**COMP1786 Logbook**

1. **Basic Information**

|  |  |
| --- | --- |
| 1.1 Student name | Bui Duy Hung |
| 1.2 Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work. | **Name:**  **Login id:** |
| 1.3 Which Exercise is this? Tick as appropriate. | * Exercise 1 * Exercise 2 * Exercise 3 |
| 1.4 How well did you complete the exercise? Tick as appropriate. | * I tried but couldn't complete it * I did it but I feel I should have done better * I did everything that was asked * I did more than was asked for |
| 1.5 Briefly explain your answer to question 1.4  Without any explanation/justification, your scores will be deducted. | I have completed the three exercises above, but there are still many points that need to be improved. Exercise 1, I was able to calculate the conversion from one unit to another, but there is a little problem that when randomly converting the first unit, it does not give the calculation based on the available value. I think it is due to the project's calculation logic and partly because the "switch-case" is not optimized, so I will find a solution to improve it as soon as possible. As for exercises 2 and 3, I need to improve the user interface a bit more to attract users and add more features to further perfect the app. Through the 3 exercises above, I have learned a lot about how to build a basic app on Android Studio using Java programming language. |

1. **Exercise answer**

**2.1 Screen shots demonstrating what you achieved**

a. Unit Converter Application

First will be the main interface of the app:

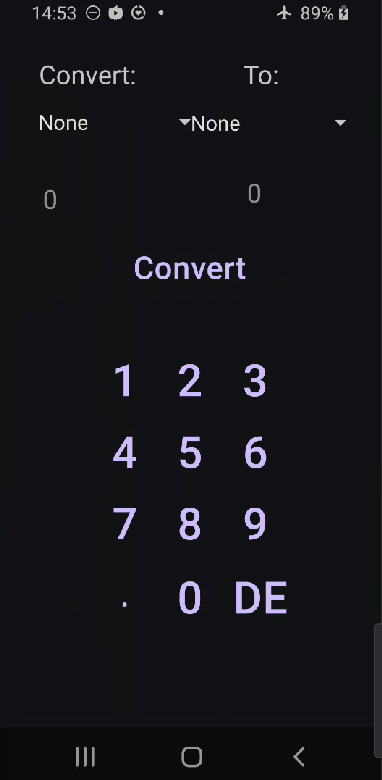


Figure 1: Unit Converter Application

All “units” input:



Figure 2: All "Units" Input

All “units” output:



Figure 3: All "units" output

Case 1 the application will output 0 if the "input" or "output" fields of "unit" are "none"

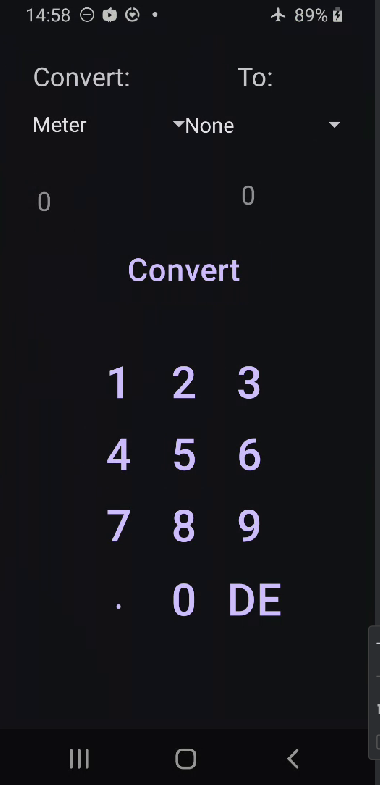


Figure 4: Case 1 Unit Converter App

In case 2 applications will give the result as the input value if 2 "units" are the same:

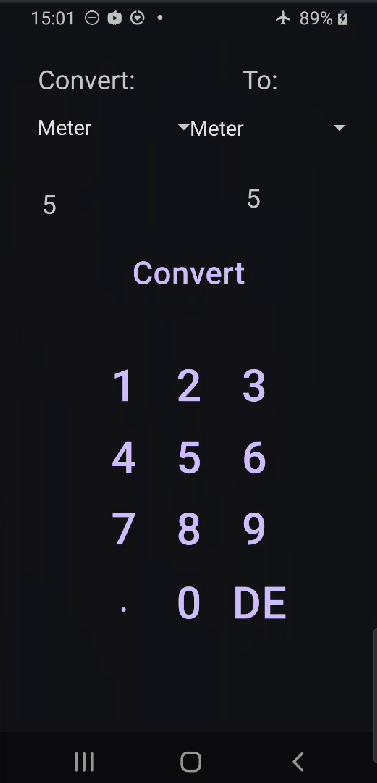


Figure 5: Case 3 Unit Converter App

Case 3 if the two "units" are different:

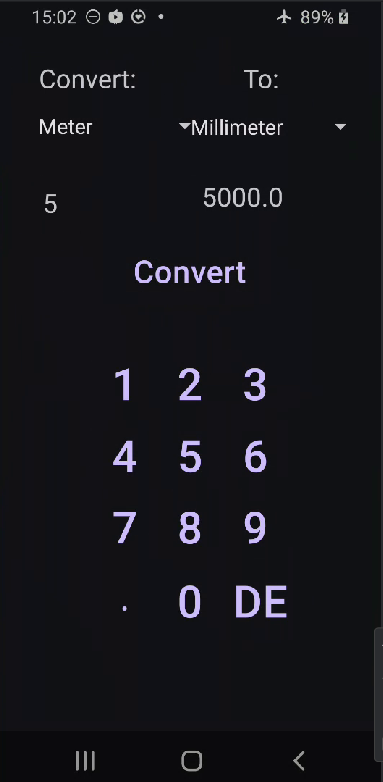


Figure 6: Case 3 Unit Converter App

b. Todo list application (Exercise 2 + 3)

In this exercise, we will develop a "To-do list" application with functions such as adding, editing, deleting, and viewing details of "tasks" in the application. The "tasks" will have two main fields: "title" and "description".

