PROFILE\_ACTIVITY";

private Toolbar profileToolbar;

private CollapsingToolbarLayout collapsingToolbarLayout;

private AppBarLayout appBarLayout;

private CircularImageView profileImageV;

private TextView usernameV, emailV, phoneV, emptyTxt;

private ImageView emptyImg;

private Uri mainImageUri;

private String user\_id;

private String emailuser\_id;

private StorageReference storageReference;

private FirebaseAuth firebaseAuth;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth.AuthStateListener mAuthListener;

private GoogleApiClient mGoogleApiClient;

private RecyclerView profileBlogListView;

private List<Blog> blog\_list;

private MyPhotoRecyclerAdapter profileBlogRecyclerAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_detail\_profile);

profileToolbar = findViewById(R.id.detail\_profile\_toolbar);

setSupportActionBar(profileToolbar);

final android.support.v7.app.ActionBar actionBar = getSupportActionBar();

actionBar.setTitle("");

collapsingToolbarLayout = findViewById(R.id.collapsingToolbar);

collapsingToolbarLayout.setElevation(10.0f);

collapsingToolbarLayout.setExpandedTitleColor(ContextCompat.getColor(DetailProfileActivity.this,R.color.putih));

user\_id = getIntent().getStringExtra("user\_id");

// Toast.makeText(this, ""+user\_id, Toast.LENGTH\_SHORT).show();

// declare and initialize FrameLayout

View frameLayout = (View)findViewById(R.id.lini);

// initializing animation drawable by getting background from frame layout

AnimationDrawable animationDrawable = (AnimationDrawable) frameLayout.getBackground();

// setting enter fade animation duration to 5 seconds

animationDrawable.setEnterFadeDuration(2500);

// setting exit fade animation duration to 2 seconds

animationDrawable.setExitFadeDuration(1000);

animationDrawable.start();

animationDrawable.run();

Button popup = findViewById(R.id.options);

firebaseAuth = FirebaseAuth.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

storageReference = FirebaseStorage.getInstance().getReference();

emailuser\_id = firebaseAuth.getCurrentUser().getEmail();

profileImageV = (CircularImageView)findViewById(R.id.profile\_image\_view);

usernameV = (TextView)findViewById(R.id.username\_view);

emailV = (TextView)findViewById(R.id.email\_view);

phoneV = (TextView)findViewById(R.id.phone\_view);

emptyTxt = findViewById(R.id.empty);

emptyImg = findViewById(R.id.emptyimg);

//USERNYA SIAPA

firebaseFirestore.collection("Users").document(user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()){

if (task.getResult().exists()){

String name = task.getResult().getString("name");

String image = task.getResult().getString("image");

String emailku = task.getResult().getString("email");

String phoneku = task.getResult().getString("phone");

mainImageUri = Uri.parse(image);

usernameV.setText(""+name);

emailV.setText(""+emailku);

phoneV.setText(""+phoneku);

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.defaultimage);

Glide.with(DetailProfileActivity.this).setDefaultRequestOptions(placeholderReq).load(image).into(profileImageV);

appBarLayout = findViewById(R.id.appbar);

appBarLayout.addOnOffsetChangedListener(new AppBarLayout.OnOffsetChangedListener() {

@Override

public void onOffsetChanged(AppBarLayout appBarLayout, int verticalOffset) {

if (Math.abs(verticalOffset) == appBarLayout.getTotalScrollRange()) {

// Collapsed

actionBar.setDisplayHomeAsUpEnabled(true);

actionBar.setTitle(name);

} else if (verticalOffset == 0) {

// Expanded

actionBar.setDisplayHomeAsUpEnabled(false);

actionBar.setTitle("");

collapsingToolbarLayout.setTitle("");

} else {

actionBar.setDisplayHomeAsUpEnabled(false);

actionBar.setTitle("");

collapsingToolbarLayout.setTitle("");

// Somewhere in between

}

}

});

}

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(DetailProfileActivity.this, ""+ errMSg, Toast.LENGTH\_SHORT).show();

}

}

});

//PHOTO POST USERNYA

///Recyclerview

blog\_list = new ArrayList<>();

profileBlogListView = findViewById(R.id.rv\_images);

profileBlogRecyclerAdapter = new MyPhotoRecyclerAdapter(DetailProfileActivity.this,blog\_list);

profileBlogListView.setLayoutManager(new GridLayoutManager(DetailProfileActivity.this,3));

profileBlogListView.setAdapter(profileBlogRecyclerAdapter);

Query firstQuery = firebaseFirestore.collection("Posts")

.orderBy("timestamp", Query.Direction.DESCENDING)

.whereEqualTo("user\_id", user\_id);

firstQuery.addSnapshotListener(this, new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String blogPostId = doc.getDocument().getId();

Blog blogPost = doc.getDocument().toObject(Blog.class).withId(blogPostId);

blog\_list.add(blogPost);

profileBlogRecyclerAdapter.notifyDataSetChanged();

}

}

} else {

profileBlogListView.setVisibility(View.GONE);

emptyTxt.setVisibility(View.VISIBLE);

emptyImg.setVisibility(View.VISIBLE);

}

}

});

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == android.R.id.home) {

onBackPressed();

return true;

}

return super.onOptionsItemSelected(item);

}

private Status status;

@Override

public void onResume() {

super.onResume();

status = new Status("online");

}

@Override

public void onPause() {

super.onPause();

status = new Status("offline");

}

@Override

public void onStop() {

super.onStop();

status = new Status("offline");

}

}

* + 1. **NearFragment**

package com.aris.crowdreporting.Fragment;

import android.Manifest;

import android.app.Activity;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.IntentSender;

import android.content.SharedPreferences;

import android.content.pm.PackageManager;

import android.graphics.Color;

import android.location.Location;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.Looper;

import android.support.v4.app.ActivityCompat;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentActivity;

import android.support.v4.content.ContextCompat;

import android.support.v4.widget.SwipeRefreshLayout;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.DetailActivity;

import com.aris.crowdreporting.Adapters.NearRecyclerAdapter;

import com.aris.crowdreporting.CheckLoc.CheckLoc;

import com.aris.crowdreporting.HelperClasses.Near;

import com.aris.crowdreporting.R;

import com.aris.crowdreporting.HelperUtils.SortPlaces;

import com.aris.crowdreporting.HelperClasses.User;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.common.api.PendingResult;

import com.google.android.gms.common.api.ResultCallback;

import com.google.android.gms.common.api.Status;

import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationCallback;

import com.google.android.gms.location.LocationListener;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationResult;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.location.LocationSettingsRequest;

import com.google.android.gms.location.LocationSettingsResult;

import com.google.android.gms.location.LocationSettingsStatusCodes;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.type.LatLng;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Date;

import java.util.List;

import dmax.dialog.SpotsDialog;

public class NearFragment extends Fragment implements GoogleApiClient.ConnectionCallbacks,

GoogleApiClient.OnConnectionFailedListener,

LocationListener {

private static final String TAG = "NearAct";

private RecyclerView near\_list\_view;

private TextView txtEmptyPost, txtGps;

private ImageView icnGps;

private List<Near> near\_list;

private List<User> user\_list;

private List<SortPlaces> sortPlaces;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private NearRecyclerAdapter nearRecyclerAdapter;

private DocumentSnapshot lastVisible;

private Boolean isFirstPageFirstLoad = true;

private Location mylocation;

private LocationManager locationManager;

private GoogleApiClient googleApiClient;

private LocationRequest mLocationRequest;

private final static int REQUEST\_CHECK\_SETTINGS\_GPS = 0x1;

private final static int REQUEST\_ID\_MULTIPLE\_PERMISSIONS = 0x2;

private SpotsDialog dialog;

private FusedLocationProviderClient client;

LatLng latLng;

private Context mContex;

private Date cDate;

private com.aris.crowdreporting.HelperUtils.Status status;

private SwipeRefreshLayout pullToRefresh;

public NearFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, final ViewGroup container,

Bundle savedInstanceState) {

View view = inflater.inflate(R.layout.fragment\_near, container, false);

setHasOptionsMenu(true);

mContex = getContext();

near\_list = new ArrayList<>();

user\_list = new ArrayList<>();

mLocationRequest = new LocationRequest();

near\_list\_view = view.findViewById(R.id.near\_list\_view);

txtEmptyPost = view.findViewById(R.id.empty\_post);

// txtGps = view.findViewById(R.id.aktifkan\_gps);

icnGps = view.findViewById(R.id.icongps);

pullToRefresh = view.findViewById(R.id.pull);

pullToRefresh.setColorSchemeColors(Color.CYAN, Color.YELLOW, Color.MAGENTA);

firebaseAuth = FirebaseAuth.getInstance();

client = LocationServices.getFusedLocationProviderClient(mContex);

dialog = new SpotsDialog(mContex, "Track Your Location, Please Wait!");

nearRecyclerAdapter = new NearRecyclerAdapter(near\_list);

near\_list\_view.setLayoutManager(new LinearLayoutManager(container.getContext()));

near\_list\_view.setAdapter(nearRecyclerAdapter);

near\_list\_view.setHasFixedSize(true);

if (firebaseAuth.getCurrentUser() != null) {

firebaseFirestore = FirebaseFirestore.getInstance();

near\_list\_view.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

}

});

nearQuery();

pullToRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

@Override

public void onRefresh() {

nearRecyclerAdapter.notifyDataSetChanged();

pullToRefresh.setRefreshing(false);

}

});

}

// Inflate the layout for this fragment

return view;

}

private void nearQuery() {

//FORMAT bln/tgl/thn untuk firestore emg gitu

Date date = new Date();

//HARI INI

long secs = date.getTime();

cDate = new Date(secs);

//7 hari yg lalu

// 7hari yang lalu

long DAY\_IN\_MS = 1000 \* 60 \* 60 \* 24;

long secs2 = date.getTime() - (7 \* DAY\_IN\_MS);

Date daysAgo = new Date(secs2);

Query nearQuery = firebaseFirestore.collection("Posts")

.whereEqualTo("reports", "false")

.whereLessThanOrEqualTo("timestamp", cDate)

.whereGreaterThanOrEqualTo("timestamp", daysAgo);

nearQuery.addSnapshotListener( new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

if (isFirstPageFirstLoad) {

lastVisible = documentSnapshots.getDocuments().get(documentSnapshots.size() - 1);

near\_list.clear();

}

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String nearPostId = doc.getDocument().getId();

Near nearPost = doc.getDocument().toObject(Near.class).withId(nearPostId);

if (ActivityCompat.checkSelfPermission((Activity) mContex, Manifest.permission.ACCESS\_FINE\_LOCATION)

!= PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission((Activity) mContex,

Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

} else {

Log.d("MYLOG ", "ERR");

}

client.getLastLocation().addOnSuccessListener((Activity) mContex, new OnSuccessListener<Location>() {

@Override

public void onSuccess(Location location) {

if (location != null) {

if (nearPost.getDesc().contains("bekasi")) {

txtEmptyPost.setVisibility(View.GONE);

icnGps.setVisibility(View.GONE);

near\_list.add(nearPost);

Double lat\_a = location.getLatitude();

Double lng\_a = location.getLongitude();

Collections.sort(near\_list, new SortPlaces(lat\_a, lng\_a));

nearRecyclerAdapter.notifyDataSetChanged();

}

}

}

});

}

}

} else {

//JIKA DALAM SEMINGGU TERAKHIR TIDAK ADA POSTS

near\_list\_view.setVisibility(View.GONE);

txtEmptyPost.setText("No Post In This Week");

txtEmptyPost.setVisibility(View.VISIBLE);

icnGps.setVisibility(View.GONE);

}

}

});

}

@Override

public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {

inflater.inflate(R.menu.action\_bar2, menu);

super.onCreateOptionsMenu(menu, inflater);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.locmu) {

// Intent intent = new Intent(getActivity(), CheckLoc.class);

// intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

// startActivity(intent);

setUpGClient();

return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onStart() {

super.onStart();

if (googleApiClient != null)

googleApiClient.connect();

}

@Override

public void onStop() {

super.onStop();

if (googleApiClient != null && googleApiClient.isConnected()) {

googleApiClient.stopAutoManage((FragmentActivity) getActivity());

googleApiClient.disconnect();

status = new com.aris.crowdreporting.HelperUtils.Status("offline");

