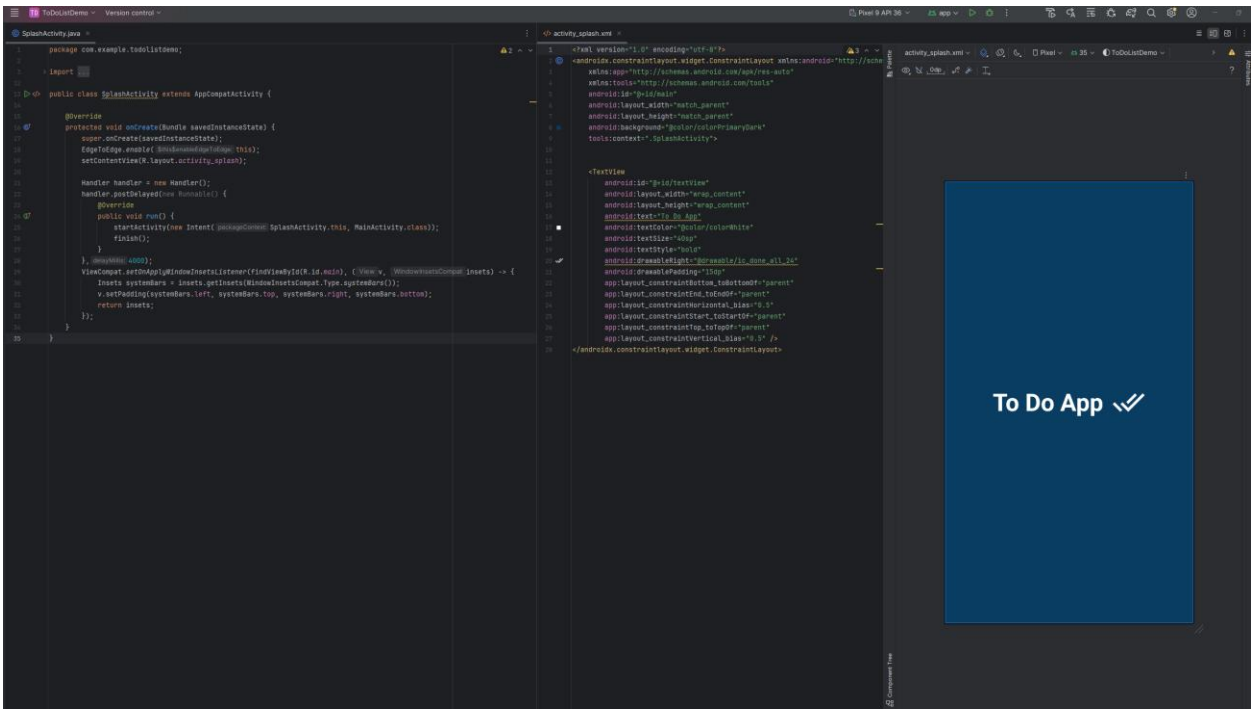
Raina Narayan

Screenshots of code:

Splash Activity



Add New Task Activity

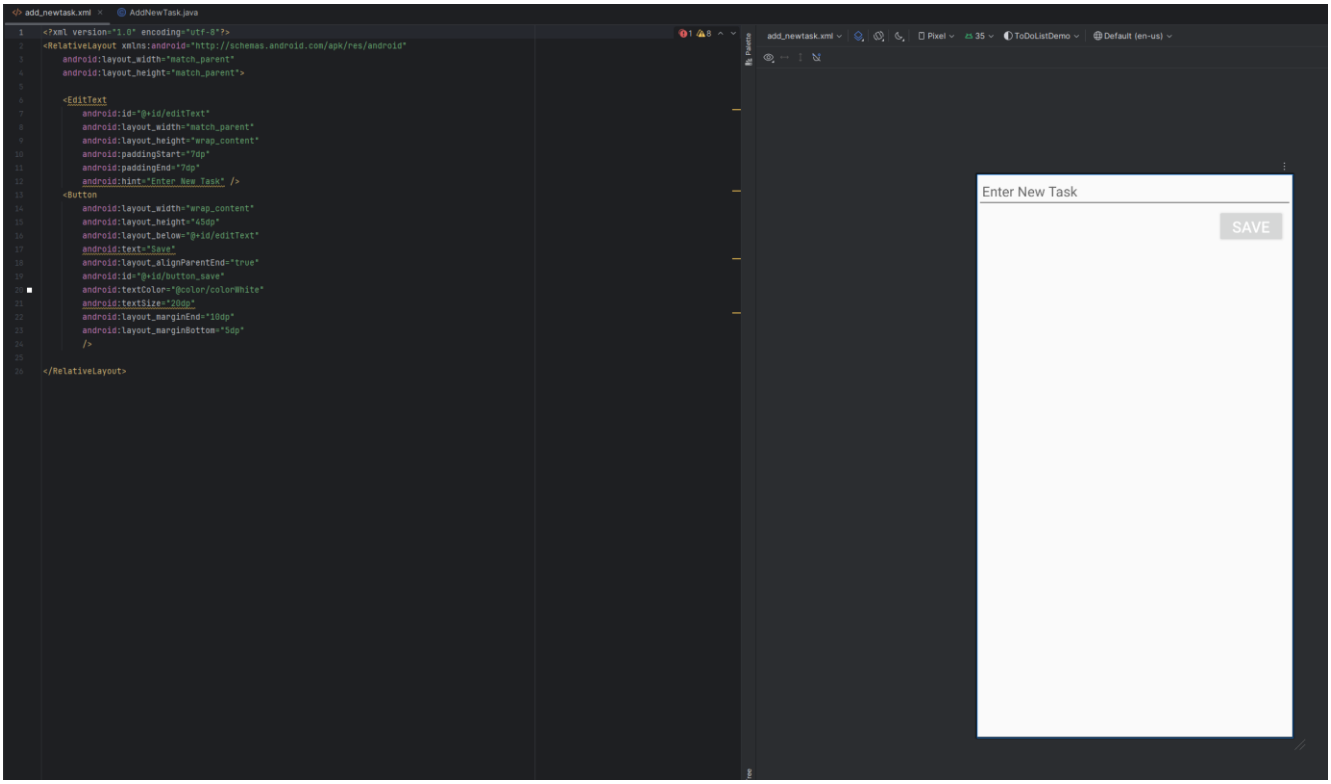
add_newtask.xml | AddNewTask.java

```java
package com.example.todolistdemo;

import ...

public class AddNewTask extends BottomSheetDialogFragment {

    1 usage
    public static final String TAG = "AddNewTask";

    //widgets
    4 usages
    private EditText mEditText;
    7 usages
    private Button mSaveButton;
    3 usages
    private DataBaseHelper myDB;

    1 usage
    public static AddNewTask newInstance() { return new AddNewTask(); }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.add_newtask,container,  attachToRoot: false);
        return v;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        mEditText = view.findViewById(R.id.editText);
        mSaveButton = view.findViewById(R.id.button_save);

        myDB = new DataBaseHelper(getActivity());

        boolean isUpdate = false;
        Bundle bundle =  getArguments();
        if(bundle !=null){
            isUpdate = true;
            String task = bundle.getString( key: "task");
            mEditText.setText(task);

            if(task.length() > 0 ){
                mSaveButton.setEnabled(false);
            }
        }