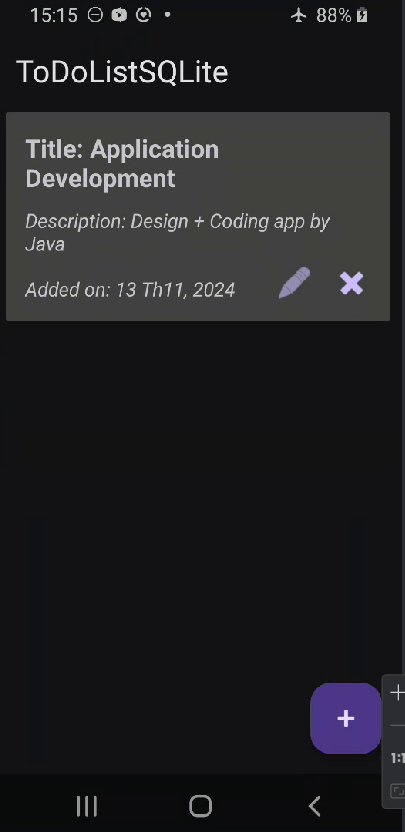


Figure 7: To-do list app main interface

To add a "task", click on the "+" sign that appears on the screen and a popup window will appear to add data:

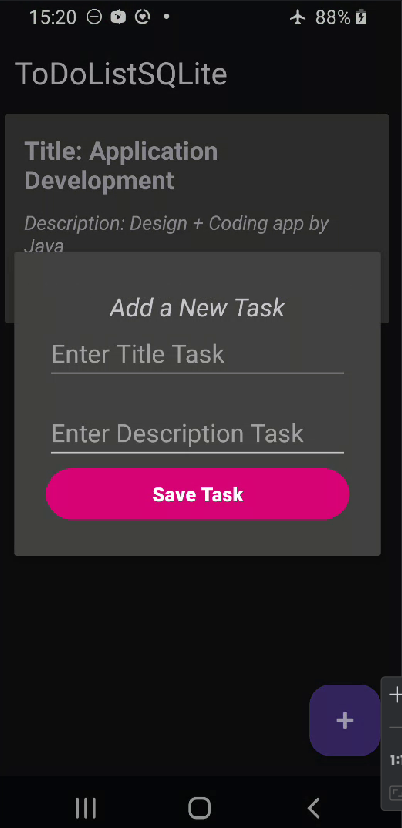


Figure 8: UI add task

After adding data, there will be a list of "tasks" as follows:

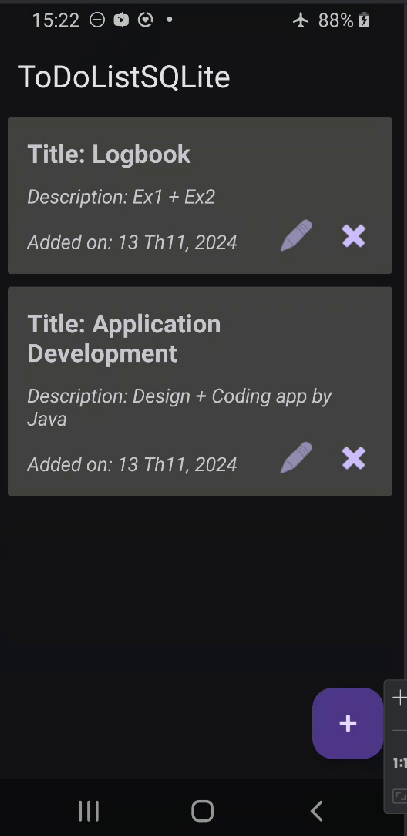


Figure 9: UI list task

Click on the "pencil" icon button to update existing data as below:

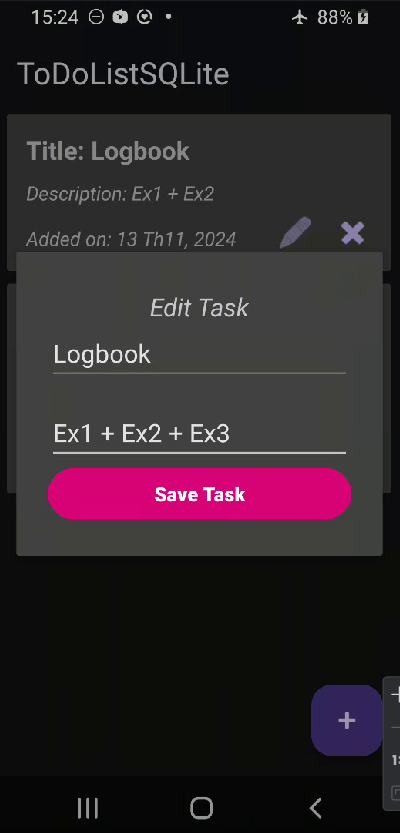


Figure 10: Update data

Data after update:

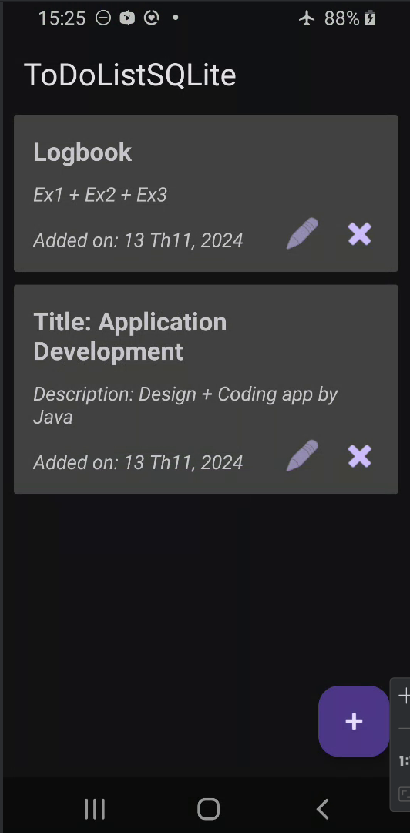


Figure 11: List task after update

Click the "X" button to delete the data and it will bring up a popup asking if you are sure you want to delete the data:

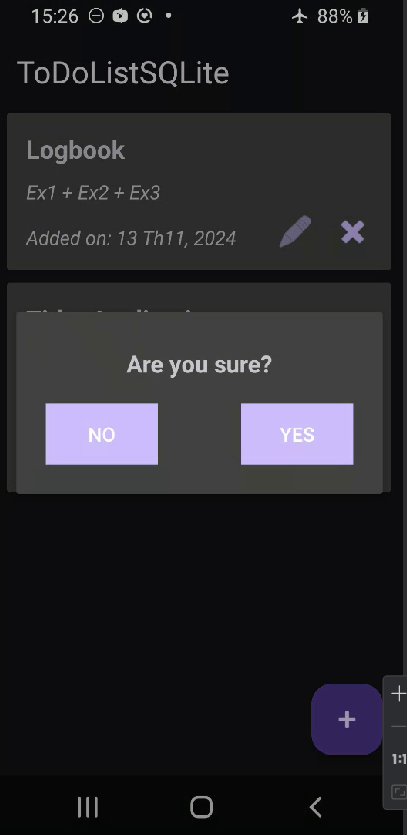


Figure 12: UI delete task

List of data after deletion:

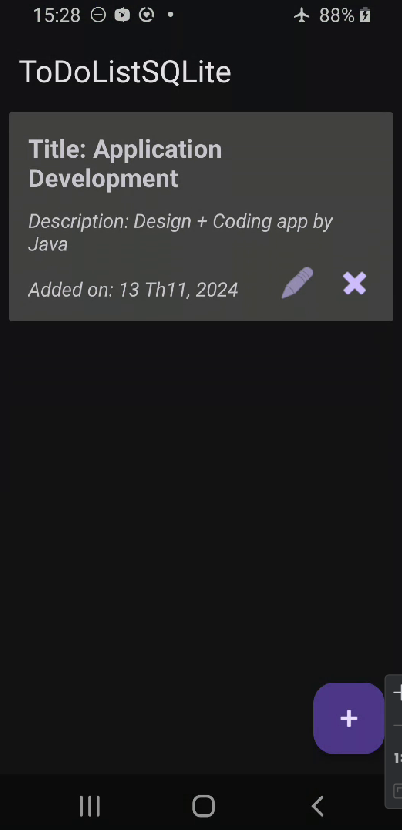


Figure 13: List task after delete

Database SQLite in “To-do list” app:

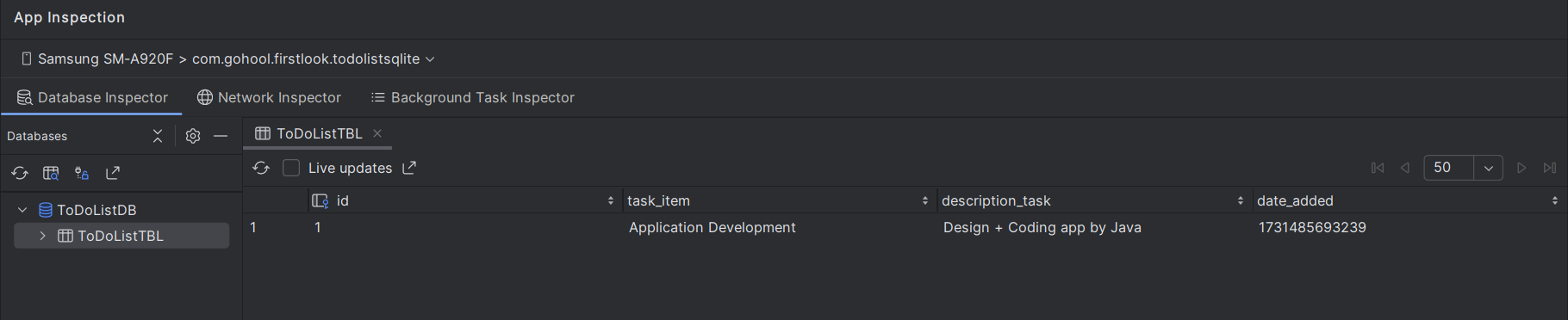


Figure 14: Database SQLite To-do list app

**2.2 Code that you wrote**

1. Unit Converter App

* Layout: activity\_main.xml: This is the interface code which includes "views", "spinners" to display the list of "units", "buttons" to calculate or enter numbers.
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:id="@+id/main"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   tools:context=".MainActivity">  
    
   <Button  
   android:id="@+id/btn\_convert"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="157dp"  
   android:layout\_height="62dp"  
   android:text="Convert"  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.323" />  
    
   <Spinner  
   android:id="@+id/spinnerFrom"  
   android:layout\_width="154dp"  
   android:layout\_height="48dp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.151"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.103" />  
    
   <TextView  
   android:id="@+id/textView"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Convert: "  
   android:textSize="20sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.144"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.044"  
   tools:ignore="MissingConstraints" />  
    
   <TextView  
   android:id="@+id/inputView"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_marginTop="32dp"  
   android:hint="0"  
   android:textSize="20sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.119"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView"  
   app:layout\_constraintVertical\_bias="0.091" />  
    
   <TextView  
   android:id="@+id/textView3"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="To:"  
   android:textSize="20sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.705"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.044" />  
    
   <TextView  
   android:id="@+id/outputView"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:hint="0"  
   android:textSize="20sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.675"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toBottomOf="@+id/textView3"  
   app:layout\_constraintVertical\_bias="0.138" />  
    
   <Button  
   android:id="@+id/btn\_2"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="2"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.501"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <Button  
   android:id="@+id/btn\_7"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="7"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.278"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.735" />  
    
   <Button  
   android:id="@+id/btn\_9"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="9"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.724"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.735" />  
    