}

}

private synchronized void setUpGClient() {

googleApiClient = new GoogleApiClient.Builder(mContex)

.enableAutoManage(getActivity(), 1, this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

googleApiClient.connect();

dialog.show();

}

@Override

public void onLocationChanged(Location location) {

mylocation = location;

if (mylocation != null) {

nearRecyclerAdapter.notifyDataSetChanged();

// Toast.makeText(mContex, "Get Location", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

near\_list.clear();

nearQuery();

}

}

@Override

public void onConnected(Bundle bundle) {

checkPermissions();

if (ActivityCompat.checkSelfPermission(mContex, Manifest.permission.ACCESS\_FINE\_LOCATION)

!= PackageManager.PERMISSION\_GRANTED

&& ActivityCompat.checkSelfPermission(mContex, Manifest.permission.ACCESS\_COARSE\_LOCATION)

!= PackageManager.PERMISSION\_GRANTED) {

return;

}

// startLocationUpdates();

}

@Override

public void onConnectionSuspended(int i) {

//Do whatever you need

//You can display a message here

}

@Override

public void onConnectionFailed(ConnectionResult connectionResult) {

//You can display a message here

}

private void getMyLocation() {

if (googleApiClient != null) {

if (googleApiClient.isConnected()) {

int permissionLocation = ContextCompat.checkSelfPermission(mContex,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

mylocation = LocationServices.FusedLocationApi.getLastLocation(googleApiClient);

LocationRequest locationRequest = new LocationRequest();

locationRequest.setInterval(10 \* 1000);

locationRequest.setFastestInterval(5 \* 1000);

locationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()

.addLocationRequest(locationRequest);

builder.setAlwaysShow(true);

// LocationServices.FusedLocationApi

// .requestLocationUpdates(googleApiClient, locationRequest, this);

LocationServices.getFusedLocationProviderClient((Activity)mContex).requestLocationUpdates(locationRequest, new LocationCallback() {

@Override

public void onLocationResult(LocationResult locationResult) {

// do work here

onLocationChanged(locationResult.getLastLocation());

}

}, Looper.myLooper());

PendingResult<LocationSettingsResult> result =

LocationServices.SettingsApi

.checkLocationSettings(googleApiClient, builder.build());

result.setResultCallback(new ResultCallback<LocationSettingsResult>() {

@Override

public void onResult(LocationSettingsResult result) {

final Status status = result.getStatus();

switch (status.getStatusCode()) {

case LocationSettingsStatusCodes.SUCCESS:

// All location settings are satisfied.

// You can initialize location requests here.

int permissionLocation = ContextCompat

.checkSelfPermission(mContex,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

mylocation = LocationServices.FusedLocationApi

.getLastLocation(googleApiClient);

}

break;

case LocationSettingsStatusCodes.RESOLUTION\_REQUIRED:

// Location settings are not satisfied.

// But could be fixed by showing the user a dialog.

try {

// Show the dialog by calling startResolutionForResult(),

// and check the result in onActivityResult().

// Ask to turn on GPS automatically

status.startResolutionForResult(getActivity(),

REQUEST\_CHECK\_SETTINGS\_GPS);

} catch (IntentSender.SendIntentException e) {

// Ignore the error.

}

break;

case LocationSettingsStatusCodes.SETTINGS\_CHANGE\_UNAVAILABLE:

// Location settings are not satisfied.

// However, we have no way

// to fix the

// settings so we won't show the dialog.

// finish();

break;

}

}

});

}

}

}

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

switch (requestCode) {

case REQUEST\_CHECK\_SETTINGS\_GPS:

switch (resultCode) {

case Activity.RESULT\_OK:

getMyLocation();

break;

case Activity.RESULT\_CANCELED:

break;

}

break;

}

}

private void checkPermissions() {

int permissionLocation = ContextCompat.checkSelfPermission(mContex,

android.Manifest.permission.ACCESS\_FINE\_LOCATION);

List<String> listPermissionsNeeded = new ArrayList<>();

if (permissionLocation != PackageManager.PERMISSION\_GRANTED) {

listPermissionsNeeded.add(android.Manifest.permission.ACCESS\_FINE\_LOCATION);

if (!listPermissionsNeeded.isEmpty()) {

ActivityCompat.requestPermissions(getActivity(),

listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), REQUEST\_ID\_MULTIPLE\_PERMISSIONS);

}

} else {

getMyLocation();

}

}

@Override

public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

int permissionLocation = ContextCompat.checkSelfPermission(mContex,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionLocation == PackageManager.PERMISSION\_GRANTED) {

getMyLocation();

}

}

@Override

public void onPause() {

super.onPause();

if (googleApiClient != null) {

googleApiClient.stopAutoManage(getActivity());

googleApiClient.disconnect();

status = new com.aris.crowdreporting.HelperUtils.Status("offline");

}

}

@Override

public void onResume() {

super.onResume();

status = new com.aris.crowdreporting.HelperUtils.Status("online");

}

}

* + 1. **NearRecyclerAdapter**

package com.aris.crowdreporting.Adapters;

import android.Manifest;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Location;

import android.location.LocationManager;

import android.support.annotation.NonNull;

import android.support.v4.app.ActivityCompat;

import android.support.v7.widget.CardView;

import android.support.v7.widget.RecyclerView;

import android.text.format.DateFormat;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.DetailActivity;

import com.aris.crowdreporting.Activities.DirectionsActivity;

import com.aris.crowdreporting.HelperClasses.Near;

import com.aris.crowdreporting.HelperClasses.User;

import com.aris.crowdreporting.Activities.MapsActivity;

import com.aris.crowdreporting.HelperUtils.TimeAgo;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Date;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;

public class NearRecyclerAdapter extends RecyclerView.Adapter<NearRecyclerAdapter.ViewHolder> {

public List<Near> near\_list;

public List<User> userList;

private FusedLocationProviderClient client;

public Context context;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

LocationManager locationManager;

private static final String TAG = "MyActivity";

public NearRecyclerAdapter(List<Near> near\_list) {

this.near\_list = near\_list;

// this.userList = userList;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {

View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.near\_row, viewGroup, false);

context = viewGroup.getContext();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

return new NearRecyclerAdapter.ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder viewHolder, int i) {

final String blogPostId = near\_list.get(i).BlogPostId;

final String currentUserId = firebaseAuth.getCurrentUser().getUid();

final String blogUserId = near\_list.get(i).getUser\_id();

String desc\_data = near\_list.get(i).getDesc();

viewHolder.setNearDescText(desc\_data);

String address\_data = near\_list.get(i).getAddress();

viewHolder.setAddress(address\_data);

String image\_url = near\_list.get(i).getImage\_uri();

String thumbUri = near\_list.get(i).getImage\_thumb();

viewHolder.setBlogImage(image\_url, thumbUri);

String user\_id = near\_list.get(i).getUser\_id();

//User Data will be retrieved here...

firebaseFirestore.collection("Users").document(user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

String userName = task.getResult().getString("name");

// String userImage = task.getResult().getString("image");

viewHolder.setUserData(userName);

} else {

//Firebase Exception

}

}

});

client = LocationServices.getFusedLocationProviderClient(context);

if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

client.getLastLocation().addOnSuccessListener((Activity) context, new OnSuccessListener<Location>() {

@Override

public void onSuccess(Location location) {

if (location != null){

Double lat\_a = location.getLatitude();

Double lng\_a = location.getLongitude();

String ll = near\_list.get(i).getLatitude();

Double lat\_b = Double.parseDouble(ll);

String lg = near\_list.get(i).getLongitude();

Double lng\_b = Double.parseDouble(lg);

String descMap = near\_list.get(i).getDesc();

viewHolder.getDistance(lat\_a, lng\_a, lat\_b, lng\_b);

viewHolder.seeMap.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intentMap = new Intent(context, DirectionsActivity.class);

intentMap.putExtra("lat\_a", lat\_a);

intentMap.putExtra("lon\_a", lng\_a);

intentMap.putExtra("lat\_b", lat\_b);

intentMap.putExtra("lon\_b", lng\_b);

intentMap.putExtra("descMap", descMap);

context.startActivity(intentMap);

}

});

}

}

});

long millisecond = near\_list.get(i).getTimestamp().getTime();

String dateString = DateFormat.format("MM/dd/yyyy", new Date(millisecond)).toString();

String timeAgo = TimeAgo.getTimeAgo(millisecond);

viewHolder.setNearTime(timeAgo);

viewHolder.cardView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context, DetailActivity.class);

intent.putExtra("blog\_id",blogPostId);

intent.putExtra("user\_id",blogUserId);

intent.putExtra("imurl",image\_url);

intent.putExtra("desc",desc\_data);

intent.putExtra("time\_post",timeAgo);

// intent.putExtra("address",address\_data);

context.startActivity(intent);

}

});

}

@Override

public int getItemCount() {

return near\_list.size();

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View mView;

private TextView nearDesc;

private TextView blogUserName;

private CircleImageView blogUserImage;

private TextView nearU, nearTime, nearAddress;

private ImageView blogImageView, seeMap;

private CardView cardView;

public ViewHolder(@NonNull View itemView) {

super(itemView);

mView = itemView;

nearU = mView.findViewById(R.id.jarak);

seeMap = mView.findViewById(R.id.see\_map);

cardView = mView.findViewById(R.id.card\_near);

}

public void setNearDescText(String descText) {

nearDesc = mView.findViewById(R.id.near\_desc);

String upperString = descText.substring(0,1).toUpperCase() + descText.substring(1);

nearDesc.setText(upperString);

}

public void setNearTime(String timePost){

nearTime = mView.findViewById(R.id.near\_time);

nearTime.setText(timePost);

}

public void setUserData(String name) {

// blogUserImage = mView.findViewById(R.id.near\_user\_image);

blogUserName = mView.findViewById(R.id.near\_username);

blogUserName.setText(name);

// RequestOptions placeholderOption = new RequestOptions();

// placeholderOption.placeholder(R.drawable.profile\_placeholder);

//

// Glide.with(context).applyDefaultRequestOptions(placeholderOption).load(image).into(blogUserImage);

}

public void setBlogImage(String downloadUri, String thumbUri){

blogImageView = mView.findViewById(R.id.near\_user\_image);

RequestOptions requestOptions = new RequestOptions();

requestOptions.placeholder(R.drawable.profile\_placeholder);

Glide.with(context).applyDefaultRequestOptions(requestOptions).load(downloadUri).thumbnail(

Glide.with(context).load(thumbUri)

).into(blogImageView);

}

public void getDistance(Double lat\_a, Double lng\_a, Double lat\_b, Double lng\_b) {

// earth radius is in mile

double earthRadius = 3958.75;

double latDiff = Math.toRadians(lat\_b - lat\_a);

double lngDiff = Math.toRadians(lng\_b - lng\_a);

double a = Math.sin(latDiff / 2) \* Math.sin(latDiff / 2)

+ Math.cos(Math.toRadians(lat\_a))

\* Math.cos(Math.toRadians(lat\_b)) \* Math.sin(lngDiff / 2)

\* Math.sin(lngDiff / 2);

double c = 2 \* Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

double distance = earthRadius \* c;

int meterConversion = 1609;

double kmConvertion = 1.6093;

// return new Float(distance \* meterConversion).floatValue();

// return String.format("%.2f", new Float(distance \* kmConvertion).floatValue()) + " km";

String fin = String.format("%.2f", new Float(distance \* kmConvertion).floatValue()) + " km";

nearU = mView.findViewById(R.id.jarak);

nearU.setText(""+ fin);

// return String.format("%.2f", distance)+" m";

}

public void setAddress(String address\_data) {

nearAddress = mView.findViewById(R.id.near\_address);

nearAddress.setText(address\_data);

}

}

}

* + 1. **MapsActivity**

package com.aris.crowdreporting.Activities;

import android.graphics.Bitmap;

import android.graphics.Color;

import android.graphics.drawable.BitmapDrawable;

import android.os.Bundle;

import android.support.design.widget.Snackbar;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import com.akexorcist.googledirection.DirectionCallback;

import com.akexorcist.googledirection.GoogleDirection;

import com.akexorcist.googledirection.constant.TransportMode;

import com.akexorcist.googledirection.model.Direction;

import com.akexorcist.googledirection.model.Route;

import com.akexorcist.googledirection.util.DirectionConverter;

import com.aris.crowdreporting.R;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.MapFragment;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.LatLngBounds;

import com.google.android.gms.maps.model.MarkerOptions;

import java.util.ArrayList;

public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback, View.OnClickListener, DirectionCallback {

private Button btnRequestDirection;

private GoogleMap googleMap;

private String serverKey = "AIzaSyCI-4RaDocwneRsw2ryTRPMf7NzGV-F1CE"; //"AIzaSyBfrkZXKYufFH2V-f-5HbO748pa-IQoW-U";