```

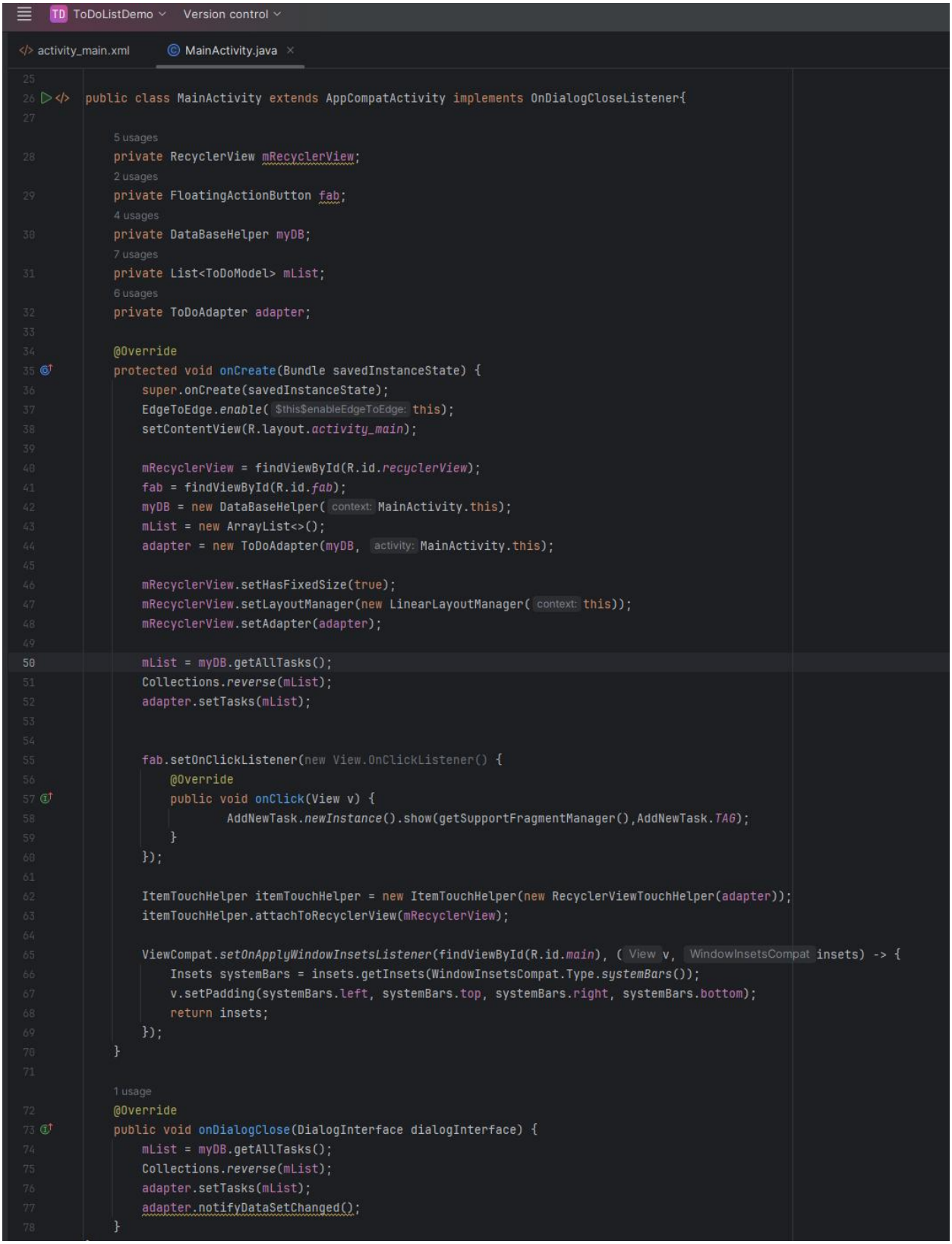
add_newtask.xml | AddNewTask.java

```
22  public class AddNewTask extends BottomSheetDialogFragment {
43      public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
61          }
62
63          mEditText.addTextChangedListener(new TextWatcher() {
64              @Override
65              public void beforeTextChanged(CharSequence s, int start, int count, int after) {
66
67              }
68
69              @Override
70              public void onTextChanged(CharSequence s, int start, int before, int count) {
71                  if(s.toString().equals("")){
72                      mSaveButton.setEnabled(false);
73                      mSaveButton.setBackgroundColor(Color.GRAY);
74                  } else{
75                      mSaveButton.setEnabled(true);
76                      mSaveButton.setBackgroundColor(getResources().getColor(R.color.colorPrimary));
77                  }
78              }
79
80              @Override
81              public void afterTextChanged(Editable s) {
82
83              }
84          });
85
86          boolean finalIsUpdate = isUpdate;
87          mSaveButton.setOnClickListener(new View.OnClickListener() {
88              @Override
89              public void onClick(View v) {
90                  String text = mEditText.getText().toString();
91                  if(finalIsUpdate){
92                      myDB.updateTask(bundle.getInt( key: "id"),text);
93                  } else {
94                      ToDoModel item = new ToDoModel();
95                      item.setTask(text);
96                      item.setStatus(0);
97                      myDB.insertTask(item);
98                  }
99                  dismiss();
100             }
101         });
102     }
103
104     @Override
105     public void onDismiss(@NonNull DialogInterface dialog) {
106         super.onDismiss(dialog);
107         Activity activity = getActivity();
108         if(activity instanceof OnDialogCloseListener){
109             ((OnDialogCloseListener) activity).onDialogClose(dialog);
110         }
111     }
112 }
113
```