   <Button  
   android:id="@+id/btn\_delete"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="90dp"  
   android:layout\_height="75dp"  
   android:text="DEL"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.757"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.856" />  
    
   <Button  
   android:id="@+id/btn\_5"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="5"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.501"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.618" />  
    
   <Button  
   android:id="@+id/btn\_6"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="6"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.724"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.618" />  
    
   <Button  
   android:id="@+id/btn\_0"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="0"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.501"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.856" />  
    
   <Button  
   android:id="@+id/btn\_8"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="8"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.501"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.735" />  
    
   <Button  
   android:id="@+id/btn\_1"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="1"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.278"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <Button  
   android:id="@+id/btn\_4"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="4"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.278"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.618" />  
    
   <Button  
   android:id="@+id/btn\_3"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="3"  
   android:textSize="34sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.724"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />  
    
   <Spinner  
   android:id="@+id/spinnerTo"  
   android:layout\_width="157dp"  
   android:layout\_height="49dp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.937"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.103" />  
    
   <Button  
   android:id="@+id/btn\_dot"  
   style="@style/Widget.Material3.Button.IconButton"  
   android:layout\_width="75dp"  
   android:layout\_height="75dp"  
   android:text="."  
   android:textSize="24sp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintHorizontal\_bias="0.278"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent"  
   app:layout\_constraintVertical\_bias="0.856" />  
    
  </androidx.constraintlayout.widget.ConstraintLayout>

* MainActivity.java: This is the main code that sets up the logic of the post. It includes declaring and initializing the buttons, setting up the array elements in the "spinner", and converting from one "unit" to another.
* package com.gohool.firstlook.unitconvertapp;  
    
  import android.os.Bundle;  
  import android.view.View;  
  import android.widget.AdapterView;  
  import android.widget.ArrayAdapter;  
  import android.widget.Button;  
  import android.widget.Spinner;  
  import android.widget.TextView;  
  import android.widget.Toast;  
    
  import androidx.activity.EdgeToEdge;  
  import androidx.appcompat.app.AppCompatActivity;  
  import androidx.core.graphics.Insets;  
  import androidx.core.view.ViewCompat;  
  import androidx.core.view.WindowInsetsCompat;  
    
  public class MainActivity extends AppCompatActivity {  
    
   private String number = null;  
    
   // Declare buttons  
   private Button btn\_1, btn\_2, btn\_3, btn\_4, btn\_5, btn\_6, btn\_7,  
   btn\_8, btn\_9, btn\_0, btn\_dot, btn\_delete, btn\_convert;  
    
   // Declare text views  
   private TextView inputView, outputView;  
    
   // Declare spinners  
   private Spinner from;  
   private Spinner to;  
   ArrayAdapter adapter;  
    
   @Override  
   protected void onCreate(Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
   EdgeToEdge.*enable*(this);  
   setContentView(R.layout.*activity\_main*);  
    
   // Initialize button  
   btn\_1 = findViewById(R.id.*btn\_1*);  
   btn\_2 = findViewById(R.id.*btn\_2*);  
   btn\_3 = findViewById(R.id.*btn\_3*);  
   btn\_4 = findViewById(R.id.*btn\_4*);  
   btn\_5 = findViewById(R.id.*btn\_5*);  
   btn\_6 = findViewById(R.id.*btn\_6*);  
   btn\_7 = findViewById(R.id.*btn\_7*);  
   btn\_8 = findViewById(R.id.*btn\_8*);  
   btn\_9 = findViewById(R.id.*btn\_9*);  
   btn\_0 = findViewById(R.id.*btn\_0*);  
   btn\_dot = findViewById(R.id.*btn\_dot*);  
   btn\_delete = findViewById(R.id.*btn\_delete*);  
   btn\_convert = findViewById(R.id.*btn\_convert*);  
    
   // Initialize text views  
   inputView = findViewById(R.id.*inputView*);  
   outputView = findViewById(R.id.*outputView*);  
    
   // Initialize spinners  
   from = (Spinner) findViewById(R.id.*spinnerFrom*);  
   adapter = ArrayAdapter.*createFromResource*(this, R.array.*spinner*, android.R.layout.*simple\_spinner\_item*);  
   adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
   from.setAdapter(adapter);  
    
   to = (Spinner) findViewById(R.id.*spinnerTo*);  
   adapter = ArrayAdapter.*createFromResource*(this, R.array.*spinner*, android.R.layout.*simple\_spinner\_item*);  
   adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
   to.setAdapter(adapter);  
    
   // OnClick button  
   btn\_1.setOnClickListener(v -> numberClick("1"));  
   btn\_2.setOnClickListener(v -> numberClick("2"));  
   btn\_3.setOnClickListener(v -> numberClick("3"));  
   btn\_4.setOnClickListener(v -> numberClick("4"));  
   btn\_5.setOnClickListener(v -> numberClick("5"));  
   btn\_6.setOnClickListener(v -> numberClick("6"));  
   btn\_7.setOnClickListener(v -> numberClick("7"));  
   btn\_8.setOnClickListener(v -> numberClick("8"));  
   btn\_9.setOnClickListener(v -> numberClick("9"));  
   btn\_0.setOnClickListener(v -> numberClick("0"));  
   btn\_dot.setOnClickListener(v -> numberClick("."));  
    
   btn\_delete.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   number = number.substring(0, number.length() - 1);  
   inputView.setText(number);  
   }  
   });  
    
