1. Introduction

Android To-Do List Application list of tasks you need to complete or things that you want to-do. Most typically, they're organised in order of priority. Traditionally, they'rewritten on a piece of paper or post it notes and act as a memory aid. To-do lists are a list of tasks that an individual needs to complete or accomplish. Tasks are typically putin order by priority or importance. A to-do list can be written on a piece of paper or byutilizing task management software. Items on the list can pertain to one's personal or professional life. aim to address the daily problem of taking notes at appropriate times when the event is fast paced and you do not have time to pull out your pen and paper.Taking a note is the process of taking information from a source or event. This often takes the form of writing , Users may type on mobile devices just like they would on paper with To-do List apps.

The To-do List app is an application, built for storing information and daily work thatare important to the person. This app can be used to store all morning to night activityand information digitally and helps to complete the written process digitally to save a lot of time.

1.1) Abstract

A to-do list is a list of tasks that need to be completed, typically organized in order of priority. It is one of the simplest solutions for task management and providesa minimal and elegant way for managing tasks a person wishes to accomplish. Our aimis to design a simple and elegant website for people to keep a track of the status of their tasks. Making a to-do list is an easy and important task that everyone should do.
The immense satisfaction that one gets when completing the task and marking it on thelist are incomparable. More Over, creating a list of tasks ensure you don't miss out onanything. It's a scientific fact that when you write the tasks that you need to complete,you are even more motivated to complete it. With this in mind,
we come to build a platform which will help people create their own task list. with thehelp of modern tools and technologies, we strive to build a minimal and efficient to-dolist which minimizes distractions and helps people achieve task management with ease and without hassle

1.2) Need for System

The need for a Todo List app can vary from person and from context to context, but generally, a Todo List app several common purposes and Fulfils Specific needs for users. Here are some of the primary Needs app can address

1) Information Capture:

Many people need a way to capture information on the go. A Todo List app provides adigital alternative a traditional Todo List taking methods like pen and paper. Users cantake a daily work on their Smartphones, tablets or computers, ensuring they don't lose important information.

2) Accessibility:

Digital Todo List are accessible from anywhere and across device(if the app supports synchronization). This address the need to access Todo List on the go, whether on a smartphone, tablet.

3) Privacy and security:

Users may have sensitive or personal information they want to keep private.

2) Proposed System

2.1) Problem Statement

Existing Android todo list app-taking apps lack seamless synchronization across devices ,robust collaborative feature, advanced privacy controls, modern user-interfaces, and comprehensive rich text editing, hindering a user- friendly and efficient note-taking experience.

2.2) Objective of Proposed System

Efficient Note-Taking:

Provide users with a platform to efficiently create, organize, and manage digital notes. Efficient note-taking is crucial for productivity and staying organized. When creating a to-do list project focused on efficient note-taking, consider the following key components:

Digital or Physical Platform:

Choose a platform that suits your preferences, whether it's a digitaltool like Todoist, Trello, or a physical notebook. Each has its advantages, so select one that aligns with your workflow.Organize tasks into categories or projects to streamline your to-do list. Prioritize tasks based on urgency and importance, ensuring that critical items are addressed first.Clear and Concise Descriptions:

User-Friendly Interface:

Create an intuitive and user-friendly interface that makes note- taking and organization easy for users.

Offline Access:

Enable users to create and access notes even when they are offline, making the app versatile and suitable for various situations.

2.3) Functional Requirement

- Todo list creation and editing
- Search functionality
- Attachments
- Offline Access

Non-Functional Requirement

- Performance
- Scalability
- Realibility
- Security

2.4) Scope of the System

This Todo list App contains an Android application for taking and managing notes. The app provides a user-friendly interface and functionality to create, edit, and delete notes, as well as organize them into categories.

2.5) Module Specification

1.Module

Add Todo List: The user can add new notes Edit Todo List: The user can edit old notes Delete Todo List: The user can delete notes

2.Module

Share module: The user can share notes with anyone.