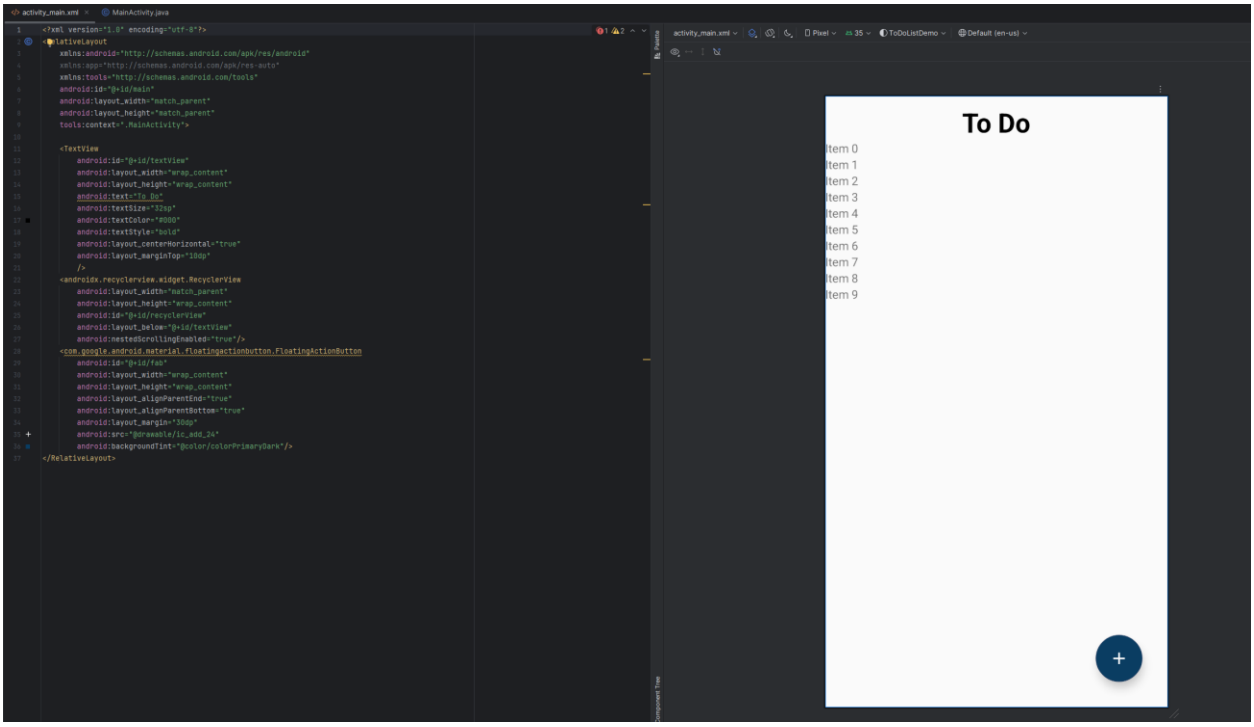
Add New Task Layout



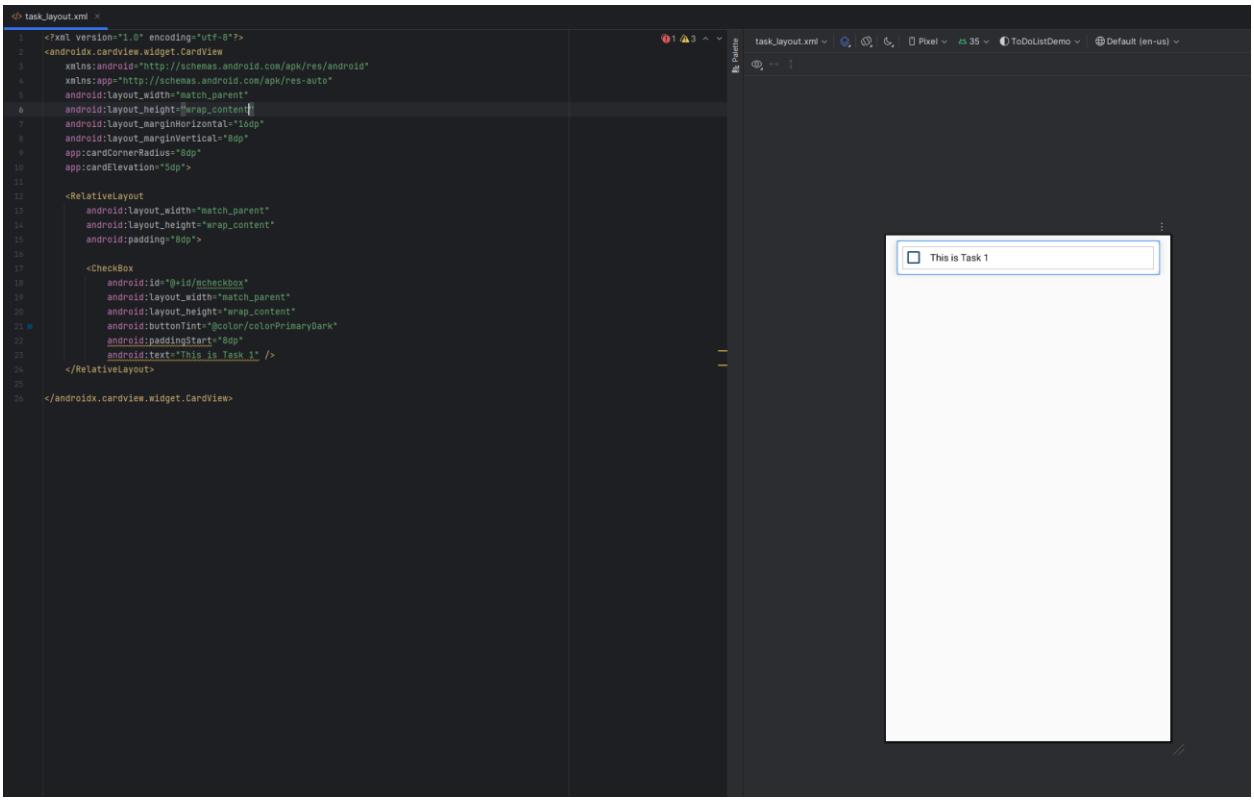
Main Activity



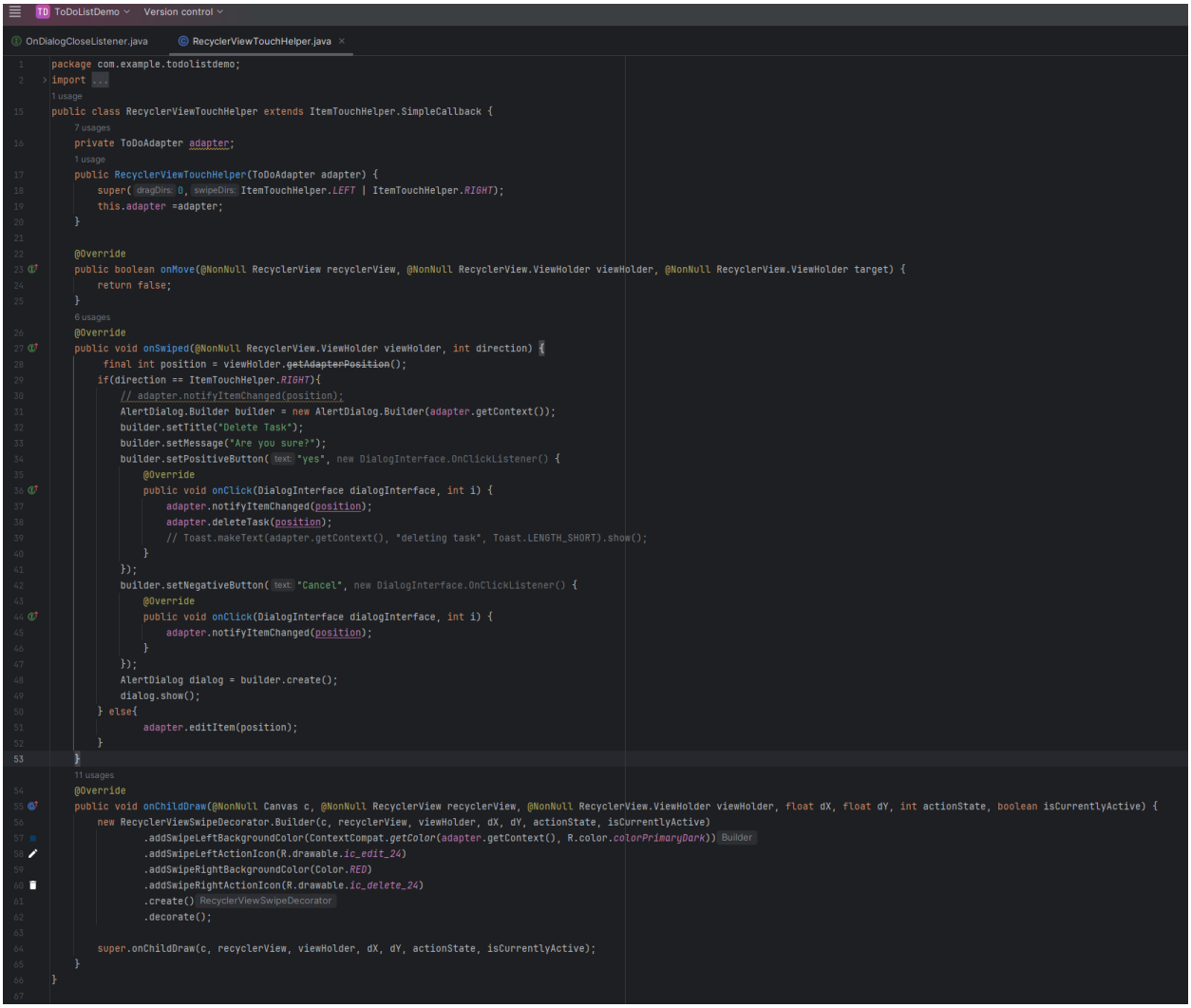
Main Layout



Task Layout



Recycler View Touch Helper



```java
package com.example.todolistdemo;
import ...
1 usage
public class RecyclerViewTouchHelper extends ItemTouchHelper.SimpleCallback {
    7 usages
    private ToDoAdapter adapter;
    1 usage
    public RecyclerViewTouchHelper(ToDoAdapter adapter) {
        super( dragDirs: 0, swipeDirs: ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT);
        this.adapter =adapter;
    }

    @Override
    public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {
        return false;
    }
    6 usages
    @Override
    public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
         final int position = viewHolder.getAdapterPosition();
        if(direction == ItemTouchHelper.RIGHT){
            // adapter.notifyItemChanged(position);
            AlertDialog.Builder builder = new AlertDialog.Builder(adapter.getContext());
            builder.setTitle("Delete Task");
            builder.setMessage("Are you sure?");
            builder.setPositiveButton( text: "yes", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    adapter.notifyItemChanged(position);
                    adapter.deleteTask(position);
                    // Toast.makeText(adapter.getContext(), "deleting task", Toast.LENGTH_SHORT).show();
                }
            });
            builder.setNegativeButton( text: "Cancel", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    adapter.notifyItemChanged(position);
                }
            });
            AlertDialog dialog = builder.create();
            dialog.show();
        } else{
                adapter.editItem(position);
        }
    }
    11 usages
    @Override
    public void onChildDraw(@NonNull Canvas c, @NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, float dX, float dY, int actionState, boolean isCurrentlyActive) {
