

```
package com.example.covid_19alertapp.activities;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Parcelable;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.example.covid_19alertapp.R;
import com.example.covid_19alertapp.adapters.CommentFeedAdapter;
import com.example.covid_19alertapp.adapters.FeedAdapter;
import com.example.covid_19alertapp.extras.Constants;
import com.example.covid_19alertapp.extras.DateTimeHandler;
import com.example.covid_19alertapp.models.Comment;
import com.google.android.gms.tasks.OnFailureListener;
```

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

import com.google.android.material.floatingactionbutton.FloatingActionButton;

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.MutableData;

import com.google.firebase.database.Transaction;

import com.google.firebase.database.ValueEventListener;


import java.util.ArrayList;
```

```
public class CommentFeedActivity extends AppCompatActivity {
```

```
    AlertDialog dialogBuilder;

    FloatingActionButton floatingActionButton;

    EditText comment_editText;

    Button btn_cancel, btn_submit, btn_home;

    TextView post_body, post_auth, post_datetime, comment_bar;

    SharedPreferences sharedPreferences;

    DatabaseReference view_reference, comment_reference, cCount_reference;

    RecyclerView recyclerView;

    ArrayList<Comment> commentList;

    RelativeLayout relativeLayout;

    CommentFeedAdapter commentFeedAdapter;
```

```

private Parcelable recyclerViewState;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.comment_feed_activity);

    comment_reference =
FirebaseDatabase.getInstance().getReference().child("Posts").child(FeedAdapter.POST.getPostID());

    recyclerView = findViewById(R.id.recyclerView_comment_feed);

    relativeLayout = findViewById(R.id.relief_line_commentfeed);

    btn_home = findViewById(R.id.home_button_comment_feed);

    post_body = findViewById(R.id.description_Text_comment_feed);

    post_auth = findViewById(R.id.textView_username_comment_feed);

    post_datetime = findViewById(R.id.textView_DatenTime_comment_feed);

    comment_bar = findViewById(R.id.comment_bar);

    //Set Post

    post_body.setText(FeedAdapter.POST.getPostBody());

    post_auth.setText(FeedAdapter.POST.getUserName());

    if(FeedAdapter.POST.getPostType().equals("RELIEF"))
    {

```

```
        RelativeLayout.setBackgroundColor(ContextCompat.getColor(getApplicationContext(),
R.color.color_item_view_relief_line));
```

```
    }
```

```
String post_date = FeedAdapter.POST.getDate();
```

```
if(post_date.equals(DateTimeHandler.DateToday()))
```

```
{
```

```
    post_date = "Today";
```

```
}
```

```
String post_date_time = post_date + " " + FeedAdapter.POST.getTime();
```

```
post_datetime.setText(post_date_time);
```

```
//Create Comment
```

```
floatingActionButton = findViewById(R.id.floatingActionButton_comment_feed);
```

```
dialogBuilder = new AlertDialog.Builder(this).create();
```

```
LayoutInflater inflater = this.getLayoutInflater();
```

```
View dialogView = inflater.inflate(R.layout.post_comment_view, null);
```

```
btn_submit = dialogView.findViewById(R.id.btn_submit_comment);
```

```
btn_cancel = dialogView.findViewById(R.id.btn_cancel_comment);
```

```
comment_editText = dialogView.findViewById(R.id.editText_comment);
```

```
dialogBuilder.setView(dialogView);
```

```
btn_cancel.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
public void onClick(View v) {  
    comment_editText.setText("");  
    dialogBuilder.dismiss();  
}  
});
```

```
btn_home.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        finish();  
    }  
});
```

```
btn_submit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        //Write Post Comment Function  
  
        sharedPreferences =  
getSharedPreferences(Constants.USER_INFO_SHARED_PREFERENCES,MODE_PRIVATE);  
  
        String user_name = sharedPreferences.getString(Constants.username_preference,null);  
  
        String user_ID = sharedPreferences.getString(Constants.uid_preference,null);  
  
        String comment_ID = comment_reference.push().getKey();  
  
  
        Comment comment = new Comment();
```

```

comment.setComment_date(DateTimeHandler.DateToday());

comment.setComment_text(comment_editText.getText().toString());

comment.setComment_time(DateTimeHandler.TimeNow());

comment.setPost_id(FeedAdapter.POST.getPostID());

comment.setUser_name(user_name);

comment.setUser_id(user_ID);

comment.setComment_id(comment_ID);

if(FeedAdapter.POST.getPostType().equals("RELIEF"))
{
    if(FeedAdapter.POST.getUserID().equals(user_ID))
    {
        comment.setUser_name("Anonymous");
    }
}

comment_reference.child("Comments").child(comment_ID).setValue(comment)

    .addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

    increaseCommentCount();

    comment_editText.setText("");

    dialogBuilder.dismiss();

}

}).addOnFailureListener(new OnFailureListener() {

```

```

        @Override

        public void onFailure(@NonNull Exception e) {

            Toast.makeText(getApplicationContext(),"Failed: Check Internet
Connection",Toast.LENGTH_SHORT);

        }

    });

}

});

```

```

floatingActionButton.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        dialogBuilder.show();

    }

});

```

```

//RecyclerView

LinearLayoutManager manager = new LinearLayoutManager(this);

recyclerView.setLayoutManager(manager);

recyclerView.setHasFixedSize(true);

view_reference =
FirebaseDatabase.getInstance().getReference().child("Posts").child(FeedAdapter.POST.getPostID()).child
("Comments");

view_reference.addValueEventListener(new ValueEventListener() {

```

```

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

    commentList = new ArrayList<>();

    for(DataSnapshot dataSnapshot1:dataSnapshot.getChildren())
    {

        Comment comment = dataSnapshot1.getValue(Comment.class);

        commentList.add(comment);

    }

    if(commentList.isEmpty()) {comment_bar.setText("No Comments Yet");}

    //Collections.reverse(commentList);

    commentFeedAdapter = new CommentFeedAdapter(getApplicationContext(),commentList);

    recyclerViewState = recyclerView.getLayoutManager().onSaveInstanceState();

    recyclerView.setAdapter(commentFeedAdapter);

    recyclerView.getLayoutManager().onRestoreInstanceState(recyclerViewState);

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

    Toast.makeText(getApplicationContext(),"Something Wrong!\nCheck Internet Connection.",
    Toast.LENGTH_LONG).show();

}

});

```



```

    }

    public void increaseCommentCount()

    {

        cCount_reference =
        FirebaseDatabase.getInstance().getReference("Posts").child(FeedAdapter.POST.getPostID()).child("commentCount");

        cCount_reference.runTransaction(new Transaction.Handler() {

            @Override

            public Transaction.Result doTransaction(MutableData mutableData) {

                Integer score = mutableData.getValue(Integer.class);

                if (score == null) { return Transaction.success(mutableData); }

                mutableData.setValue(score + 1);

                return Transaction.success(mutableData);

            }

            @Override

            public void onComplete(DatabaseError databaseError, boolean b, DataSnapshot dataSnapshot) {}

        });

    }

}

```