3.Module

Search module: The user can search any note

4.Module

Color module: The user can select any color to the notes

2.6 Operating Environment:

The mobile Application is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility.

Hardware:

Processor: Android 7
RAM: 8GB
Database: Firebase
IDE: Android Studio 2022

Software:

Operating System: Windows
RAM: 8GB
Database: Firebase
IDE: Android Studio 2022

3 Requirement Determination And Analysis

3.1) Fact Finding Methods:

When embarking on a to-do list project with a focus on fact-finding methods, it's essential to gather information effectively and efficiently. Here are steps and considerations for incorporating fact-finding methods into your to-do list project:

Define Information Needs:

Clearly outline the information you need to gather. Be specific about the type of facts, data, details required for each task.

Research Tools and Resources:

Identify the tools and resources necessary for your fact-finding efforts. This could online databases, books, articles, interviews, surveys, or any other relevant sources.

3.2) Feasibility Study:

A feasibility study for an Android notes app project is a critical step in determining whether the project is viable, worth pursuing, and likely to succeed. This study assesses various aspects, including technical, economic, operational, legal, and scheduling considerations.

Technical Feasibility:

App Development Skills:

Evaluate whether the development team possesses the required technical skills for Android app development. Determine if they can handle the complexity of creating a notes app.

Technology Stack:

Choose the appropriate development tools, frameworks, and libraries for Android app development. Integration: the feasibility of integrating with cloud services, third-party APIs, or databases for features like synchronization and backup.

Economic Feasibility:

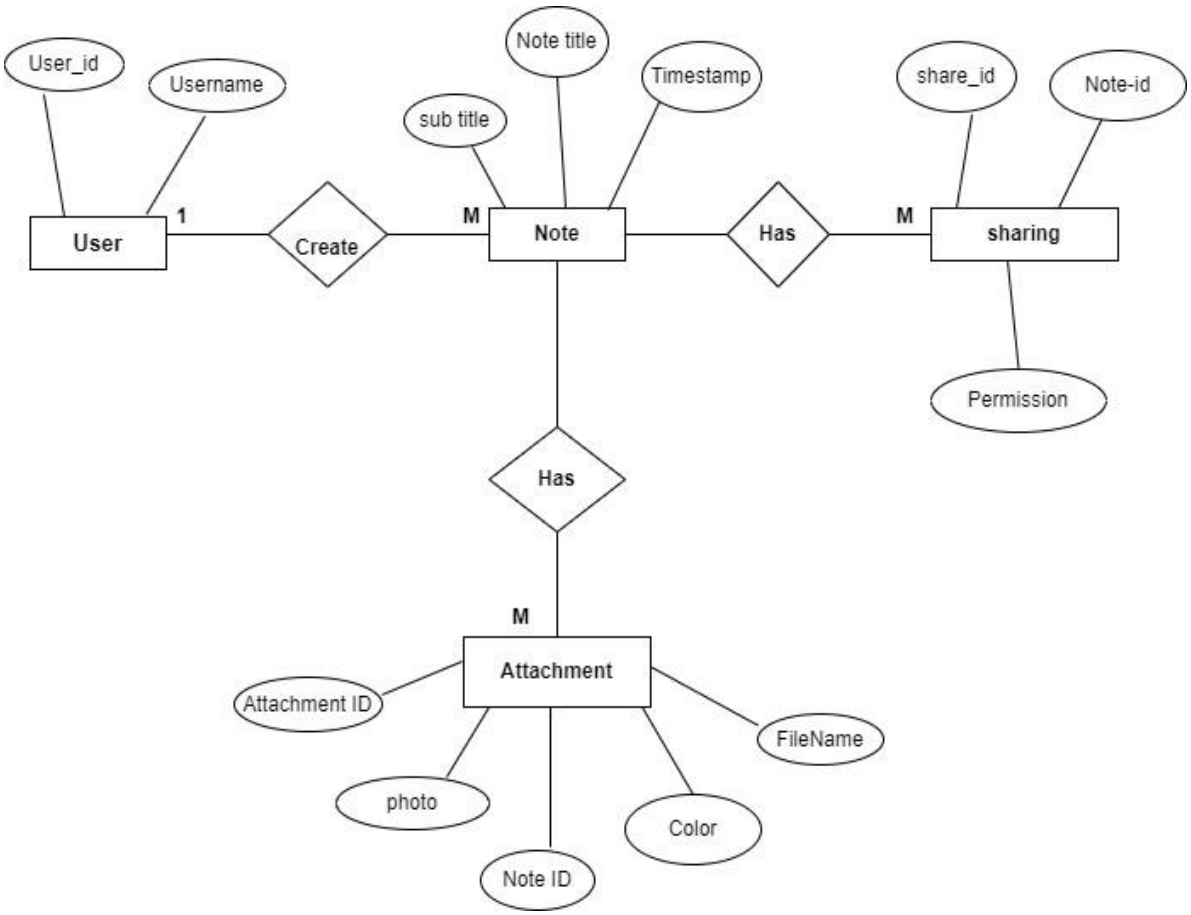
Budget: Calculate the estimated costs of developing, launching, and maintaining the app. This includes developer salaries, infrastructure costs, marketing expenses, and ongoing maintenance.