        new RecyclerViewSwipeDecorator.Builder(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive)
                .addSwipeLeftBackgroundColor(ContextCompat.getColor(adapter.getContext(), R.color.colorPrimaryDark)) Builder
                .addSwipeLeftActionIcon(R.drawable.ic_edit_24)
                .addSwipeRightBackgroundColor(Color.RED)
                .addSwipeRightActionIcon(R.drawable.ic_delete_24)
                .create() RecyclerViewSwipeDecorator
                .decorate();

        super.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);
    }
}
```

On Dialog Close Listener

```java
package com.example.todolistdemo;
import android.content.DialogInterface;
3 usages  1 implementation
public interface OnDialogCloseListener {
    1 usage  1 implementation
    void onDialogClose(DialogInterface dialogInterface);
}
```

Database Helper

DataBaseHelper.java

```java
package com.example.todolistdemo.Utils;
import ...
9 usages
public class DataBaseHelper extends SQLiteOpenHelper {
    12 usages
    private SQLiteDatabase db;
    1 usage
    private static final String DATABASE_NAME = "TODO_DATABASE";
    7 usages
    private static final String TABLE_NAME = "TODO_TABLE";
    1 usage
    private static final String COL_1 = "ID";
    3 usages
    private static final String COL_2 = "TASK";
    3 usages
    private static final String COL_3 = "STATUS";
    2 usages
    public DataBaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, factory: null, version: 1); //error says this is null
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE IF NOT EXISTS " + TABLE_NAME + "(ID INTEGER PRIMARY KEY AUTOINCREMENT, TASK TEXT, STATUS INTEGER)");
    }
    10 usages
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {...}
    //INSERT METHOD
    1 usage
    public void insertTask(ToDoModel model){
        db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COL_2, model.getTask());