// private LatLng origin = new LatLng(37.7849569, -122.4068855);

// private LatLng destination = new LatLng(37.7814432, -122.4460177);

Double lat\_a, lat\_b, long\_a, long\_b;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_maps);

// btnRequestDirection = findViewById(R.id.btnGetDirection);

// btnRequestDirection.setOnClickListener(this);

lat\_a = getIntent().getDoubleExtra("lat\_a", -6.244208);

long\_a = getIntent().getDoubleExtra("lon\_a", 107.014806);

lat\_b = getIntent().getDoubleExtra("lat\_b", -6.244208);

long\_b = getIntent().getDoubleExtra("lon\_b", 107.014806);

MapFragment mapFragment = (MapFragment) getFragmentManager()

.findFragmentById(R.id.mapNearBy);

mapFragment.getMapAsync(this);

requestDirection();

}

@Override

public void onMapReady(GoogleMap googleMap) {

this.googleMap = googleMap;

googleMap.setMapType(GoogleMap.MAP\_TYPE\_TERRAIN);

LatLng origin = new LatLng(lat\_a, long\_a);

LatLng destination = new LatLng(lat\_b, long\_b);

String descMap = getIntent().getStringExtra("descMap");

// GANTI UKURAN Pin Maps

int height = 100;

int width = 100;

BitmapDrawable bitmapdraw=(BitmapDrawable)getResources().getDrawable(R.drawable.pinmu1);

Bitmap b=bitmapdraw.getBitmap();

Bitmap smallMarker = Bitmap.createScaledBitmap(b, width, height, false);

BitmapDrawable bitmapdraw2=(BitmapDrawable)getResources().getDrawable(R.drawable.pinkejadian);

Bitmap b2=bitmapdraw2.getBitmap();

Bitmap smallMarker2 = Bitmap.createScaledBitmap(b2, width, height, false);

googleMap.addMarker(new MarkerOptions().position(origin).title("Lokasi Anda")

.icon(BitmapDescriptorFactory.fromBitmap(smallMarker)));

googleMap.addMarker(new MarkerOptions().position(destination).title("Lokasi Kejadian").snippet(descMap)

.icon(BitmapDescriptorFactory.fromBitmap(smallMarker2)));

//Zoom and Animateku

LatLng uman = new LatLng(lat\_b,long\_b);

googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(uman, 14));

}

@Override

public void onClick(View v) {

int id = v.getId();

// if (id == R.id.btnGetDirection) {

// requestDirection();

// }

}

public void requestDirection() {

LatLng origin = new LatLng(lat\_a, long\_a);

LatLng destination = new LatLng(lat\_b, long\_b);

// Snackbar.make(btnRequestDirection, "Direction Requesting...", Snackbar.LENGTH\_SHORT).show();

Toast.makeText(this, "Direction Requesting...", Toast.LENGTH\_SHORT).show();

GoogleDirection.withServerKey(serverKey)

.from(origin)

.to(destination)

.transportMode(TransportMode.DRIVING)

.execute(this);

}

@Override

public void onDirectionSuccess(Direction direction, String rawBody) {

// Snackbar.make(btnRequestDirection, "Success with status : " + direction.getStatus(), Snackbar.LENGTH\_SHORT).show();

Toast.makeText(this, "Success with status : " + direction.getStatus(), Toast.LENGTH\_SHORT).show();

if (direction.isOK()) {

LatLng origin = new LatLng(lat\_a, long\_a);

LatLng destination = new LatLng(lat\_b, long\_b);

Route route = direction.getRouteList().get(0);

// googleMap.addMarker(new MarkerOptions().position(origin).title("Lokasi Anda"));

// googleMap.addMarker(new MarkerOptions().position(destination).title("Lokasi Kejadian"));

ArrayList<LatLng> directionPositionList = route.getLegList().get(0).getDirectionPoint();

googleMap.addPolyline(DirectionConverter.createPolyline(this, directionPositionList, 5, Color.RED));

setCameraWithCoordinationBounds(route);

// btnRequestDirection.setVisibility(View.GONE);

} else {

// Snackbar.make(btnRequestDirection, direction.getStatus(), Snackbar.LENGTH\_SHORT).show();

Toast.makeText(this, direction.getStatus(), Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onDirectionFailure(Throwable t) {

// Snackbar.make(findViewById(R.id.mapNearBy), t.getMessage(), Snackbar.LENGTH\_SHORT).show();

Toast.makeText(this, t.getMessage(), Toast.LENGTH\_SHORT).show();

}

private void setCameraWithCoordinationBounds(Route route) {

LatLng southwest = route.getBound().getSouthwestCoordination().getCoordination();

LatLng northeast = route.getBound().getNortheastCoordination().getCoordination();

LatLngBounds bounds = new LatLngBounds(southwest, northeast);

googleMap.animateCamera(CameraUpdateFactory.newLatLngBounds(bounds, 100));

}

}

* + 1. **ChatsFragment**

package com.aris.crowdreporting.Fragment;

import android.content.Intent;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.design.widget.TabLayout;

import android.support.v4.app.DialogFragment;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentManager;

import android.support.v4.app.FragmentPagerAdapter;

import android.support.v4.view.ViewPager;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.VerificationEmailActivity;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.R;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.ArrayList;

import java.util.HashMap;

public class ChatsFragment extends DialogFragment {

private final static String TAG = "CHATS\_ACTIVITY";

private EditText editTextTest;

private Button btnTest;

private TabLayout tabLayout;

private ViewPager viewPager;

private FragmentManager fragmentManager;

private FirebaseFirestore firebaseFirestore;

private Status status;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_chats, container, false);

tabLayout = view.findViewById(R.id.tablay);

viewPager = view.findViewById(R.id.view\_pager);

ViewPagerAdapter viewPagerAdapter = new ViewPagerAdapter(getChildFragmentManager());

viewPagerAdapter.addFragment(new VPChatsFragment(), "Chats");

viewPagerAdapter.addFragment(new VPUsersChatFragment(), "Users");

viewPager.setAdapter(viewPagerAdapter);

tabLayout.setupWithViewPager(viewPager);

return view;

}

class ViewPagerAdapter extends FragmentPagerAdapter{

private ArrayList<Fragment> fragments;

private ArrayList<String> titles;

public ViewPagerAdapter(FragmentManager fm) {

super(fm);

this.fragments = new ArrayList<>();

this.titles = new ArrayList<>();

}

@Override

public Fragment getItem(int i) {

return fragments.get(i);

}

@Override

public int getCount() {

return fragments.size();

}

public void addFragment(Fragment fragment, String title){

fragments.add(fragment);

titles.add(title);

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return titles.get(position);

}

}

@Override

public void onResume() {

super.onResume();

status = new Status("online");

}

@Override

public void onPause() {

super.onPause();

status = new Status("offline");

}

@Override

public void onStop() {

super.onStop();

status = new Status("offline");

}

}

* + 1. **ChatUserAdapter**

package com.aris.crowdreporting.Adapters;

import android.content.Context;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.v7.widget.RecyclerView;

import android.text.format.DateFormat;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.MessageActivity;

import com.aris.crowdreporting.HelperClasses.Chats;

import com.aris.crowdreporting.HelperClasses.User;

import com.aris.crowdreporting.HelperUtils.TimeAgo;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.Date;

import java.util.List;

import javax.annotation.Nullable;

import de.hdodenhof.circleimageview.CircleImageView;

public class ChatsUsersAdapter extends RecyclerView.Adapter<ChatsUsersAdapter.ViewHolder> {

public List<User> userList;

public Context context;

public boolean isChat;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private FirebaseUser firebaseUser;

String theLastMsg;

long lastTimeChat;

public ChatsUsersAdapter(List<User> userList, boolean isChat){

this.userList = userList;

this.isChat = isChat;

}

@Override

public ChatsUsersAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.userchats\_row, parent, false);

context = parent.getContext();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

firebaseUser = firebaseAuth.getCurrentUser();

return new ChatsUsersAdapter.ViewHolder(view);

}

@Override

public void onBindViewHolder(final ChatsUsersAdapter.ViewHolder holder, int position) {

holder.setIsRecyclable(false);

final User user = userList.get(position);

String image = user.getImage();

String name = user.getName();

// holder.setUserData(name, image);

holder.setChatUserImage(image);

holder.setChatUserName(name);

holder.setChatTimestamp();

//start chat

holder.relativeLayout.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intentSC = new Intent(context, MessageActivity.class);

intentSC.putExtra("user\_id\_msg", user.getUser\_id());

context.startActivity(intentSC);

}

});

if (isChat){

holder.lastMessage(user.getUser\_id(), holder.lastMsg, userList.get(position).getUser\_id());

// Toast.makeText(context, ""+userList.get(position).getUser\_id(), Toast.LENGTH\_SHORT).show();

} else {

holder.lastMsg.setVisibility(View.GONE);

}

if (isChat){

if (user.getStatus().equals("online")){

holder.img\_on.setVisibility(View.VISIBLE);

holder.img\_off.setVisibility(View.GONE);

} else {

holder.img\_on.setVisibility(View.GONE);

holder.img\_off.setVisibility(View.VISIBLE);

}

} else {

holder.img\_on.setVisibility(View.GONE);

holder.img\_off.setVisibility(View.GONE);

}

}

@Override

public int getItemCount() {

if(userList != null) {

return userList.size();

} else {

return 0;

}

}

public class ViewHolder extends RecyclerView.ViewHolder {

private View mView;

private CircleImageView chatUserImage, img\_on, img\_off;

private TextView chatUserName,chatTimestamp, lastMsg;

private RelativeLayout relativeLayout;

public ViewHolder(View itemView) {

super(itemView);

mView = itemView;

relativeLayout = mView.findViewById(R.id.rl\_userchatrow);

chatUserImage = mView.findViewById(R.id.chat\_userimage);

chatUserName = mView.findViewById(R.id.chat\_username);

chatTimestamp = mView.findViewById(R.id.chat\_timestamp);

img\_on = mView.findViewById(R.id.img\_on);

img\_off = mView.findViewById(R.id.img\_off);

lastMsg = mView.findViewById(R.id.last\_msg);

}

public void setChatUserImage(String image){

RequestOptions placeholderOption = new RequestOptions();

placeholderOption.placeholder(R.drawable.profile\_placeholder);

Glide.with(context).applyDefaultRequestOptions(placeholderOption).load(image).into(chatUserImage);

}

public void setChatUserName(String name){

chatUserName.setText(name);

}

//CHeck last msg

public void lastMessage(String userid, TextView lastMsg, String pss){

theLastMsg = "default";

lastTimeChat = 0;

Query query = firebaseFirestore.collection("Chats/" + firebaseUser.getUid() + "/" + pss)

.orderBy("timestamp");

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

for (DocumentSnapshot doc : queryDocumentSnapshots.getDocuments()){

Chats chats = doc.toObject(Chats.class);

if (chats.getReceiver().equals(firebaseUser.getUid()) && chats.getSender().equals(userid)

|| chats.getReceiver().equals(userid) && chats.getSender().equals(firebaseUser.getUid())){

theLastMsg = chats.getMessage();

try {

lastTimeChat = chats.getTimestamp().getTime();

} catch (Exception e1){

e1.printStackTrace();

}

}

}

switch (theLastMsg){

case "default":

lastMsg.setText("");

chatTimestamp.setText("");

break;

default:

lastMsg.setText(theLastMsg);

try {

chatTimestamp.setText(TimeAgo.getTimeAgo(lastTimeChat).toString());

} catch (Exception e1){

chatTimestamp.setText("");

}

break;

}

theLastMsg = "default";

lastTimeChat = 0;

}

});