    
   // Convert button  
    
   from.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
   @Override  
   public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
   String selectItemFrom = parent.getItemAtPosition(position).toString();  
    
   to.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
   @Override  
   public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
   String selectItemTo = parent.getItemAtPosition(position).toString();  
    
   btn\_convert.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   double input;  
   double output;  
    
   switch (selectItemFrom)  
   {  
   case "None":  
   outputView.setText("0");  
   break;  
    
   case "Meter":  
   switch (selectItemTo)  
   {  
   case "None":  
   outputView.setText("0");  
   break;  
    
   case "Meter":  
   outputView.setText(inputView.getText());  
   break;  
    
   case "Millimeter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1000;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Mile":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.000621371;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Foot":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 3.28084;  
   outputView.setText(String.*valueOf*(output));  
   break;  
   }  
   break;  
    
   case "Millimeter":  
   switch (selectItemTo)  
   {  
   case "None":  
   outputView.setText("0");  
   break;  
    
   case "Meter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.001;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Millimeter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Mile":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.00000621371;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Foot":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.00328084;  
   outputView.setText(String.*valueOf*(output));  
   break;  
   }  
   break;  
    
   case "Mile":  
   switch (selectItemTo)  
   {  
   case "None":  
   outputView.setText("0");  
   break;  
    
   case "Meter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1609;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Millimeter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1609344;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Mile":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Foot":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 5280;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   }  
   break;  
    
   case "Foot":  
   switch (selectItemTo)  
   {  
   case "None":  
   outputView.setText("0");  
   break;  
    
   case "Meter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.3048;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Millimeter":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 304.8;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Mile":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 0.000189394;  
   outputView.setText(String.*valueOf*(output));  
   break;  
    
   case "Foot":  
   input = Double.*parseDouble*(inputView.getText().toString());  
   output = input \* 1;  
   outputView.setText(String.*valueOf*(output));  
   break;  
   }  
   break;  
   }  
   }  
   });  
    
   }  
    
    
   @Override  
   public void onNothingSelected(AdapterView<?> parent) {  
    
   }  
   });  
   }  
    
   @Override  
   public void onNothingSelected(AdapterView<?> parent) {  
    
   }  
   });  
    
   ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
   Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
   v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
   return insets;  
   });  
   }  
    
   // Click number  
   public void numberClick(String view)  
   {  
   if (number == null)  
   {  
   number = view;  
   }  
   else  
   {  
   number += view;  
   }  
   inputView.setText(number);  
   }  
  }

b. To-do list application

- CreateDB.java: This is the code to create the database table form

package com.gohool.firstlook.todolistsqlite.Util;  
  
// Database table form  
public class CreateDB  
{  
 public static final int *DB\_VERSION* = 1;  
 public static final String *DB\_NAME* = "ToDoListDB";  
 public static final String *TABLE\_NAME* = "ToDoListTBL";  
  
 // Table columns  
 public static final String *KEY\_ID* = "id";  
 public static final String *KEY\_TASK\_ITEM* = "task\_item";  
 public static final String *KEY\_DESCRIPTION\_TASK* = "description\_task";  
 public static final String *KEY\_DATE\_NAME* = "date\_added";  
}

* RecyclerViewAdapter.java: This is the code to connect data to user interface components such as ListView, GridView, Spinner and RecyclerView to allow you to display data from different sources (such as arrays, lists, databases) to these components in a flexible and efficient way.
* public class RecycleViewAdapter extends RecyclerView.Adapter<RecycleViewAdapter.ViewHolder>  
  {  
   // Declare variables  
   private Context context;  
   private List<Task> taskList;  
   private AlertDialog.Builder alertDialogBuilder;  
   private AlertDialog dialog;  
   private LayoutInflater inflater;  
    
   // Constructor  
   public RecycleViewAdapter(Context context, List<Task> taskList) {  
   this.context = context;  
   this.taskList = taskList;  
   }  
    
   // Override methods  
   @NonNull  
   @Override  
   public RecycleViewAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
   View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*list\_row*, parent, false);  
   return new ViewHolder(view, context);  
   }  
    
   @Override  
   public void onBindViewHolder(@NonNull RecycleViewAdapter.ViewHolder holder, int position) {  
   Task task = taskList.get(position);  
   holder.titleTask.setText(task.getTitle());  
   holder.descriptionTask.setText(task.getDescription());  
   holder.dateAdded.setText(task.getDateItemAdded());  
   }  
    
   @Override  
   public int getItemCount() {  
   return taskList.size();  
   }  
    
   // ViewHolder class  
   public class ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener  
   {  
   public int id;  
   public TextView titleTask;  
   public TextView descriptionTask;  
   public TextView dateAdded;  
   public Button editButton;  
   public Button deleteButton;  
    
   public ViewHolder(@NonNull View view, Context ctx) {  
   super(view);  
   context = ctx;  
    
   // Initialize views  
   titleTask = (TextView) view.findViewById(R.id.*titleTask*);  
   descriptionTask = (TextView) view.findViewById(R.id.*descriptionTask*);  
   dateAdded = (TextView) view.findViewById(R.id.*dateAdded*);  
    
   editButton = (Button) view.findViewById(R.id.*editButton*);  
   deleteButton = (Button) view.findViewById(R.id.*deleteButton*);  
    
   editButton.setOnClickListener(this);  
   deleteButton.setOnClickListener(this);  
    
   // Set onclick listener for each item  
   view.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   // Go to next screen DetailActivity  
   int position = getAdapterPosition();  
    
   Task task = taskList.get(position);  
   Intent intent = new Intent(context, DetailActivity.class);  
   intent.putExtra("title", task.getTitle());  
   intent.putExtra("description", task.getDescription());  
   intent.putExtra("dateAdded", task.getDateItemAdded());  
   intent.putExtra("id", task.getId());  
   context.startActivity(intent);  
   }  
   });  
   }  
    
   // Handle click events  
   @Override  
   public void onClick(View v) {  
   int position = getAdapterPosition();  
   Task task = taskList.get(position);  
   if (v.getId() == R.id.*editButton*) {  
   editTask(task);  
   }  
    
   if (v.getId() == R.id.*deleteButton*) {  
   deleteTask(task.getId());  
   }  
   }  
    
   // Delete task  
   public void deleteTask(int id)  
   {  
   // create an AlertDialog  
   alertDialogBuilder = new AlertDialog.Builder(context);  
    
   inflater = LayoutInflater.*from*(context);  
   View view = inflater.inflate(R.layout.*confirm\_dialog*, null);  
    