        values.put(COL_3, 0);
        db.insert(TABLE_NAME, nullColumnHack: null, values);
    }
    //UPDATE METHOD
    1 usage
    public void updateTask(int id, String task) {
      db = this.getWritableDatabase();
      ContentValues values = new ContentValues();
      values.put(COL_2, task);
      db.update(TABLE_NAME, values, whereClause: "ID=?", new String[]{String.valueOf(id)});
    }
    //UPDATE STATUS METHOD
    2 usages
    public void updateStatus(int id, int status){
        db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COL_3, status);
        db.update(TABLE_NAME, values, whereClause: "ID=?", new String[]{String.valueOf(id)});
    }
    //Delete TASK METHOD
    1 usage
    public void deleteTask(int id){
        db =this.getWritableDatabase();
        db.delete(TABLE_NAME, whereClause: "ID=?",new String[]{String.valueOf(id)});
    }
```

```java
    //GET ALL TASK METHOD
    2 usages
    public List<ToDoModel> getAllTasks(){
        db = this.getWritableDatabase();
        Cursor cursor = null;
        List<ToDoModel> modelList = new ArrayList<>();

        db.beginTransaction();
        try{
            cursor = db.query(TABLE_NAME,  columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
            if (cursor !=null){
                if (cursor.moveToFirst()){
                    do {
                        ToDoModel task = new ToDoModel();
                        task.setId(cursor.getInt(cursor.getColumnIndexOrThrow(COL_1)));
                        task.setTask(cursor.getString(cursor.getColumnIndexOrThrow(COL_2)));
                        task.setStatus(cursor.getInt(cursor.getColumnIndexOrThrow(COL_3)));
                        modelList.add(task);
                    } while (cursor.moveToNext());
                }
            }
        } finally {
            db.endTransaction();
            cursor.close();
        }
        return modelList;
    }
}
```