}

public void setChatTimestamp() {

}

}

}

* + 1. **VPChatFragment**

package com.aris.crowdreporting.Fragment;

import android.content.Context;

import android.net.Uri;

import android.os.Bundle;

import android.support.annotation.NonNull;

import android.support.v4.app.Fragment;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Adapters.ChatsUsersAdapter;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.HelperClasses.Chatlist;

import com.aris.crowdreporting.HelperClasses.Chats;

import com.aris.crowdreporting.HelperClasses.Comments;

import com.aris.crowdreporting.HelperClasses.User;

import com.aris.crowdreporting.Notifications.Token;

import com.aris.crowdreporting.R;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.iid.FirebaseInstanceId;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import javax.annotation.Nullable;

import static android.support.constraint.Constraints.TAG;

public class VPChatsFragment extends Fragment {

private RecyclerView recyclerView;

private ChatsUsersAdapter chatsUsersAdapter;

private TextView noMsg;

private ImageView noMsgImg;

private List<User> mUsers;

private List<Chatlist> userList;

FirebaseUser firebaseUser;

FirebaseFirestore firebaseFirestore;

public VPChatsFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_vpchats, container, false);

noMsg = view.findViewById(R.id.no\_msg);

noMsgImg = view.findViewById(R.id.no\_msg\_img);

recyclerView = view.findViewById(R.id.rv\_vpchat);

recyclerView.setHasFixedSize(true);

recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));

firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

firebaseFirestore = FirebaseFirestore.getInstance();

userList = new ArrayList<>();

Query query = firebaseFirestore.collection("Chatlist/" + firebaseUser.getUid() + "/" +firebaseUser.getUid());

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

// userList.clear();

for (DocumentChange doc : queryDocumentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

Chatlist chatlist = doc.getDocument().toObject(Chatlist.class);

userList.add(chatlist);

}

}

if (userList.isEmpty()){

noMsg.setVisibility(View.VISIBLE);

noMsgImg.setVisibility(View.VISIBLE);

}

chatList();

}

});

updateToken(FirebaseInstanceId.getInstance().getToken());

return view;

}

private void updateToken(String token){

DocumentReference documentReference = FirebaseFirestore.getInstance().collection("Tokens").document(firebaseUser.getUid());

Token token1 = new Token(token);

documentReference.set(token1);

}

private void chatList() {

mUsers = new ArrayList<>();

Query query1 = firebaseFirestore.collection("Users");

query1.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

mUsers.clear();

for (DocumentSnapshot doc : queryDocumentSnapshots.getDocuments()) {

User user = doc.toObject(User.class);

for (Chatlist chatlist : userList){

if (user.getUser\_id().equals(chatlist.getId())){

mUsers.add(user);

}

}

}

chatsUsersAdapter = new ChatsUsersAdapter(mUsers, true);

recyclerView.setAdapter(chatsUsersAdapter);

}

});

}

}

* + 1. **VPUserChatFragment**

package com.aris.crowdreporting.Fragment;

import android.graphics.Color;

import android.os.Bundle;

import android.support.annotation.NonNull;

import android.support.v4.app.Fragment;

import android.support.v4.widget.SwipeRefreshLayout;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.support.v7.widget.SearchView;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.Toast;

import com.aris.crowdreporting.Adapters.BlogRecyclerAdapter;

import com.aris.crowdreporting.Adapters.ChatsUsersAdapter;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.HelperClasses.User;

import com.aris.crowdreporting.R;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

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.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import javax.annotation.Nullable;

import dmax.dialog.SpotsDialog;

import static android.support.constraint.Constraints.TAG;

public class VPUsersChatFragment extends Fragment {

private RecyclerView recyclerView;

private SwipeRefreshLayout pullToRefresh;

private ChatsUsersAdapter chatsUsersAdapter;

private List<User> userList;

private EditText txtSearchUsers;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private FirebaseUser firebaseUser;

private DocumentSnapshot lastVisible;

private Boolean isFirstPageFirstLoad = true;

private String user\_id;

private SpotsDialog dialog;

public VPUsersChatFragment() {

// Required empty public constructor

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_vpusers\_chat, container, false);

userList = new ArrayList<>();

pullToRefresh = view.findViewById(R.id.pull1);

pullToRefresh.setColorSchemeColors(Color.CYAN, Color.YELLOW, Color.MAGENTA);

recyclerView = view.findViewById(R.id.rv\_chatsusers);

txtSearchUsers = view.findViewById(R.id.src\_users);

firebaseAuth = FirebaseAuth.getInstance();

chatsUsersAdapter = new ChatsUsersAdapter(userList, true);

recyclerView.setLayoutManager(new LinearLayoutManager(container.getContext()));

recyclerView.setAdapter(chatsUsersAdapter);

recyclerView.setHasFixedSize(true);

dialog = new SpotsDialog(getContext(), "Loading..");

dialog.show();

if (firebaseAuth.getCurrentUser() != null) {

firebaseFirestore = FirebaseFirestore.getInstance();

user\_id = firebaseAuth.getCurrentUser().getUid();

recyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

Boolean reachedBottom = !recyclerView.canScrollVertically(1);

if (reachedBottom) {

// loadMorePost();

}

}

});

Query firstQuery = firebaseFirestore.collection("Users").orderBy("name");

firstQuery.addSnapshotListener(getActivity(), new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

if (isFirstPageFirstLoad) {

lastVisible = documentSnapshots.getDocuments().get(documentSnapshots.size() - 1);

userList.clear();

}

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String blogPostId = doc.getDocument().getId();

User userPost = doc.getDocument().toObject(User.class);

if (isFirstPageFirstLoad) {

if (!userPost.getUser\_id().equals(FirebaseAuth.getInstance().getCurrentUser().getUid())){

userList.add(userPost);

}

chatsUsersAdapter.notifyItemInserted(userList.size());

chatsUsersAdapter.notifyDataSetChanged();

} else {

if (!userPost.getUser\_id().equals(FirebaseAuth.getInstance().getCurrentUser().getUid())){

userList.add(0, userPost);

}

chatsUsersAdapter.notifyItemInserted(0);

chatsUsersAdapter.notifyDataSetChanged();

}

dialog.dismiss();

}

}

isFirstPageFirstLoad = false;

}

}

});

}

pullToRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

@Override

public void onRefresh() {

chatsUsersAdapter.notifyDataSetChanged();

pullToRefresh.setRefreshing(false);

}

});

txtSearchUsers.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence s, int start, int count, int after) {

}

@Override

public void onTextChanged(CharSequence s, int start, int before, int count) {

searchUsers(s.toString());

}

@Override

public void afterTextChanged(Editable s) {

}

});

// Inflate the layout for this fragment

return view;

}

private void searchUsers(String usersearch) {

FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();

Query query = firebaseFirestore.collection("Users").orderBy("name", Query.Direction.ASCENDING);

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

userList.clear();

for (DocumentSnapshot snapshot : queryDocumentSnapshots.getDocuments()){

User user = snapshot.toObject(User.class);

if (user.getName().toLowerCase().contains(usersearch) && !user.getUser\_id().equals(firebaseUser.getUid())){

userList.add(user);

}

}

chatsUsersAdapter = new ChatsUsersAdapter(userList, false);

recyclerView.setAdapter(chatsUsersAdapter);

}

});

}

public void loadMorePost() {

if (firebaseAuth.getCurrentUser() != null) {

Query nextQuery = firebaseFirestore.collection("Users")

.startAfter(lastVisible)

.limit(2);

nextQuery.addSnapshotListener(getActivity(), new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (!documentSnapshots.isEmpty()) {

lastVisible = documentSnapshots.getDocuments().get(documentSnapshots.size() - 1);

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String blogPostId = doc.getDocument().getId();

User userPost = doc.getDocument().toObject(User.class);

if (!userPost.getUser\_id().equals(FirebaseAuth.getInstance().getCurrentUser().getUid())){

userList.add( userPost);

}

chatsUsersAdapter.notifyItemInserted(userList.size());

}

}

}

}

});

}

}

}

* + 1. **MessageActivity**

package com.aris.crowdreporting.Activities;

import android.app.Activity;

import android.content.Intent;

import android.content.SharedPreferences;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.support.v7.widget.Toolbar;

import android.util.Log;

import android.view.Gravity;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.APIService;

import com.aris.crowdreporting.Adapters.MessageAdapter;

import com.aris.crowdreporting.Adapters.MsgAdapter;

import com.aris.crowdreporting.HelperClasses.Chats;

import com.aris.crowdreporting.HelperClasses.Comments;

import com.aris.crowdreporting.HelperClasses.User;

import com.aris.crowdreporting.HelperUtils.Status;

import com.aris.crowdreporting.Notifications.Client;

import com.aris.crowdreporting.Notifications.Data;

import com.aris.crowdreporting.Notifications.MyResponse;

import com.aris.crowdreporting.Notifications.Sender;

import com.aris.crowdreporting.Notifications.Token;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.firebase.client.Firebase;

import com.firebase.client.ValueEventListener;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.ListenerRegistration;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.Nullable;

import de.hdodenhof.circleimageview.CircleImageView;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

public class MessageActivity extends AppCompatActivity {

CircleImageView profileImageMsg;

TextView username\_msg, txt\_useronline;

ImageButton btnSendMsg;

EditText txt\_send;

MessageAdapter messageAdapter;

List<Chats> mChats;

RecyclerView recyclerView;

FirebaseUser firebaseUser;

FirebaseFirestore firebaseFirestore;

Intent intent;

APIService apiService;

boolean notify = false;

private String TAG = "MSG\_ACTY";

private final List<Chats> chatsList = new ArrayList<>();

private LinearLayoutManager linearLayoutManager;

private MsgAdapter msgAdapter;

private String user\_id\_msg;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_message);

Toolbar toolbar = findViewById(R.id.toolbar\_msg);

setSupportActionBar(toolbar);

getSupportActionBar().setTitle("");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

toolbar.setNavigationOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// startActivity(new Intent(MessageActivity.this, MainActivity.class).setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP));

finish();

}

});

apiService = Client.getClient("https://fcm.googleapis.com/").create(APIService.class);

intent = getIntent();

user\_id\_msg = intent.getStringExtra("user\_id\_msg");

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

String cUser = firebaseUser.getUid();

msgAdapter = new MsgAdapter(chatsList);

recyclerView = findViewById(R.id.rv\_msg);

linearLayoutManager = new LinearLayoutManager(getApplicationContext());

linearLayoutManager.setStackFromEnd(true);

recyclerView.setHasFixedSize(true);

recyclerView.setLayoutManager(linearLayoutManager);

recyclerView.setAdapter(msgAdapter);

loadMsg(cUser, user\_id\_msg);

profileImageMsg = findViewById(R.id.image\_profile\_msg);

username\_msg = findViewById(R.id.username\_msg);

btnSendMsg = findViewById(R.id.send\_msg);

txt\_send = findViewById(R.id.text\_send);

txt\_useronline = findViewById(R.id.user\_online);

firebaseFirestore.collection("Users").document(user\_id\_msg).addSnapshotListener(new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(@Nullable DocumentSnapshot documentSnapshot, @Nullable FirebaseFirestoreException e) {

String name = documentSnapshot.getString("name");

String image = documentSnapshot.getString("image");

String userstts = documentSnapshot.getString("status");

//SET IMG USER CHAT

Glide.with(getApplicationContext()).load(image).into(profileImageMsg);

//SET USERNAME

username\_msg.setText(name);

//SET STATUS

if (userstts.equals("online")){

txt\_useronline.setVisibility(View.VISIBLE);

} else {

txt\_useronline.setVisibility(View.GONE);

}

// readMessage(firebaseUser.getUid(), user\_id\_msg, image);

}

});

btnSendMsg.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

notify = true;

String msg = txt\_send.getText().toString();

if (!msg.equals("")){

sendMessage(firebaseUser.getUid(), user\_id\_msg, msg);

} else {

Toast toast = Toast.makeText(MessageActivity.this, "You can't send empty message", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

}

txt\_send.setText("");

}

});

}

private void loadMsg(String cUser, String user\_id\_msg) {

String chatRef = "Chats/" + cUser + "/" + user\_id\_msg;

Query query = firebaseFirestore.collection(chatRef).orderBy("timestamp", Query.Direction.ASCENDING);

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

for (DocumentChange doc : queryDocumentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

Chats chats = doc.getDocument().toObject(Chats.class);

chatsList.add(chats);

msgAdapter.notifyDataSetChanged();

}

}

msgAdapter = new MsgAdapter(chatsList);

recyclerView.setAdapter(msgAdapter);

msgAdapter.notifyDataSetChanged();