   Button noButton = (Button) view.findViewById(R.id.*noButton*);  
   Button yesButton = (Button) view.findViewById(R.id.*yesButton*);  
    
   alertDialogBuilder.setView(view);  
   dialog = alertDialogBuilder.create();  
   dialog.show();  
    
   noButton.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   dialog.dismiss();  
   }  
   });  
    
   yesButton.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   // delete task  
   DatabaseHandle db = new DatabaseHandle(context);  
   db.deleteTask(id);  
   taskList.remove(getAdapterPosition());  
   notifyItemRemoved(getAdapterPosition());  
    
   dialog.dismiss();  
    
   }  
   });  
   }  
    
   // Edit task  
   public void editTask(Task task)  
   {  
   alertDialogBuilder = new AlertDialog.Builder(context);  
    
   inflater = LayoutInflater.*from*(context);  
   View view = inflater.inflate(R.layout.*popup*, null);  
    
   final EditText taskTitle = (EditText) view.findViewById(R.id.*taskItem*);  
   final EditText descriptionTask = (EditText) view.findViewById(R.id.*descriptionTask*);  
   final TextView title = (TextView) view.findViewById(R.id.*titleTask*);  
    
   title.setText("Edit Task");  
   Button saveButton = (Button) view.findViewById(R.id.*saveButton*);  
    
   alertDialogBuilder.setView(view);  
   dialog = alertDialogBuilder.create();  
   dialog.show();  
    
   saveButton.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   DatabaseHandle db = new DatabaseHandle(context);  
    
   // Update task  
   task.setTitle(taskTitle.getText().toString());  
   task.setDescription(descriptionTask.getText().toString());  
    
   if (!taskTitle.getText().toString().isEmpty() &&  
   !descriptionTask.getText().toString().isEmpty()) {  
   db.updateTask(task);  
   notifyItemChanged(getAdapterPosition(), task);  
   }  
   else {  
   Toast.*makeText*(context, "Task cannot be blank", Toast.*LENGTH\_LONG*).show();  
   }  
    
   dialog.dismiss();  
   }  
   });  
   }  
    
   }  
  }

- Task.java: This is the code that represents the application's data layer. It contains properties and methods to store and manage the application's data.

// Model Task  
public class Task  
{  
 private int id;  
 private String title;  
 private String description;  
 private String dateItemAdded;  
  
 public Task() {  
 }  
  
 public Task(int id, String title, String description) {  
 this.id = id;  
 this.title = title;  
 this.description = description;  
 this.dateItemAdded = dateItemAdded;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public void setDescription(String description) {  
 this.description = description;  
 }  
  
 public String getDateItemAdded() {  
 return dateItemAdded;  
 }  
  
 public void setDateItemAdded(String dateItemAdded) {  
 this.dateItemAdded = dateItemAdded;  
 }  
}

* DatabseHandle.java: This is the code to create a database table based on the previously written form and includes features to add, edit, and delete data when entering it into the database.
* public class DatabaseHandle extends SQLiteOpenHelper  
  {  
   private Context ctx;  
    
   // Constructor  
   public DatabaseHandle(@Nullable Context context) {  
   super(context, CreateDB.*DB\_NAME*, null, CreateDB.*DB\_VERSION*);  
   this.ctx = context;  
   }  
    
   // Create table  
   @Override  
   public void onCreate(SQLiteDatabase db) {  
   String CREATE\_TODOLIST\_TABLE = "CREATE TABLE " + CreateDB.*TABLE\_NAME* + "("  
   + CreateDB.*KEY\_ID* + " INTEGER PRIMARY KEY,"  
   + CreateDB.*KEY\_TASK\_ITEM* + " TEXT,"  
   + CreateDB.*KEY\_DESCRIPTION\_TASK* + " TEXT,"  
   + CreateDB.*KEY\_DATE\_NAME* + " LONG);";  
    
   db.execSQL(CREATE\_TODOLIST\_TABLE);  
   }  
    
   // Upgrade table  
   @Override  
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
   db.execSQL("DROP TABLE IF EXISTS " + CreateDB.*TABLE\_NAME*);  
   onCreate(db);  
   }  
    
   /\*  
   CRUD Operations (Create, Read, Update, Delete)  
   \*/  
    
   // Add task  
   public void addTask(Task task)  
   {  
   SQLiteDatabase db = this.getWritableDatabase();  
    
   ContentValues values = new ContentValues();  
   values.put(CreateDB.*KEY\_TASK\_ITEM*, task.getTitle());  
   values.put(CreateDB.*KEY\_DESCRIPTION\_TASK*, task.getDescription());  
   values.put(CreateDB.*KEY\_DATE\_NAME*, java.lang.System.*currentTimeMillis*());  
    
   // Insert the row  
   db.insert(CreateDB.*TABLE\_NAME*, null, values);  
    
   Log.*d*("Saved!!", "Saved to DB");  
   }  
    
   // Get a task  
   public Task getTask(int id)  
   {  
   SQLiteDatabase db = this.getWritableDatabase();  
    
   Cursor cursor = db.query(CreateDB.*TABLE\_NAME*, new String[] {CreateDB.*KEY\_ID*,  
   CreateDB.*KEY\_TASK\_ITEM*, CreateDB.*KEY\_DESCRIPTION\_TASK*,  
   CreateDB.*KEY\_DATE\_NAME*},  
   CreateDB.*KEY\_ID* + "=?",  
   new String[] {String.*valueOf*(id)}, null, null, null, null);  
    
   Task task = new Task();  
   if (cursor != null)  
   {  
   cursor.moveToFirst();  
    