To Do Model

ToDoModel.java

```java
package com.example.todolistdemo.Model;

17 usages
public class ToDoModel {
    2 usages
    private String task;
    2 usages
    private int id, status;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    1 usage
    public int getStatus() {
        return status;
    }

    2 usages
    public void setStatus(int status) {
        this.status = status;
    }

    3 usages
    public String getTask() {
        return task;
    }

    2 usages
    public void setTask(String task) {
        this.task = task;
    }
}
```

To Do Adapter

ToDoAdapter.java

```java
package com.example.todolistdemo.Adapter;
import ...
7 usages
public class ToDoAdapter extends RecyclerView.Adapter<ToDoAdapter.MyViewHolder> {
    6 usages
    private List<ToDoModel> mList;
    3 usages
    private MainActivity activity;
    4 usages
    private DataBaseHelper myDB;
    1 usage
    public ToDoAdapter (DataBaseHelper myDB, MainActivity activity){
        this.activity = activity;
        this.myDB = myDB;
    }
    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.task_layout, parent, attachToRoot: false);
        return new MyViewHolder(v);
    }
    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        final ToDoModel item = mList.get(position);
        holder.mCheckBox.setText(item.getTask());
        holder.mCheckBox.setChecked(toBoolean(item.getStatus()));
        holder.mCheckBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked){
                    myDB.updateStatus(item.getId(), status: 1);
                } else{
                    myDB.updateStatus(item.getId(), status: 0);
                }
            }
        });
    }
    1 usage
    public boolean toBoolean(int num){
        return num !=0;
    }
    //METHOD TO RETURN CONTEXT
    2 usages
    public Context getContext() { return activity; }

    2 usages
    public void setTasks(List<ToDoModel>mList){
        this.mList = mList;
        notifyDataSetChanged();
    }
    1 usage
    public void deleteTask(int position){
        ToDoModel item =  mList.get(position);
        myDB.deleteTask(item.getId());
        mList.remove(position);
        notifyItemRemoved(position);
    }
```

```
    public void editItem(int position){
        ToDoModel item =  mList.get(position);

        Bundle bundle = new Bundle();
        bundle.putInt("id",item.getId());
        bundle.putString("task", item.getTask());


        AddNewTask task =  new AddNewTask();
        task.setArguments(bundle);
        task.show(activity.getSupportFragmentManager(),task.getTag());
    }
    @Override
    public int getItemCount() {
        return mList.size();
    }
    4 usages
    public static class MyViewHolder extends RecyclerView.ViewHolder{
        4 usages
        CheckBox mCheckBox;
        1 usage
        public MyViewHolder(@NonNull View itemView) {
            super(itemView);
            mCheckBox = itemView.findViewById(R.id.mcheckbox);
        }
    }
}
```

Colors

colors.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>


    <color name="colorWhite">#FFFFFFFF</color>
    <color name="colorPrimary">#192A56</color>
    <color name="colorPrimaryDark">#0A3D62</color>
    <color name="colorAccent">#192A56</color>

</resources>
```

Build Gradle (Model: app)

build.gradle.kts (:app)

You can use the Project Structure dialog to view and edit your project configuration

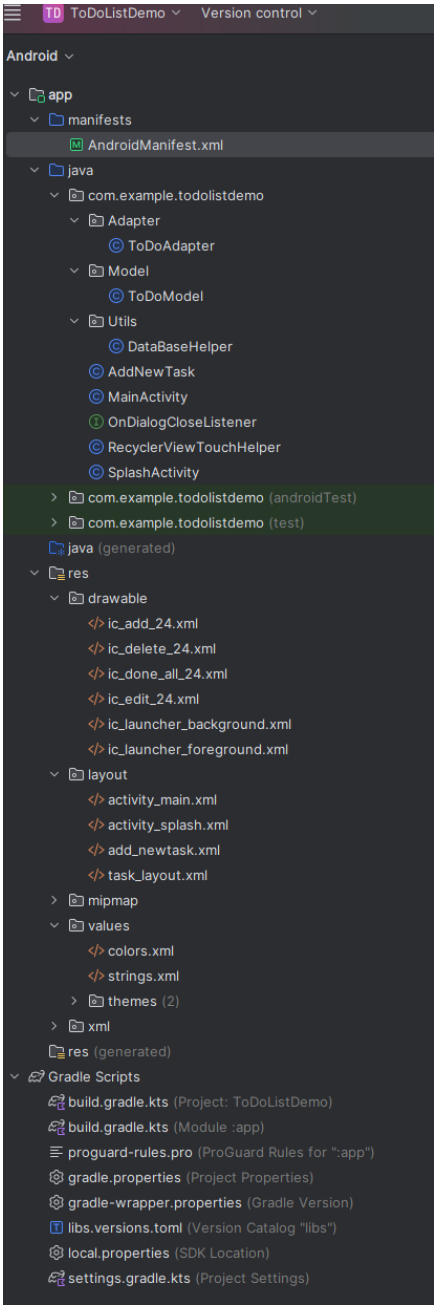
```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.example.todolistdemo"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.todolistdemo"
        minSdk = 26
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile( name: "proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