}

});

}

private void sendMessage(String sender, String receiver, String message) {

//ADD CHATLIST JIKA TIDAK ADA

HashMap<String, Object> hs = new HashMap<>();

hs.put("id", receiver);

hs.put("timestamp", FieldValue.serverTimestamp());

HashMap<String, Object> hs1 = new HashMap<>();

hs1.put("id", firebaseUser.getUid());

hs1.put("timestamp", FieldValue.serverTimestamp());

firebaseFirestore.collection("Chatlist/" + firebaseUser.getUid() +"/"+firebaseUser.getUid())

.document(receiver).addSnapshotListener(new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(@Nullable DocumentSnapshot documentSnapshot, @Nullable FirebaseFirestoreException e) {

if (!documentSnapshot.exists()){

firebaseFirestore.collection("Chatlist/" + firebaseUser.getUid() +"/"+firebaseUser.getUid())

.document(receiver).set(hs);

firebaseFirestore.collection("Chatlist/" + receiver +"/"+receiver)

.document(firebaseUser.getUid()).set(hs1);

}

}

});

// ADD MSG

HashMap<String, Object> hashMap = new HashMap<>();

hashMap.put("sender", sender);

hashMap.put("receiver", receiver);

hashMap.put("message", message);

hashMap.put("timestamp", FieldValue.serverTimestamp());

HashMap<String, Object> hashMap1 = new HashMap<>();

hashMap1.put("sender", sender);

hashMap1.put("receiver", receiver);

hashMap1.put("message", message);

hashMap1.put("timestamp", FieldValue.serverTimestamp());

String current\_user\_ref = "Chats/" + firebaseUser.getUid() + "/" + receiver;

String chat\_user\_ref = "Chats/" + receiver + "/" + firebaseUser.getUid();

firebaseFirestore.collection(current\_user\_ref).add(hashMap);

firebaseFirestore.collection(chat\_user\_ref).add(hashMap1);

final String msg = message;

firebaseFirestore.collection("Users").document(firebaseUser.getUid()).addSnapshotListener(new EventListener<DocumentSnapshot>() {

@Override

public void onEvent(@Nullable DocumentSnapshot documentSnapshot, @Nullable FirebaseFirestoreException e) {

User user = documentSnapshot.toObject(User.class);

sendNotification(receiver, user.getName(), msg);

notify = false;

}

});

}

private void sendNotification(String receiver, final String username, String message) {

user\_id\_msg = intent.getStringExtra("user\_id\_msg");

CollectionReference tokens = FirebaseFirestore.getInstance().collection("Tokens");

Query query = tokens;

query.addSnapshotListener(this, new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

Token token = doc.getDocument().toObject(Token.class);

Data data = new Data(firebaseUser.getUid(), R.mipmap.ic\_launcher, username+": "+message, "New Message", user\_id\_msg);

Sender sender = new Sender(data, token.getToken());

apiService.sendNotification(sender)

.enqueue(new Callback<MyResponse>() {

@Override

public void onResponse(Call<MyResponse> call, Response<MyResponse> response) {

if (response.code() == 200){

if (response.body().success != 1){

// Toast.makeText(MessageActivity.this, "Failed!", Toast.LENGTH\_SHORT).show();

Log.d("NOTIF:", "Failed Notif");

}

}

}

@Override

public void onFailure(Call<MyResponse> call, Throwable t) {

}

});

}

}

}

}

});

}

private void currentUser(String userid){

SharedPreferences.Editor editor = getSharedPreferences("PREFS", MODE\_PRIVATE).edit();

editor.putString("currentuser", userid);

editor.apply();

}

private Status status;

@Override

public void onResume() {

super.onResume();

status = new Status("online");

currentUser(user\_id\_msg);

}

@Override

public void onPause() {

super.onPause();

status = new Status("offline");

currentUser("none");

}

@Override

public void onStop() {

super.onStop();

status = new Status("offline");

currentUser("none");

}

}

* + 1. **MsgAdapter**

package com.aris.crowdreporting.Adapters;

import android.content.Context;

import android.support.annotation.NonNull;

import android.support.v7.widget.RecyclerView;

import android.text.format.DateFormat;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import com.aris.crowdreporting.HelperClasses.Chats;

import com.aris.crowdreporting.HelperUtils.TimeAgo;

import com.aris.crowdreporting.R;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.Date;

import java.util.List;

import de.hdodenhof.circleimageview.CircleImageView;

public class MsgAdapter extends RecyclerView.Adapter<MsgAdapter.MsgViewHolder> {

List<Chats> chatsList;

public static final int MSG\_TYPE\_LEFT = 0;

public static final int MSG\_TYPE\_RIGHT = 1;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private FirebaseUser firebaseUser;

private Context context;

public MsgAdapter(List<Chats> chatsList) {

this.chatsList = chatsList;

}

@NonNull

@Override

public MsgViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

firebaseUser = firebaseAuth.getCurrentUser();

context = viewGroup.getContext();

if (i == MSG\_TYPE\_RIGHT) {

View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.chat\_item\_right, viewGroup, false);

return new MsgViewHolder(view);

} else {

View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.chat\_item\_left, viewGroup, false);

return new MsgViewHolder(view);

}

}

@Override

public void onBindViewHolder(@NonNull MsgViewHolder msgViewHolder, int i) {

Chats chats = chatsList.get(i);

//SET MSG

msgViewHolder.showMsg.setText(chats.getMessage());

//SET TIME

try {

long millisecond = chats.getTimestamp().getTime();

String dateString = DateFormat.format("HH.mm", new Date(millisecond)).toString();

String timeAgo = TimeAgo.getTimeAgo(millisecond);

msgViewHolder.timeMsg.setText(timeAgo);

} catch (Exception e){

msgViewHolder.timeMsg.setText("just now");

}

}

@Override

public int getItemCount() {

return chatsList.size();

}

public class MsgViewHolder extends RecyclerView.ViewHolder {

private View mView;

private CircleImageView profileImage;

private TextView showMsg, timeMsg, txtSeen;

public MsgViewHolder(View itemView) {

super(itemView);

mView = itemView;

profileImage = mView.findViewById(R.id.profile\_chat\_img);

showMsg = mView.findViewById(R.id.show\_msg);

timeMsg = mView.findViewById(R.id.time\_msg);

}

}

@Override

public int getItemViewType(int position) {

firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

if (chatsList.get(position).getSender().equals(firebaseUser.getUid())){

return MSG\_TYPE\_RIGHT;

} else {

return MSG\_TYPE\_LEFT;

}

}

}

* + 1. **AccountFragment**

package com.aris.crowdreporting.Fragment;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.graphics.drawable.AnimationDrawable;

import android.location.Location;

import android.net.Uri;

import android.os.Bundle;

import android.support.annotation.NonNull;

import android.support.design.widget.BottomSheetDialog;

import android.support.v4.app.DialogFragment;

import android.support.v7.widget.GridLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.AboutUsActivity;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.Activities.LoginActivity;

import com.aris.crowdreporting.Adapters.MyPhotoRecyclerAdapter;

import com.aris.crowdreporting.R;

import com.aris.crowdreporting.Activities.SetupActivity;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.auth.api.Auth;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.common.api.ResultCallback;

import com.google.android.gms.common.api.Status;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentChange;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.mikhaellopez.circularimageview.CircularImageView;

import java.util.ArrayList;

import java.util.List;

public class AccountFragment extends DialogFragment implements

GoogleApiClient.OnConnectionFailedListener {

private final static String TAG = "PROFILE\_ACTIVITY";

private Context mContext;

private CircularImageView profileImageV;

private TextView usernameV, emailV, phoneV, emptyTxt;

private ImageView emptyImg;

private Uri mainImageUri;

private LinearLayout edit, hapus;

private String user\_id;

private String emailuser\_id;

private StorageReference storageReference;

private FirebaseAuth firebaseAuth;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth.AuthStateListener mAuthListener;

private GoogleApiClient mGoogleApiClient;

private RecyclerView profileBlogListView;

private List<Blog> blog\_list;

private MyPhotoRecyclerAdapter myPhotoRecyclerAdapter;

private Location mylocation;

private GoogleApiClient googleApiClient;

private final static int REQUEST\_CHECK\_SETTINGS\_GPS = 0x1;

private final static int REQUEST\_ID\_MULTIPLE\_PERMISSIONS = 0x2;

private com.aris.crowdreporting.HelperUtils.Status getstatus;

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

mContext = getContext();

// Inflate the layout for this fragment

View view = inflater.inflate(R.layout.fragment\_account, container, false);

// declare and initialize FrameLayout

View frameLayout = (View) view.findViewById(R.id.lini);

// initializing animation drawable by getting background from frame layout

AnimationDrawable animationDrawable = (AnimationDrawable) frameLayout.getBackground();

// setting enter fade animation duration to 5 seconds

animationDrawable.setEnterFadeDuration(2500);

// setting exit fade animation duration to 2 seconds

animationDrawable.setExitFadeDuration(1000);

animationDrawable.start();

animationDrawable.run();

Button popup = view.findViewById(R.id.options);

firebaseAuth = FirebaseAuth.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

storageReference = FirebaseStorage.getInstance().getReference();

user\_id = firebaseAuth.getCurrentUser().getUid();

emailuser\_id = firebaseAuth.getCurrentUser().getEmail();

profileImageV = (CircularImageView) view.findViewById(R.id.profile\_image\_view);

usernameV = (TextView) view.findViewById(R.id.username\_view);

emailV = (TextView) view.findViewById(R.id.email\_view);

phoneV = (TextView) view.findViewById(R.id.phone\_view);

emptyTxt = view.findViewById(R.id.empty);

emptyImg = view.findViewById(R.id.emptyimg);

//USERNYA SIAPA

firebaseFirestore.collection("Users").document(user\_id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

if (task.getResult().exists()) {

String name = task.getResult().getString("name");

String image = task.getResult().getString("image");

String emailku = task.getResult().getString("email");

String phoneku = task.getResult().getString("phone");

mainImageUri = Uri.parse(image);

usernameV.setText("" + name);

emailV.setText("" + emailku);

phoneV.setText("" + phoneku);

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.defaultimage);

try {

Glide.with(mContext).setDefaultRequestOptions(placeholderReq).load(image).into(profileImageV);

} catch (Exception e) {

Log.d(TAG, e.getMessage());

}

}

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(mContext, "" + errMSg, Toast.LENGTH\_SHORT).show();

}

}

});

//PHOTO POST USERNYA

///Recyclerview

blog\_list = new ArrayList<>();

profileBlogListView = view.findViewById(R.id.rv\_images);

myPhotoRecyclerAdapter = new MyPhotoRecyclerAdapter(mContext, blog\_list);

profileBlogListView.setLayoutManager(new GridLayoutManager(mContext, 3));

profileBlogListView.setAdapter(myPhotoRecyclerAdapter);

Query firstQuery = firebaseFirestore.collection("Posts")

.orderBy("timestamp", Query.Direction.DESCENDING)

.whereEqualTo("user\_id", user\_id);

firstQuery.addSnapshotListener(getActivity(), new EventListener<QuerySnapshot>() {

@Override

public void onEvent(QuerySnapshot documentSnapshots, FirebaseFirestoreException e) {

if (e != null) {

Log.w(TAG, "listen:error", e);

return;

}

if (!documentSnapshots.isEmpty()) {

for (DocumentChange doc : documentSnapshots.getDocumentChanges()) {

if (doc.getType() == DocumentChange.Type.ADDED) {

String blogPostId = doc.getDocument().getId();

Blog blogPost = doc.getDocument().toObject(Blog.class).withId(blogPostId);

blog\_list.add(blogPost);

myPhotoRecyclerAdapter.notifyDataSetChanged();

}

}

} else {

profileBlogListView.setVisibility(View.GONE);

emptyTxt.setVisibility(View.VISIBLE);

emptyImg.setVisibility(View.VISIBLE);

}

}

});

popup.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

BottomSheetDialog mBottomSheetDialog = new BottomSheetDialog(getActivity());

View sheetView = getActivity().getLayoutInflater().inflate(R.layout.fragment\_bottom\_sheet\_options, null);

mBottomSheetDialog.setContentView(sheetView);

mBottomSheetDialog.show();

LinearLayout edit = (LinearLayout) sheetView.findViewById(R.id.bottom\_sheet\_edit\_profile);

LinearLayout logout = (LinearLayout) sheetView.findViewById(R.id.bottom\_sheet\_logout);

LinearLayout about = (LinearLayout) sheetView.findViewById(R.id.bottom\_sheet\_about);

edit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getActivity(), SetupActivity.class);

startActivity(intent);

mBottomSheetDialog.dismiss();

}

});

logout.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

signOut();

mBottomSheetDialog.dismiss();

}

});

about.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(getActivity(), AboutUsActivity.class);

startActivity(intent);

mBottomSheetDialog.dismiss();

}

});

}

});

GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)

.requestEmail()

.build();

if (mGoogleApiClient == null || !mGoogleApiClient.isConnected()) {

try {

mGoogleApiClient = new GoogleApiClient.Builder(getActivity())

.enableAutoManage(getActivity() /\* FragmentActivity \*/, this /\* OnConnectionFailedListener \*/)

.addApi(Auth.GOOGLE\_SIGN\_IN\_API, gso)

.build();

} catch (Exception e) {

e.printStackTrace();

}

}

return view;

}

//sign out method

public void signOut() {

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

builder.setMessage("Are you sure to EXIT ?")