Operational Feasibility:

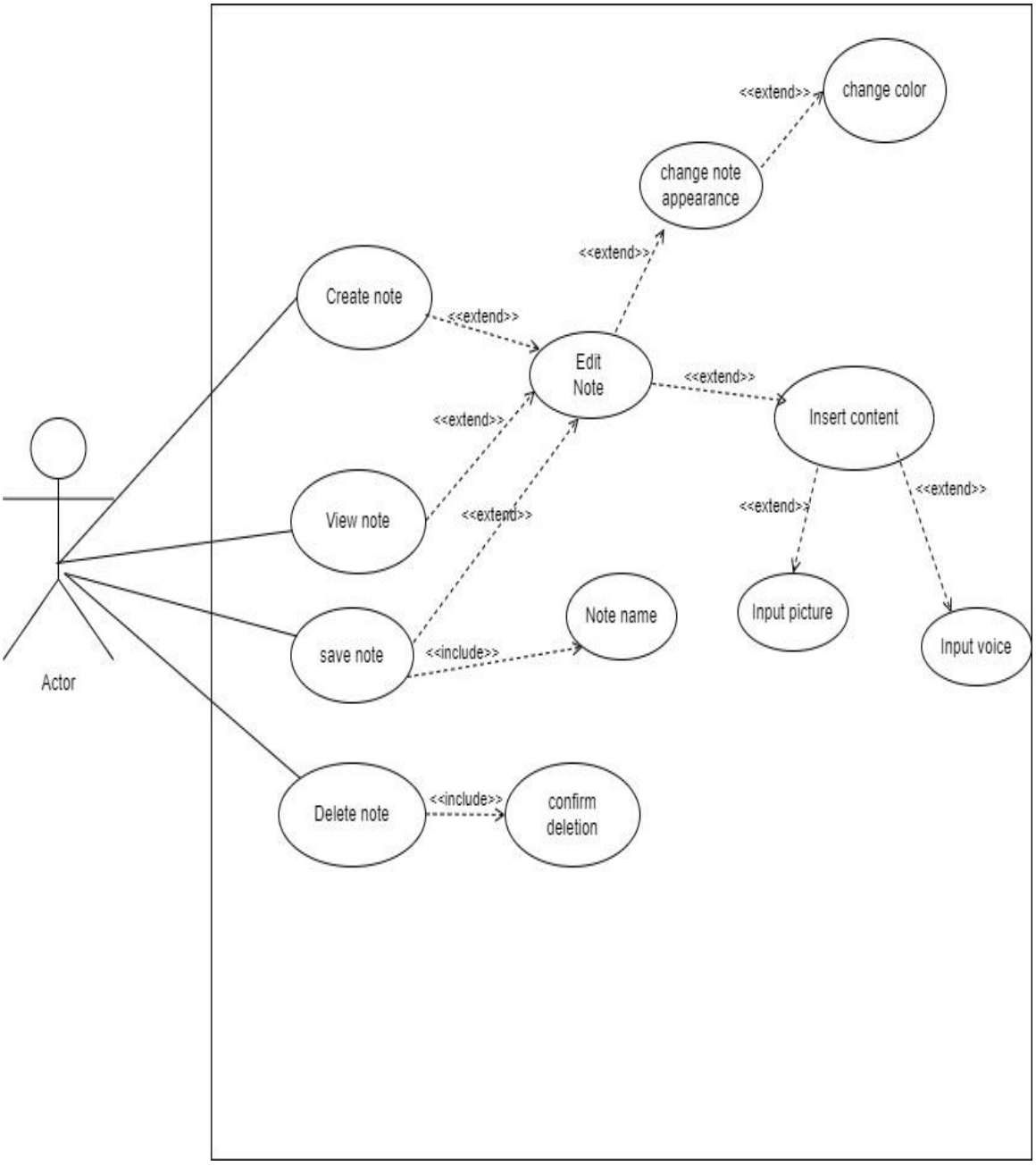
Market Research: Conduct market research to understand the demand for a notes app. Identify your target audience and assess their needs and preferences. Support and Maintenance: Plan how you will handle ongoing support and maintenance, including updates, bug fixes, and user support