   // Get task  
   int idIndex = cursor.getColumnIndex(CreateDB.*KEY\_ID*);  
   if (idIndex >= 0)  
   {  
   task.setId(Integer.*parseInt*(cursor.getString(idIndex)));  
   }  
    
   int titleIndex = cursor.getColumnIndex(CreateDB.*KEY\_TASK\_ITEM*);  
   if (titleIndex >= 0) {  
   task.setTitle(cursor.getString(titleIndex));  
   }  
    
   int descriptionIndex = cursor.getColumnIndex(CreateDB.*KEY\_DESCRIPTION\_TASK*);  
   if (descriptionIndex >= 0) {  
   task.setDescription(cursor.getString(descriptionIndex));  
   }  
    
   // convert timestamp to something readable  
   int dateIndex = cursor.getColumnIndex(CreateDB.*KEY\_DATE\_NAME*);  
   if (dateIndex >= 0) {  
   java.text.DateFormat dateFormat = java.text.DateFormat.*getDateInstance*();  
   String formattedDate = dateFormat.format(new Date(cursor.getLong(dateIndex)).getTime());  
   task.setDateItemAdded(formattedDate);  
   }  
   }  
   return task;  
   }  
    
   // Get all tasks  
   public List<Task> getAllTasks()  
   {  
   SQLiteDatabase db = this.getReadableDatabase();  
    
   List<Task> taskList = new ArrayList<>();  
    
   Cursor cursor = db.query(CreateDB.*TABLE\_NAME*, new String[] {  
   CreateDB.*KEY\_ID*, CreateDB.*KEY\_TASK\_ITEM*, CreateDB.*KEY\_DESCRIPTION\_TASK*, CreateDB.*KEY\_DATE\_NAME*},  
   null, null, null, null, CreateDB.*KEY\_DATE\_NAME* + " DESC", null); // ASC: ascending, DESC: descending  
    
   if (cursor.moveToFirst())  
   {  
   do {  
   Task task = new Task();  
   int idIndex = cursor.getColumnIndex(CreateDB.*KEY\_ID*);  
   if (idIndex >= 0)  
   {  
   task.setId(Integer.*parseInt*(cursor.getString(idIndex)));  
   }  
    
   int titleIndex = cursor.getColumnIndex(CreateDB.*KEY\_TASK\_ITEM*);  
   if (titleIndex >= 0) {  
   task.setTitle(cursor.getString(titleIndex));  
   }  
    
   int descriptionIndex = cursor.getColumnIndex(CreateDB.*KEY\_DESCRIPTION\_TASK*);  
   if (descriptionIndex >= 0) {  
   task.setDescription(cursor.getString(descriptionIndex));  
   }  
    
   // convert timestamp to something readable  
   int dateIndex = cursor.getColumnIndex(CreateDB.*KEY\_DATE\_NAME*);  
   if (dateIndex >= 0) {  
   java.text.DateFormat dateFormat = java.text.DateFormat.*getDateInstance*();  
   String formattedDate = dateFormat.format(new Date(cursor.getLong(dateIndex)).getTime());  
   task.setDateItemAdded(formattedDate);  
   }  
    
   // Add to the groceryList  
   taskList.add(task);  
   } while (cursor.moveToNext());  
   }  
   return taskList;  
   }  
    
   // Update task  
   public int updateTask(Task task)  
   {  
   SQLiteDatabase db = this.getWritableDatabase();  
    
   ContentValues values = new ContentValues();  
   values.put(CreateDB.*KEY\_TASK\_ITEM*, task.getTitle());  
   values.put(CreateDB.*KEY\_DESCRIPTION\_TASK*, task.getDescription());  
   values.put(CreateDB.*KEY\_DATE\_NAME*, java.lang.System.*currentTimeMillis*());  
    
   // update row  
   return db.update(CreateDB.*TABLE\_NAME*, values, CreateDB.*KEY\_ID* + "=?",  
   new String[] {String.*valueOf*(task.getId())}); // return ID of row updated  
   }  
    
   // Delete task  
   public void deleteTask(int id)  
   {  
   SQLiteDatabase db = this.getWritableDatabase();  
   db.delete(CreateDB.*TABLE\_NAME*, CreateDB.*KEY\_ID* + "=?",  
   new String[] {String.*valueOf*(id)});  
   db.close();  
   }  
    
   // Get count  
   public int getTaskCount()  
   {  
   String countQuery = "SELECT \* FROM " + CreateDB.*TABLE\_NAME*;  
   SQLiteDatabase db = this.getReadableDatabase();  
    
   Cursor cursor = db.rawQuery(countQuery, null); // rawQuery: Runs the provided SQL and returns a "Cursor" over the result set (multiple results).  
    
   return cursor.getCount();  
   }  
    
  }

- MainActivity.java: This is the code to create the main interface when starting the app.

public class MainActivity extends AppCompatActivity {  
  
 // Declare variables  
 private AlertDialog.Builder dialogBuilder;  
 private AlertDialog dialog;  
 private EditText titleTask;  
 private EditText descriptionTask;  
 private Button saveButton;  
 private DatabaseHandle db;  
  
 private AppBarConfiguration appBarConfiguration;  
 private ActivityMainBinding binding;  
  
 // Override methods  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 binding = ActivityMainBinding.*inflate*(getLayoutInflater());  
 setContentView(binding.getRoot());  
  
 setSupportActionBar(binding.toolbar);  
  
 db = new DatabaseHandle(this);  
  
 showListTask(); // if exist task, go to ListActivity  
  
 binding.fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
// Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)  
// .setAnchorView(R.id.fab)  
// .setAction("Action", null).show();  
 createPopupDialog();  
 }  
 });  
 }  
  
 // Override methods  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Inflate the menu; this adds items to the action bar if it is present.  
 getMenuInflater().inflate(R.menu.*menu\_main*, 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.*action\_settings*) {  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 // Create Popup Dialog for adding new task  
 private void createPopupDialog()  
 {  
 dialogBuilder = new AlertDialog.Builder(this);  
 View view = getLayoutInflater().inflate(R.layout.*popup*, null);  
  
 titleTask = (EditText) view.findViewById(R.id.*taskItem*);  
 descriptionTask = (EditText) view.findViewById(R.id.*descriptionTask*);  
 saveButton = (Button) view.findViewById(R.id.*saveButton*);  
  
 dialogBuilder.setView(view); // Thiết lập layout cho cửa sổ popup.  
 dialog = dialogBuilder.create(); // Tạo cửa sổ popup từ AlertDialog.Builder.  
 dialog.show();  
  
 saveButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (!titleTask.getText().toString().isEmpty() &&  
 !descriptionTask.getText().toString().isEmpty())  
 {  
 saveTaskToDB(v);  
 }  
  