.setCancelable(false)

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

if (firebaseAuth.getCurrentUser().getUid() != null) {

getstatus = new com.aris.crowdreporting.HelperUtils.Status("offline");

}

// FirebaseAuth.getInstance().signOut();

try {

if (mGoogleApiClient.isConnected()) {

Auth.GoogleSignInApi.signOut(mGoogleApiClient).setResultCallback(

new ResultCallback<Status>() {

@Override

public void onResult(Status status) {

mGoogleApiClient.disconnect();

//mGoogleApiClient.connect();

// [START\_EXCLUDE]

FirebaseAuth.getInstance().signOut();

Intent i = new Intent(getActivity(), LoginActivity.class);

i.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(i);

// [END\_EXCLUDE]

}

});

} else {

Toast.makeText(getActivity(), "err", Toast.LENGTH\_SHORT).show();

}

} catch (Exception e) {

Log.d(TAG, "" + e.getMessage());

firebaseAuth.signOut();

}

}

})

.setNegativeButton("No", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

AlertDialog alertDialog = builder.create();

alertDialog.show();

}

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override

public void onResume() {

super.onResume();

getstatus = new com.aris.crowdreporting.HelperUtils.Status("online");

}

@Override

public void onPause() {

super.onPause();

try {

if (firebaseAuth.getCurrentUser().getUid() != null) {

getstatus = new com.aris.crowdreporting.HelperUtils.Status("offline");

}

} catch (Exception e){

e.printStackTrace();

}

}

@Override

public void onStop() {

super.onStop();

try {

if (firebaseAuth.getCurrentUser().getUid() != null) {

getstatus = new com.aris.crowdreporting.HelperUtils.Status("offline");

}

} catch (Exception e){

e.printStackTrace();

}

}

}

* + 1. **MyPhotoRecyclerAdapter**

package com.aris.crowdreporting.Adapters;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.support.annotation.NonNull;

import android.support.design.widget.BottomSheetDialog;

import android.support.v7.widget.RecyclerView;

import android.text.format.DateFormat;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.Toast;

import com.aris.crowdreporting.Activities.DetailActivity;

import com.aris.crowdreporting.Activities.EditPostActivity;

import com.aris.crowdreporting.HelperClasses.Blog;

import com.aris.crowdreporting.HelperUtils.TimeAgo;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

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.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import com.google.firebase.firestore.SetOptions;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.annotation.Nullable;

import dmax.dialog.SpotsDialog;

public class MyPhotoRecyclerAdapter extends RecyclerView.Adapter<MyPhotoRecyclerAdapter.ProfileBlogViewHolder> {

private Context context;

private List<Blog> blog\_list;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

public MyPhotoRecyclerAdapter(Context context, List<Blog> blog\_list) {

this.context = context;

this.blog\_list = blog\_list;

}

@NonNull

@Override

public MyPhotoRecyclerAdapter.ProfileBlogViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.myphoto\_row, parent, false);

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

return new ProfileBlogViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull MyPhotoRecyclerAdapter.ProfileBlogViewHolder holder, int position) {

final Blog blogPost = blog\_list.get(holder.getAdapterPosition());

String title = blogPost.getTitle();

String image = blogPost.getImage\_uri();

String thumb = blogPost.getImage\_thumb();

String desc\_data = blogPost.getDesc();

final String blogPostId = blog\_list.get(position).BlogPostId;

long millisecond = blog\_list.get(position).getTimestamp().getTime();

String dateString = DateFormat.format("MM/dd/yyyy", new Date(millisecond)).toString();

String timeAgo = TimeAgo.getTimeAgo(millisecond);

// GlideLoadImage.loadImage(context,holder.blogImage,thumb,image);

RequestOptions placeholderOption = new RequestOptions();

placeholderOption.placeholder(R.drawable.rectangle);

Glide.with(context).applyDefaultRequestOptions(placeholderOption).load(thumb).into(holder.blogImage);

holder.view.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(context, DetailActivity.class);

intent.putExtra("blog\_id", blogPost.BlogPostId);

intent.putExtra("user\_id", blogPost.getUser\_id());

intent.putExtra("imurl", blogPost.getImage\_uri());

intent.putExtra("desc", desc\_data);

intent.putExtra("time\_post", timeAgo);

context.startActivity(intent);

}

});

//longpress

holder.view.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

final String currentUserId = firebaseAuth.getCurrentUser().getUid();

String user\_id = blog\_list.get(position).getUser\_id();

if (user\_id.equals(currentUserId)) {

LayoutInflater li = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

View sheetView = li.inflate(R.layout.fragment\_bottom\_sheet\_account, null);

holder.mBottomSheetDialog = new BottomSheetDialog(context);

holder.mBottomSheetDialog.setContentView(sheetView);

holder.mBottomSheetDialog.show();

holder.edit = sheetView.findViewById(R.id.bottom\_sheet\_edit\_post);

holder.hapus = sheetView.findViewById(R.id.bottom\_sheet\_delete\_post);

holder.edit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(context, EditPostActivity.class);

intent.putExtra("blog\_id", blogPost.BlogPostId);

intent.putExtra("user\_id", blogPost.getUser\_id());

context.startActivity(intent);

holder.mBottomSheetDialog.dismiss();

}

});

holder.hapus.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

AlertDialog.Builder builder = new AlertDialog.Builder(context);

builder.setMessage("Are you sure to DELETE this post ?")

.setCancelable(false)

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

holder.deletePost(blogPost.BlogPostId, position);

holder.mBottomSheetDialog.dismiss();

}

})

.setNegativeButton("No", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

AlertDialog alertDialog = builder.create();

alertDialog.show();

holder.mBottomSheetDialog.dismiss();

}

});

} else {

LayoutInflater li = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

View sheetView = li.inflate(R.layout.fragment\_bottom\_sheet\_lapor, null);

holder.mBottomSheetDialog = new BottomSheetDialog(context);

holder.mBottomSheetDialog.setContentView(sheetView);

holder.mBottomSheetDialog.show();

holder.report = sheetView.findViewById(R.id.bottom\_sheet\_report);

holder.report.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//Lapor Feature

holder.reportPost(blogPostId, currentUserId);

}

});

}

return false;

}

});

}

@Override

public int getItemCount() {

return blog\_list.size();

}

public class ProfileBlogViewHolder extends RecyclerView.ViewHolder {

private ImageView blogImage;

private View view;

private LinearLayout edit, hapus, report;

private BottomSheetDialog mBottomSheetDialog;

private SpotsDialog spotsDialog;

public ProfileBlogViewHolder(View itemView) {

super(itemView);

view = itemView;

blogImage = itemView.findViewById(R.id.blog\_image\_mini);

}

public void deletePost(String id, int index) {

firebaseFirestore.collection("Posts").document(id)

.delete()

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Toast.makeText(context, "Successfully deleted post!", Toast.LENGTH\_SHORT).show();

blog\_list.remove(index);

notifyItemRemoved(index);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(context, "Error deleting post.", Toast.LENGTH\_SHORT).show();

}

});

}

public void reportPost(String blogPostId, String currentUserId) {

spotsDialog = new SpotsDialog(context, "Please Wait.");

spotsDialog.show();

CollectionReference collectionReference = firebaseFirestore.collection("Posts");

collectionReference.document(blogPostId + "/PostHoax/" + currentUserId).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.getResult().exists()){

spotsDialog.dismiss();

Toast.makeText(context, "You has been reported this post!", Toast.LENGTH\_SHORT).show();

mBottomSheetDialog.dismiss();

} else {

Map<String, Object> reportsMap = new HashMap<>();

reportsMap.put("reports", "true");

collectionReference.document(blogPostId).collection("PostHoax").document(currentUserId)

.set(reportsMap).addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

spotsDialog.dismiss();

Toast.makeText(context, "Thanks for reported this post..", Toast.LENGTH\_SHORT).show();

Query query = collectionReference.document(blogPostId).collection("PostHoax");

query.addSnapshotListener(new EventListener<QuerySnapshot>() {

@Override

public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {

if (queryDocumentSnapshots.size() > 3){

firebaseFirestore.collection("Posts").document(blogPostId).set(reportsMap, SetOptions.merge());

}

}

});

mBottomSheetDialog.dismiss();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

spotsDialog.dismiss();

Toast.makeText(context, ""+e.getMessage(), Toast.LENGTH\_SHORT).show();

mBottomSheetDialog.dismiss();

}

});

}

}

});

}

}

}

* + 1. **AboutUsActivity**

package com.aris.crowdreporting.Activities;

import android.app.Dialog;

import android.content.Intent;

import android.net.Uri;

import android.support.annotation.NonNull;

import android.support.constraint.ConstraintLayout;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.storage.FirebaseStorage;

import de.hdodenhof.circleimageview.CircleImageView;

import dmax.dialog.SpotsDialog;

public class AboutUsActivity extends AppCompatActivity {

private Button btnWa, btnKetentuan;

private RelativeLayout c1,c2;

private CircleImageView imgAris;

private SpotsDialog spotsDialog;

private FirebaseAuth firebaseAuth;

private FirebaseStorage firebaseStorage;

private FirebaseFirestore firebaseFirestore;

private Dialog mDialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about\_us);

btnWa = findViewById(R.id.btnOpenWhatsapp);

imgAris = findViewById(R.id.foto\_aris);

c1 = findViewById(R.id.c1);

c2 = findViewById(R.id.c2);

btnKetentuan = findViewById(R.id.showKentuntuan);

firebaseAuth = FirebaseAuth.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseFirestore = FirebaseFirestore.getInstance();

spotsDialog = new SpotsDialog(this, "Wait.");

spotsDialog.show();

String uIdaris = "Ug2oAiwl0gQfPL9RBEIaK9hcF283";

firebaseFirestore.collection("Users").document(uIdaris).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

String fotoArisUrl = task.getResult().get("image").toString();

Animation animation1 = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);

c1.startAnimation(animation1);

Animation an2 = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);

c2.startAnimation(an2);

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.profile\_placeholder);

Glide.with(AboutUsActivity.this).setDefaultRequestOptions(placeholderReq).load(fotoArisUrl).into(imgAris);

spotsDialog.dismiss();

}

});

btnWa.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String nomorHp = "895400054130";

if (nomorHp.isEmpty()) {

Toast.makeText(AboutUsActivity.this, "Nomor Wa Tidak Ada",

Toast.LENGTH\_SHORT).show();

} else {

openWhatsApp("+62" + nomorHp);

}

}

});

btnKetentuan.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

mDialog = new Dialog(AboutUsActivity.this);

mDialog.setContentView(R.layout.popup\_about\_app);

mDialog.show();

}

});

}

private void openWhatsApp(String number) {

String url = "https://wa.me/" + number +"?text=I'm%20interested%20with%20your%20App";

Intent i = new Intent(Intent.ACTION\_VIEW);

i.setData(Uri.parse(url));

startActivity(i);