4 System Analysis And Design

4.1 ER Diagram

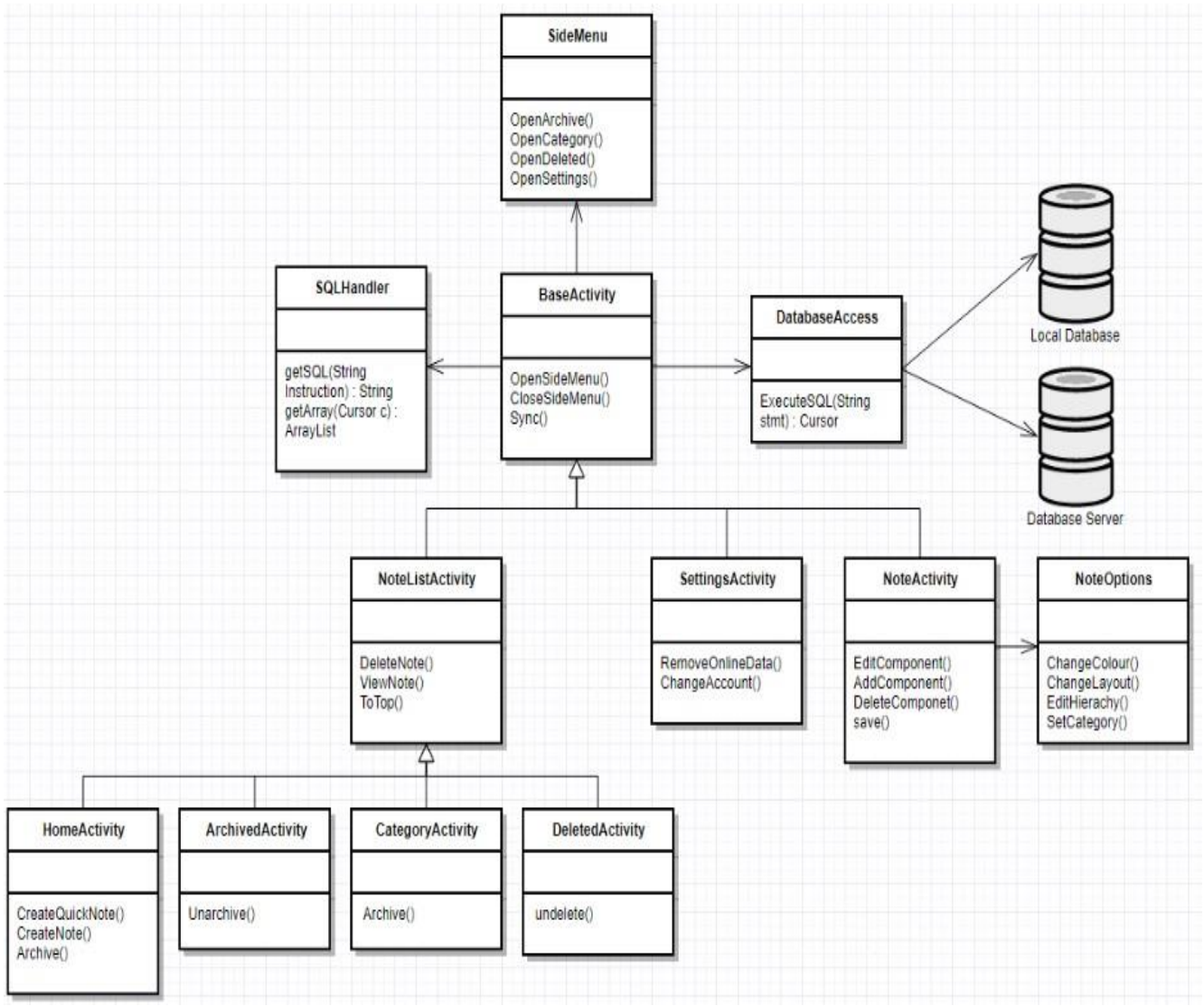