dependencies {

    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
    //implementation("it.xabaras.android:recyclerview-swipedecorator:1.4")
    implementation(libs.recyclerview.swipedecorator)

}
```

Android Manifest

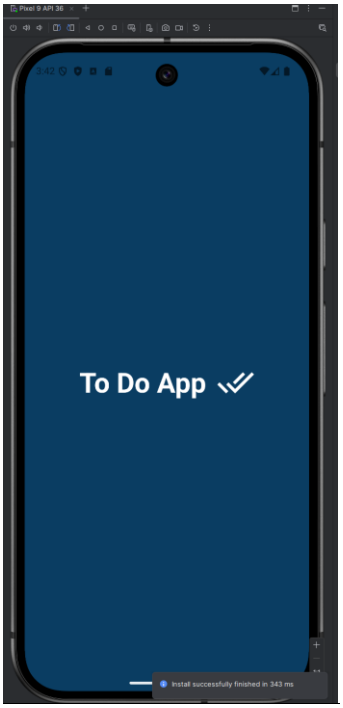
build.gradle.kts (:app) × AndroidManifest.xml ×

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="ToDoListDemo"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ToDoListDemo"
        tools:targetApi="31">
        <activity
            android:name=".SplashActivity"
            android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        </activity>

        <activity
            android:name=".MainActivity"
            android:exported="true">

        </activity>
    </application>

</manifest>
```

Project Structure



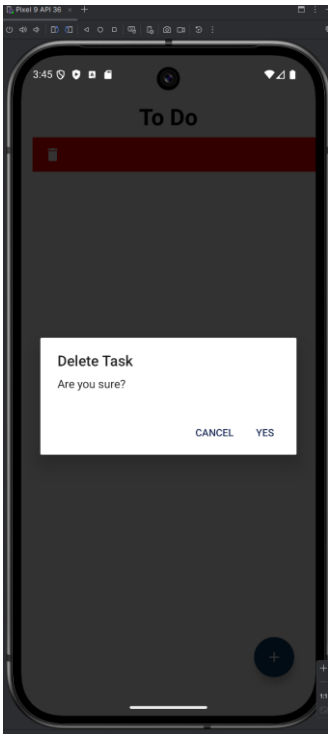
Screenshots of output:




3:44
To Do
New Task

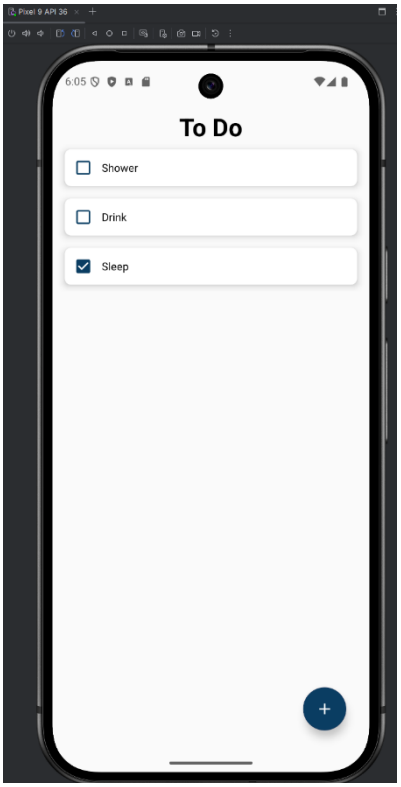

3:44
To Do
New Task
SAVE


3:45
To Do
New Task - EDITED


3:45
To Do
Delete Task
Are you sure?
CANCEL
YES


3:45
To Do


3:47
To Do
Sleep


6:05
To Do
Shower
Drink
Sleep