}

}

* + 1. **EditPostActivity**

package com.aris.crowdreporting.Activities;

import android.Manifest;

import android.app.AlertDialog;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.PorterDuff;

import android.net.Uri;

import android.os.Build;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.support.v7.widget.Toolbar;

import android.text.TextUtils;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ProgressBar;

import android.widget.Toast;

import com.aris.crowdreporting.R;

import com.bumptech.glide.Glide;

import com.bumptech.glide.request.RequestOptions;

import com.developers.imagezipper.ImageZipper;

import com.google.android.gms.tasks.Continuation;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import com.theartofdev.edmodo.cropper.CropImage;

import com.theartofdev.edmodo.cropper.CropImageView;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import java.util.UUID;

import de.hdodenhof.circleimageview.CircleImageView;

import dmax.dialog.SpotsDialog;

public class EditPostActivity extends AppCompatActivity {

private ImageButton mSelectImage;

private EditText mLati, mLongi;

private EditText mDesc;

private Button mSubmit;

private Toolbar newPostToolbar;

private Uri mainImageUri;

private Uri mainImageUriThumb;

private StorageReference storageReference;

private FirebaseFirestore firebaseFirestore;

private FirebaseAuth firebaseAuth;

private String current\_user\_id;

private String blogPostId, blogUserId;

private Bitmap imageBitmap;

private SpotsDialog dialog;

private boolean is\_Changed = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_edit\_post);

newPostToolbar = findViewById(R.id.edit\_post\_toolbar);

setSupportActionBar(newPostToolbar);

getSupportActionBar().setTitle("Edit Post");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

newPostToolbar.getNavigationIcon().setColorFilter(getResources().getColor(R.color.putih), PorterDuff.Mode.SRC\_ATOP);

storageReference = FirebaseStorage.getInstance().getReference();

firebaseFirestore = FirebaseFirestore.getInstance();

firebaseAuth = FirebaseAuth.getInstance();

mSelectImage = (ImageButton) findViewById(R.id.post\_image);

mDesc = (EditText) findViewById(R.id.post\_desc);

current\_user\_id = firebaseAuth.getCurrentUser().getUid();

dialog = new SpotsDialog(EditPostActivity.this);

blogPostId = getIntent().getStringExtra("blog\_id");

blogUserId = getIntent().getStringExtra("user\_id");

dialog.show();

firebaseFirestore.collection("Posts").document(blogPostId).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

if (task.getResult().exists()) {

String desc = task.getResult().getString("desc");

String image = task.getResult().getString("image\_uri");

String image\_thumb = task.getResult().getString("image\_thumb");

mainImageUri = Uri.parse(image);

mainImageUriThumb = Uri.parse(image\_thumb);

mDesc.setText(desc);

RequestOptions placeholderReq = new RequestOptions();

placeholderReq.placeholder(R.drawable.profile\_placeholder);

Glide.with(EditPostActivity.this).setDefaultRequestOptions(placeholderReq).load(image).into(mSelectImage);

}

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(EditPostActivity.this, "" + errMSg, Toast.LENGTH\_SHORT).show();

}

dialog.dismiss();

}

});

mSelectImage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

BringImagePicker();

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.post, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

AlertDialog dialog = new SpotsDialog(EditPostActivity.this, "Uploading...");

//noinspection SimplifiableIfStatement

if (id == R.id.postit) {

final String desc = mDesc.getText().toString();

if (!TextUtils.isEmpty(desc) && mainImageUri != null) {

// setupProgress.setVisibility(View.VISIBLE);

dialog.show();

if (is\_Changed) {

String randomName = UUID.randomUUID().toString();

StorageReference file\_path = storageReference.child("post\_image").child(randomName + ".jpg");

UploadTask uploadTask = file\_path.putFile(mainImageUri);

Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {

@Override

public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {

if (!task.isSuccessful()) {

throw task.getException();

}

// Continue with the task to get the download URL

return file\_path.getDownloadUrl();

}

}).addOnCompleteListener(new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> task) {

if (task.isSuccessful()) {

File newImageFile = new File(mainImageUri.getPath());

try {

File imageZipperFile = new ImageZipper(EditPostActivity.this)

.setQuality(10)

.setMaxWidth(200)

.setMaxHeight(200)

.compressToFile(newImageFile);

} catch (IOException e) {

e.printStackTrace();

}

try {

imageBitmap = new ImageZipper(EditPostActivity.this).compressToBitmap(newImageFile);

} catch (IOException e) {

e.printStackTrace();

}

ByteArrayOutputStream baos = new ByteArrayOutputStream();

imageBitmap.compress(Bitmap.CompressFormat.JPEG, 80, baos);

byte[] thumbData = baos.toByteArray();

final StorageReference ref = storageReference.child("post\_image/thumbs").child(randomName + ".jpg");

UploadTask uploadTask = ref.putBytes(thumbData);

Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {

@Override

public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {

if (!task.isSuccessful()) {

throw task.getException();

}

// Continue with the task to get the download URL

return ref.getDownloadUrl();

}

}).addOnCompleteListener(new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> taskThumb) {

if (taskThumb.isSuccessful()) {

Uri downloadUri = task.getResult();

Uri downloadUriThumb = taskThumb.getResult();

String imageUri = downloadUri.toString();

String imageUriThumb = downloadUriThumb.toString();

String des = desc.toString();

Map<String, Object> postMap = new HashMap<>();

postMap.put("image\_uri", imageUri);

postMap.put("image\_thumb", imageUriThumb);

postMap.put("desc", des);

postMap.put("user\_id", current\_user\_id);

postMap.put("timestamp", FieldValue.serverTimestamp());

postMap.put("reports", "false");

storeFirestore(task, taskThumb, blogPostId, desc);

} else {

// Handle failures

// ...

}

}

});

} else {

String errMsg = task.getException().getMessage();

Toast.makeText(EditPostActivity.this, "Image Error " + errMsg, Toast.LENGTH\_SHORT).show();

// progressBar.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

}

});

} else {

storeFirestore(null, null, blogPostId, desc);

}

} else if (mainImageUri == null) {

Toast toast = Toast.makeText(EditPostActivity.this, "Please select photo", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

} else if (TextUtils.isEmpty(desc)) {

Toast toast = Toast.makeText(EditPostActivity.this, "Please enter Description Post", Toast.LENGTH\_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();

}

return true;

}

return super.onOptionsItemSelected(item);

}

private void BringImagePicker() {

// start picker to get image for cropping and then use the image in cropping activity

CropImage.activity()

.setGuidelines(CropImageView.Guidelines.ON)

.setAspectRatio(1, 1)

.start(EditPostActivity.this);

}

private void storeFirestore(@NonNull Task<Uri> task, @NonNull Task<Uri> taskThumb, String post\_id, String desc) {

SpotsDialog dialog = new SpotsDialog(EditPostActivity.this);

Uri downloadUriImg;

Uri downloadUriThumb;

if (task != null && taskThumb != null) {

downloadUriImg = task.getResult();

downloadUriThumb = taskThumb.getResult();

} else {

downloadUriImg = mainImageUri;

downloadUriThumb = mainImageUriThumb;

}

Map<String, Object> userMap = new HashMap<>();

userMap.put("desc" , desc);

userMap.put("image\_uri", downloadUriImg.toString());

userMap.put("image\_thumb", downloadUriThumb.toString());

firebaseFirestore.collection("Posts").document(post\_id).update(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()){

// setupProgress.setVisibility(View.INVISIBLE);

dialog.show();

Toast.makeText(EditPostActivity.this, "Post Updated", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(EditPostActivity.this, MainActivity.class);

startActivity(intent);

finish();

} else {

String errMSg = task.getException().getMessage();

Toast.makeText(EditPostActivity.this, ""+ errMSg, Toast.LENGTH\_SHORT).show();

}

// setupProgress.setVisibility(View.INVISIBLE);

dialog.dismiss();

}

});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == CropImage.CROP\_IMAGE\_ACTIVITY\_REQUEST\_CODE) {

CropImage.ActivityResult result = CropImage.getActivityResult(data);

if (resultCode == RESULT\_OK) {

mainImageUri = result.getUri();

mSelectImage.setImageURI(mainImageUri);

is\_Changed = true;

} else if (resultCode == CropImage.CROP\_IMAGE\_ACTIVITY\_RESULT\_ERROR\_CODE) {

Exception error = result.getError();

}

}

}

}

* + 1. **Blog**

package com.aris.crowdreporting.HelperClasses;

import java.util.Date;

public class Blog extends BlogPostId{

public String user\_id;

public String image\_uri;

public String desc;

public String image\_thumb;

public String title;

public String address;

public Date timestamp;

public Blog() {

}

public Blog(String user\_id, String image\_uri, String desc, String image\_thumb, Date timestamp, String title, String address) {

this.user\_id = user\_id;

this.image\_uri = image\_uri;

this.desc = desc;

this.image\_thumb = image\_thumb;

this.timestamp = timestamp;

this.title = title;

}

public String getUser\_id() {

return user\_id;

}

public void setUser\_id(String user\_id) {

this.user\_id = user\_id;

}

public String getImage\_uri() {

return image\_uri;

}

public void setImage\_uri(String image\_uri) {

this.image\_uri = image\_uri;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

public String getImage\_thumb() {

return image\_thumb;

}

public void setImage\_thumb(String image\_thumb) {

this.image\_thumb = image\_thumb;

}

public Date getTimestamp() {

return timestamp;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

* + 1. **BlogPostId**

package com.aris.crowdreporting.HelperClasses;

import android.support.annotation.NonNull;

import com.google.firebase.firestore.Exclude;

public class BlogPostId {

@Exclude

public String BlogPostId;

public <T extends BlogPostId> T withId(@NonNull final String id) {

this.BlogPostId = id;

return (T) this;

}

}

* + 1. **Chatlist**

package com.aris.crowdreporting.HelperClasses;

import java.util.Date;

public class Chatlist {

public String id;

public Date timestamp;

public Chatlist() {

}

public Chatlist(String id, Date timestamp) {

this.id = id;

this.timestamp = timestamp;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public Date getTimestamp() {

return timestamp;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

}

* + 1. **Chats**

package com.aris.crowdreporting.HelperClasses;

import java.util.Date;

public class Chats {

private String sender;

private String receiver;

private String message;

private Date timestamp;

public Chats(String sender, String receiver, String message, Date timestamp) {

this.sender = sender;

this.receiver = receiver;

this.message = message;

this.timestamp = timestamp;

}

public Chats() {

}

public String getSender() {

return sender;

}

public void setSender(String sender) {

this.sender = sender;

}

public String getReceiver() {

return receiver;

}

public void setReceiver(String receiver) {

this.receiver = receiver;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public Date getTimestamp() {

return timestamp;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

}

* + 1. **Comments**

package com.aris.crowdreporting.HelperClasses;

import java.util.Date;

public class Comments {

private String message, user\_id;

private Date timestamp;

public Comments(){

}

public Comments(String message, String user\_id, Date timestamp) {

this.message = message;

this.user\_id = user\_id;

this.timestamp = timestamp;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public String getUser\_id() {

return user\_id;

}

public void setUser\_id(String user\_id) {

this.user\_id = user\_id;

}

public Date getTimestamp() {

return timestamp;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

}

* + 1. **Near**

package com.aris.crowdreporting.HelperClasses;

import java.util.Date;

public class Near extends NearPostId{

public String user\_id;

public String image\_uri;

public String desc;

public String longitude;

public String latitude;

public Date timestamp;

public String address;

public String getImage\_thumb() {

return image\_thumb;

}

public void setImage\_thumb(String image\_thumb) {

this.image\_thumb = image\_thumb;

}

public String image\_thumb;

public Near(){

}

public String getUser\_id() {

return user\_id;

}

public void setUser\_id(String user\_id) {

this.user\_id = user\_id;

}

public String getImage\_uri() {

return image\_uri;

}

public void setImage\_uri(String image\_uri) {

this.image\_uri = image\_uri;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

public String getLongitude() {

return longitude;

}

public void setLongitude(String longitude) {

this.longitude = longitude;

}

public String getLatitude() {

return latitude;

}

public void setLatitude(String latitude) {

this.latitude = latitude;

}

public Date getTimestamp() {

return timestamp;

}

public Near(String user\_id, String image\_uri, String desc, String longitude, String latitude, Date timestamp, String image\_thumb, String address) {

this.user\_id = user\_id;

this.image\_uri = image\_uri;

this.desc = desc;

this.longitude = longitude;

this.latitude = latitude;

this.timestamp = timestamp;

this.image\_thumb = image\_thumb;

this.address = address;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

* + 1. **NearPostId**

package com.aris.crowdreporting.HelperClasses;

import android.support.annotation.NonNull;

import com.google.firebase.firestore.Exclude;

public class NearPostId {

@Exclude

public String BlogPostId;

public <T extends NearPostId> T withId(@NonNull final String id) {

this.BlogPostId = id;

return (T) this;