4.2 Use case diagram

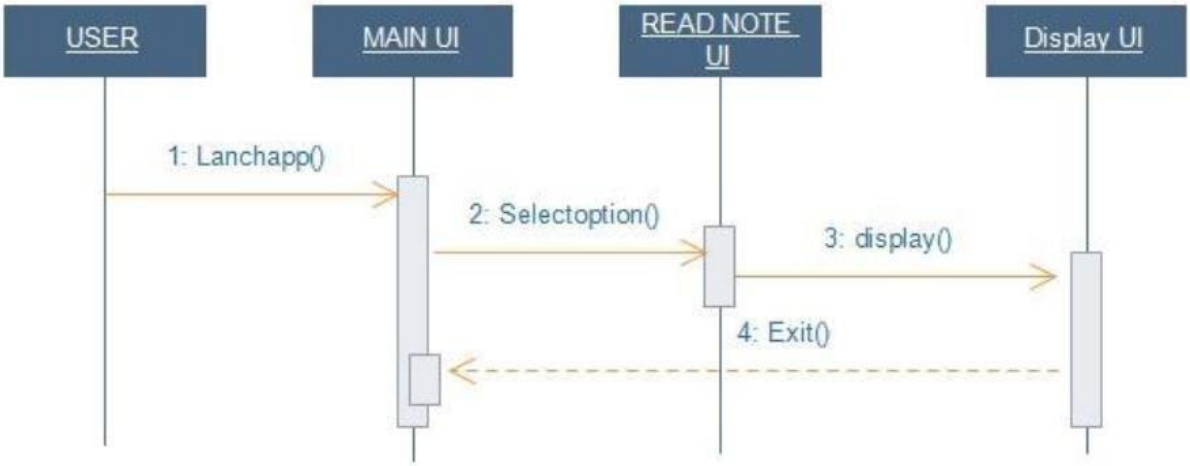


4.3 Class diagram

Class Diagram