 }  
 });  
 }  
  
 // Save task to database  
 private void saveTaskToDB(View v)  
 {  
 Task task = new Task();  
 String newTaskTitle = titleTask.getText().toString();  
 String newTaskDescription = descriptionTask.getText().toString();  
  
 task.setTitle(newTaskTitle);  
 task.setDescription(newTaskDescription);  
  
 // Save to db  
 db.addTask(task);  
 Snackbar.*make*(v, "Task saved", Snackbar.*LENGTH\_LONG*).show();  
  
 new Handler().postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 dialog.dismiss();  
 startActivity(new Intent(MainActivity.this, ListActivity.class));  
 }  
 }, 1000);  
 }  
  
 // Check if exist task  
 public void showListTask()  
 {  
 if (db.getTaskCount() > 0)  
 {  
 startActivity(new Intent(MainActivity.this, ListActivity.class));  
 finish();  
 }  
 }  
  
}

* ListActivity.java: This is the code to display the list of "tasks" created, edited or deleted.
* public class ListActivity extends AppCompatActivity {  
    
   // Declare variables  
   private RecyclerView recyclerView;  
   private RecycleViewAdapter recyclerViewAdapter;  
   private List<Task> taskList;  
   private List<Task> taskListEdit;  
   private DatabaseHandle db;  
   private AlertDialog.Builder dialogBuilder;  
   private AlertDialog dialog;  
    
   private EditText titleTask;  
   private EditText descriptionTask;  
   private Button saveButton;  
    
   private AppBarConfiguration appBarConfiguration;  
   private ActivityListBinding binding;  
    
   // Override methods  
   @Override  
   protected void onCreate(Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
    
   binding = ActivityListBinding.*inflate*(getLayoutInflater());  
   setContentView(binding.getRoot());  
    
   setSupportActionBar(binding.toolbar);  
    
   binding.fab.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View view) {  
  // Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)  
  // .setAnchorView(R.id.fab)  
  // .setAction("Action", null).show();  
   createPopupDialog();  
   }  
   });  
    
   // Initialize variables  
   db = new DatabaseHandle(this);  
   recyclerView = (RecyclerView) findViewById(R.id.*recyclerViewID*);  
   recyclerView.setHasFixedSize(true);  
   recyclerView.setLayoutManager(new LinearLayoutManager(this));  
    
   taskList = new ArrayList<>();  
   taskListEdit = new ArrayList<>();  
    
   // Get task from db  
   taskList = db.getAllTasks();  
    
   for (Task t : taskList)  
   {  
   Task task = new Task();  
   task.setId(t.getId());  
   task.setTitle("Title: " + t.getTitle());  
   task.setDescription("Description: " + t.getDescription());  
   task.setDateItemAdded("Added on: " + t.getDateItemAdded());  
   taskListEdit.add(task);  
   }  
    
   // Set adapter  
   recyclerViewAdapter = new RecycleViewAdapter(this, taskListEdit);  
   recyclerView.setAdapter(recyclerViewAdapter);  
   recyclerViewAdapter.notifyDataSetChanged();  
    
   }  
    
   // Create Popup Dialog for adding new task  
   private void createPopupDialog()  
   {  
   dialogBuilder = new AlertDialog.Builder(this);  
   View view = getLayoutInflater().inflate(R.layout.*popup*, null);  
   titleTask = (EditText) view.findViewById(R.id.*taskItem*);  
   descriptionTask = (EditText) view.findViewById(R.id.*descriptionTask*);  
   saveButton = (Button) view.findViewById(R.id.*saveButton*);  
    
   dialogBuilder.setView(view);  
   dialog = dialogBuilder.create();  
   dialog.show();  
    
   saveButton.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View v) {  
   saveTaskToDB(v);  
   }  
   });  
   }  
    
   // Save task to database  
   private void saveTaskToDB(View v)  
   {  
   Task task = new Task();  
   String newTaskTitle = titleTask.getText().toString();  
   String newTaskDescription = descriptionTask.getText().toString();  
    
   task.setTitle(newTaskTitle);  
   task.setDescription(newTaskDescription);  
    
   // Save to db  
   db.addTask(task);  
   Snackbar.*make*(v, "Task saved", Snackbar.*LENGTH\_LONG*).show();  
    
   new Handler().postDelayed(new Runnable() {  
   @Override  
   public void run() {  
   dialog.dismiss();  
   startActivity(new Intent(ListActivity.this, ListActivity.class));  
   finish();  
   }  
   }, 1000);  
   }  
    
    
  }

- DetailActivity.java: This is a detailed display of each "task" in the list.

public class DetailActivity extends AppCompatActivity {  
  
 // Declare variables  
 private TextView titleDet;  
 private TextView descriptionDet;  
 private TextView dateAddedDet;  
 private int taskID;  
  
 // Override methods  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_detail*);  
  
 titleDet = (TextView) findViewById(R.id.*titleDet*);  
 descriptionDet = (TextView) findViewById(R.id.*descriptionDet*);  
 dateAddedDet = (TextView) findViewById(R.id.*dateAddedDet*);  
  
 // Get data from intent  
 Bundle bundle = getIntent().getExtras();  
  
 if (bundle != null)  
 {  
 titleDet.setText(bundle.getString("title"));  
 descriptionDet.setText(bundle.getString("description"));  
 dateAddedDet.setText(bundle.getString("dateAdded"));  
 taskID = bundle.getInt("id");  
 }  
  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
 }  
}