}

}

* + 1. **User**

package com.aris.crowdreporting.HelperClasses;

public class User {

public String image;

public String name;

public String user\_id;

public String status;

public User(){

}

public User(String image, String name, String user\_id, String status) {

this.image = image;

this.name = name;

this.user\_id = user\_id;

this.status = status;

}

public String getImage() {

return image;

}

public void setImage(String image) {

this.image = image;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUser\_id() {

return user\_id;

}

public void setUser\_id(String user\_id) {

this.user\_id = user\_id;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

}

* + 1. **SortPlace**

package com.aris.crowdreporting.HelperUtils;

import com.aris.crowdreporting.HelperClasses.Near;

import com.google.type.LatLng;

import java.util.Comparator;

public class SortPlaces implements Comparator<Near> {

LatLng currentLoc;

public Double lt;

public Double lg;

public SortPlaces(Double lt, Double lg){

this.lt = lt;

this.lg = lg;

}

@Override

public int compare(final Near place1, final Near place2) {

String lat1 = place1.getLatitude();

String lon1 = place1.getLongitude();

String lat2 = place2.getLatitude();

String lon2 = place2.getLongitude();

Double l1 = Double.parseDouble(lat1);

Double lg1 = Double.parseDouble(lon1);

Double l2 = Double.parseDouble(lat2);

Double lg2 = Double.parseDouble(lon2);

double distanceToPlace1 = distance(lt, lg, l1, lg1);

double distanceToPlace2 = distance(lt, lg, l2, lg2);

return (int) (distanceToPlace1 - distanceToPlace2);

}

public double distance(double fromLat, double fromLon, double toLat, double toLon) {

double radius = 6378137; // approximate Earth radius, \*in meters\*

double deltaLat = toLat - fromLat;

double deltaLon = toLon - fromLon;

double angle = 2 \* Math.asin( Math.sqrt(

Math.pow(Math.sin(deltaLat/2), 2) +

Math.cos(fromLat) \* Math.cos(toLat) \*

Math.pow(Math.sin(deltaLon/2), 2) ) );

return radius \* angle;

}

}

* + 1. **Status**

package com.aris.crowdreporting.HelperUtils;

import android.util.Log;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;

import static android.support.constraint.Constraints.TAG;

public class Status {

FirebaseFirestore firebaseFirestore;

public Status(String status) {

firebaseFirestore = FirebaseFirestore.getInstance();

HashMap<String, Object> hashMap = new HashMap<>();

hashMap.put("status", status);

firebaseFirestore.collection("Users").document(FirebaseAuth.getInstance().getUid()).update(hashMap).addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

// Toast.makeText(MessageActivity.this, "online", Toast.LENGTH\_SHORT).show();

Log.d("STATUSUSER", "USER ONLINE");

}

});

}

}

* + 1. **TimeAgo**

package com.aris.crowdreporting.HelperUtils;

public class TimeAgo {

private static final int SECOND\_MILLIS = 1000;

private static final int MINUTE\_MILLIS = 60 \* SECOND\_MILLIS;

private static final int HOUR\_MILLIS = 60 \* MINUTE\_MILLIS;

private static final int DAY\_MILLIS = 24 \* HOUR\_MILLIS;

public static String getTimeAgo(long time) {

if (time < 1000000000000L) {

time \*= 1000;

}

long now = System.currentTimeMillis();

if (time > now || time <= 0) {

return null;

}

final long diff = now - time;

if (diff < MINUTE\_MILLIS) {

return "just now";

} else if (diff < 2 \* MINUTE\_MILLIS) {

return "a minute ago";

} else if (diff < 50 \* MINUTE\_MILLIS) {

return diff / MINUTE\_MILLIS + " minutes ago";

} else if (diff < 90 \* MINUTE\_MILLIS) {

return "an hour ago";

} else if (diff < 24 \* HOUR\_MILLIS) {

return diff / HOUR\_MILLIS + " hours ago";

} else if (diff < 48 \* HOUR\_MILLIS) {

return "yesterday";

} else {

return diff / DAY\_MILLIS + " days ago";

}

}

}

* + 1. **Client**

package com.aris.crowdreporting.Notifications;

import retrofit2.Retrofit;

import retrofit2.converter.gson.GsonConverterFactory;

public class Client {

private static Retrofit retrofit = null;

public static Retrofit getClient(String url){

if (retrofit == null){

retrofit = new Retrofit.Builder()

.baseUrl(url)

.addConverterFactory(GsonConverterFactory.create())

.build();

}

return retrofit;

}

}

* + 1. **Data**

package com.aris.crowdreporting.Notifications;

public class Data {

private String user;

private int icon;

private String body;

private String title;

private String sented;

public Data() {

}

public Data(String user, int icon, String body, String title, String sented) {

this.user = user;

this.icon = icon;

this.body = body;

this.title = title;

this.sented = sented;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

public int getIcon() {

return icon;

}

public void setIcon(int icon) {

this.icon = icon;

}

public String getBody() {

return body;

}

public void setBody(String body) {

this.body = body;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getSented() {

return sented;

}

public void setSented(String sented) {

this.sented = sented;

}

}

* + 1. **MyFirebaseIsService**

package com.aris.crowdreporting.Notifications;

import com.firebase.client.Firebase;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.iid.FirebaseInstanceId;

import com.google.firebase.iid.FirebaseInstanceIdService;

public class MyFirebaseIdService extends FirebaseInstanceIdService {

@Override

public void onTokenRefresh() {

super.onTokenRefresh();

FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

String refreshToken = FirebaseInstanceId.getInstance().getToken();

if (firebaseUser != null){

updateToken(refreshToken);

}

}

private void updateToken(String refreshToken) {

FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

DocumentReference documentReference = FirebaseFirestore.getInstance().collection("Tokens").document(firebaseUser.getUid());

Token token = new Token(refreshToken);

documentReference.set(token);

}

}

* + 1. **MyFirebaseMessaging**

package com.aris.crowdreporting.Notifications;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.media.RingtoneManager;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.support.v4.app.NotificationCompat;

import com.aris.crowdreporting.Activities.MessageActivity;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.messaging.FirebaseMessagingService;

import com.google.firebase.messaging.RemoteMessage;

public class MyFirebaseMessaging extends FirebaseMessagingService {

@Override

public void onMessageReceived(RemoteMessage remoteMessage) {

super.onMessageReceived(remoteMessage);

String sented = remoteMessage.getData().get("sented");

String user = remoteMessage.getData().get("user");

SharedPreferences preferences = getSharedPreferences("PREFS", MODE\_PRIVATE);

String currentUser = preferences.getString("currentuser", "none");

FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

if (firebaseUser != null && sented.equals(firebaseUser.getUid())){

if (!currentUser.equals(user)) {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

sendOreoNotification(remoteMessage);

} else {

sendNotification(remoteMessage);

}

}

}

}

private void sendOreoNotification(RemoteMessage remoteMessage) {

String user = remoteMessage.getData().get("user");

String icon = remoteMessage.getData().get("icon");

String title = remoteMessage.getData().get("title");

String body = remoteMessage.getData().get("body");

RemoteMessage.Notification notification = remoteMessage.getNotification();

int j = Integer.parseInt(user.replaceAll("[\\D]", ""));

Intent intent = new Intent(this, MessageActivity.class);

Bundle bundle = new Bundle();

bundle.putString("user\_id\_msg", user);

intent.putExtras(bundle);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

PendingIntent pendingIntent = PendingIntent.getActivity(this, j, intent, PendingIntent.FLAG\_ONE\_SHOT);

Uri defaultSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

OreoNotification oreoNotification = new OreoNotification(this);

Notification.Builder builder = oreoNotification.getOreoNotification(title, body, pendingIntent, defaultSound, icon);

int i = 0;

if (j > 0){

i = j;

}

oreoNotification.getManager().notify(i, builder.build());

}

private void sendNotification(RemoteMessage remoteMessage) {

String user = remoteMessage.getData().get("user");

String icon = remoteMessage.getData().get("icon");

String title = remoteMessage.getData().get("title");

String body = remoteMessage.getData().get("body");

RemoteMessage.Notification notification = remoteMessage.getNotification();

int j = Integer.parseInt(user.replaceAll("[\\D]", ""));

Intent intent = new Intent(this, MessageActivity.class);

Bundle bundle = new Bundle();

bundle.putString("user\_id\_msg", user);

intent.putExtras(bundle);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

PendingIntent pendingIntent = PendingIntent.getActivity(this, j, intent, PendingIntent.FLAG\_ONE\_SHOT);

Uri defaultSound = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

NotificationCompat.Builder builder = new NotificationCompat.Builder(this)

.setSmallIcon(Integer.parseInt(icon))

.setContentTitle(title)

.setContentText(body)

.setAutoCancel(true)

.setSound(defaultSound)

.setContentIntent(pendingIntent);

NotificationManager notif = (NotificationManager)getSystemService(Context.NOTIFICATION\_SERVICE);

int i = 0;

if (j > 0){

i = j;

}

notif.notify(i, builder.build());

}

}

* + 1. **MyResponse**

package com.aris.crowdreporting.Notifications;

public class MyResponse {

public int success;

}

**OreoNotification**

package com.aris.crowdreporting.Notifications;

import android.annotation.TargetApi;

import android.app.Notification;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.ContextWrapper;

import android.net.Uri;

import android.os.Build;

public class OreoNotification extends ContextWrapper {

private static final String CHANNEL\_ID = "com.aris.crowdreporting";

private static final String CHANNEL\_NAME = "crowdreporting";

private NotificationManager notificationManager;

public OreoNotification(Context base) {

super(base);

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O){

createChannel();

}

}

@TargetApi(Build.VERSION\_CODES.O)

private void createChannel() {

NotificationChannel channel = new NotificationChannel(CHANNEL\_ID, CHANNEL\_NAME, NotificationManager.IMPORTANCE\_DEFAULT);

channel.enableLights(false);

channel.enableVibration(true);

channel.setLockscreenVisibility(Notification.VISIBILITY\_PRIVATE);

getManager().createNotificationChannel(channel);

}

public NotificationManager getManager(){

if (notificationManager == null){

notificationManager = (NotificationManager)getSystemService(Context.NOTIFICATION\_SERVICE);

}

return notificationManager;

}

@TargetApi(Build.VERSION\_CODES.O)

public Notification.Builder getOreoNotification(String title, String body,

PendingIntent pendingIntent, Uri soundUri, String icon){

return new Notification.Builder(getApplicationContext(), CHANNEL\_ID)

.setContentIntent(pendingIntent)

.setContentTitle(title)

.setContentText(body)

.setSmallIcon(Integer.parseInt(icon))

.setSound(soundUri)

.setAutoCancel(true);

}

}

* + 1. **Sender**

package com.aris.crowdreporting.Notifications;

public class Sender {

public Data data;

public String to;

public Sender(Data data, String to) {

this.data = data;

this.to = to;

}

}

* + 1. **Token**

package com.aris.crowdreporting.Notifications;

public class Token {

private String token;

public Token() {

}

public Token(String token) {

this.token = token;

}

public String getToken() {

return token;

}

public void setToken(String token) {

this.token = token;

}

}

* + 1. **APIService**

package com.aris.crowdreporting;

import com.aris.crowdreporting.Notifications.MyResponse;

import com.aris.crowdreporting.Notifications.Sender;

import retrofit2.Call;

import retrofit2.http.Body;

import retrofit2.http.Headers;

import retrofit2.http.POST;

public interface APIService {

@Headers(

{

"Content-Type:application/json",

"Authorization:key=AAAAstsgwEw:APA91bGNtCGpdpDeQN6OgJ6uoAS-v9TNC18aFtNM35lEqppA9KbCcQylAgfCWONyWmeiE92HOT0WJHEI8-GAFxwhzp6rmMtOB-pKqAjTQBxYGo20RWt2-ixugU-EvhZGFegemPvf6UV6"

}

)

@POST("fcm/send")

Call<MyResponse> sendNotification(@Body Sender body);

}