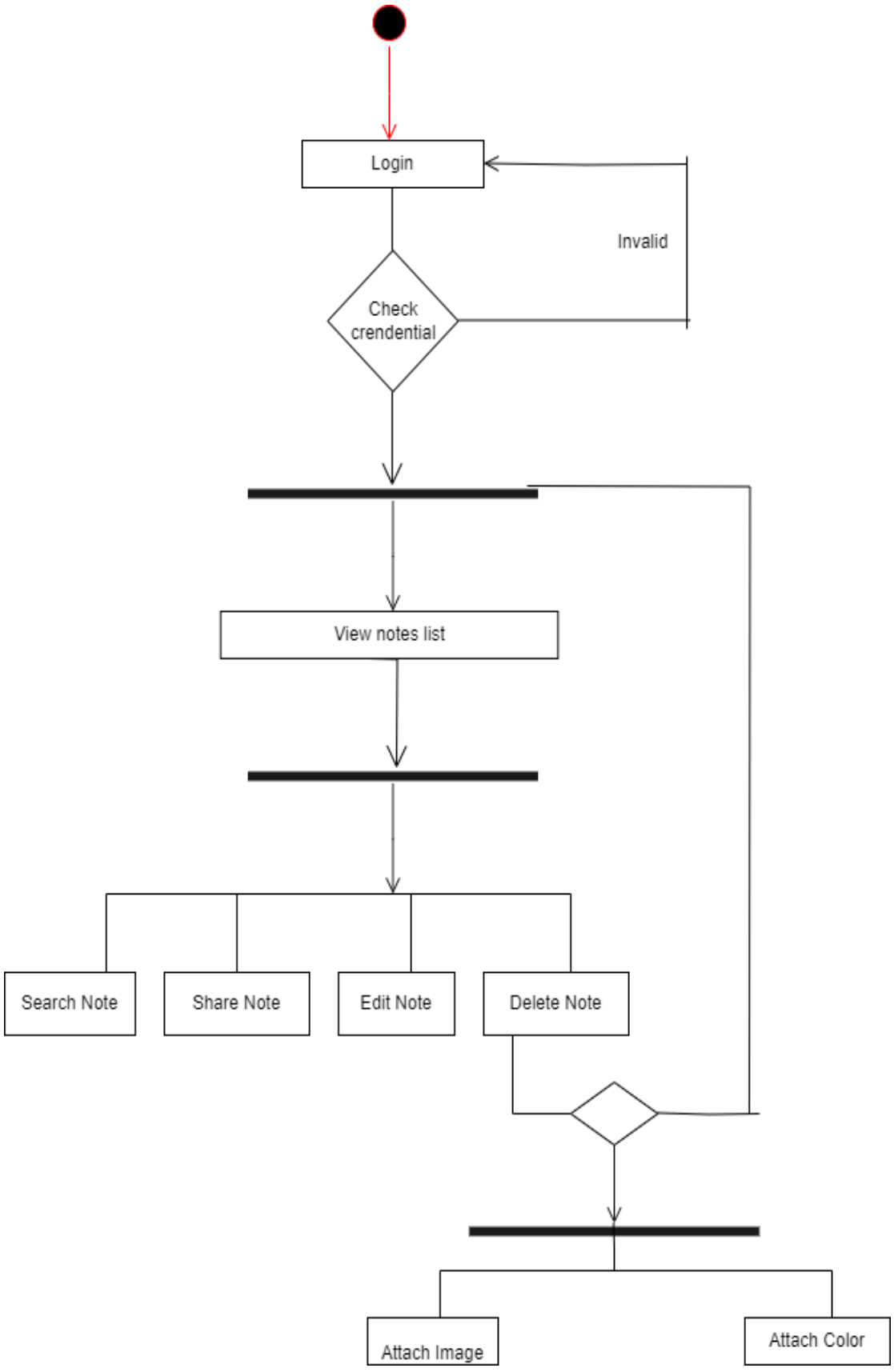


4.4 Sequence diagram



Sequence Diagram for Read Note module.

4.5 Activity Diagram



4.6) TABLE SPECIFICATION [DATA DICTIONARY]

Table: Note

Field Name	Data type	Constraint	Description
Note Id	Integer	Primary Key, Auto Increment	Unique identifier for each note
Title	Text		Title of note
Content	Text		The main text or body of the note
Timestamp	Datetime		The date was created

Table : Sharing

Field Name	Data type	Constraint	Description
Share Id	Int	Primary key, Auto Increment	Unique identifier for each sharing
Note Id	Int	Foreign Key	Reference to the note being shared
Permission	Varchar(20)	Not Null	

Table: Users

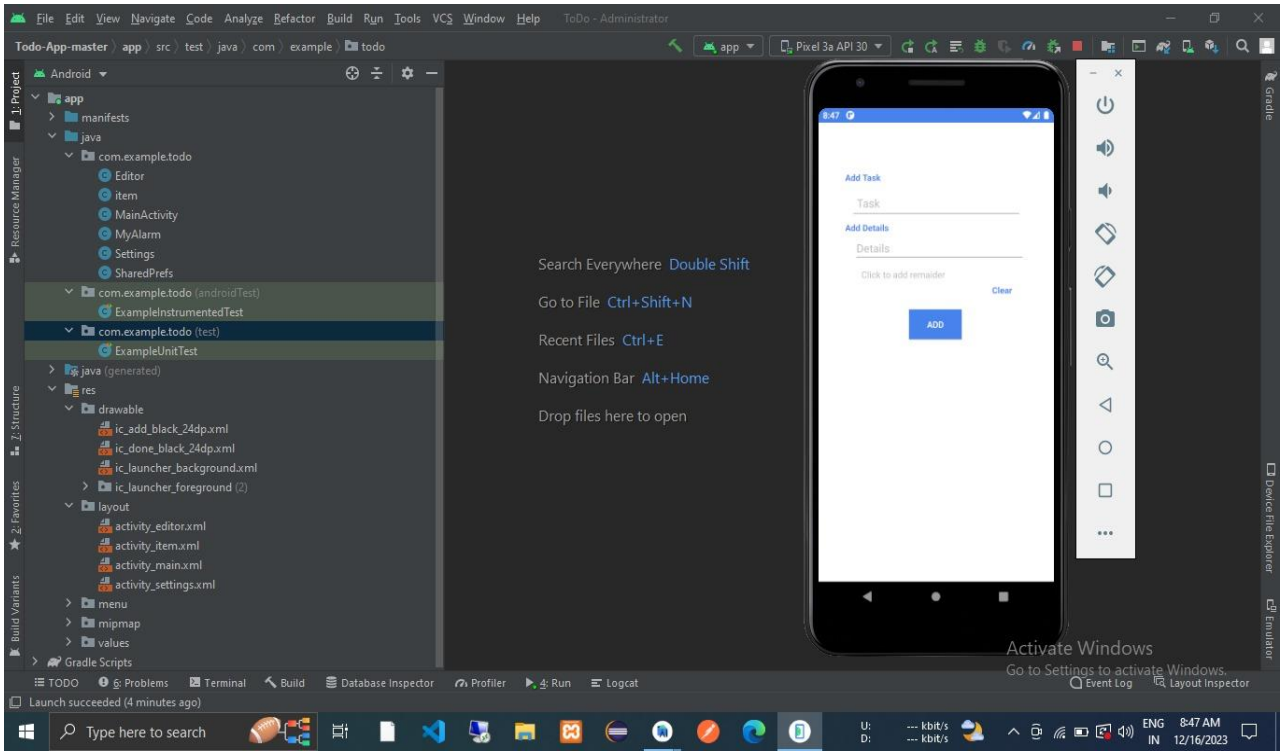
Field Name	Data type	Constraint	Description
UserId	Integer	Primary key	Unique identifier foe each user
Username	Text		

Table: Attachments

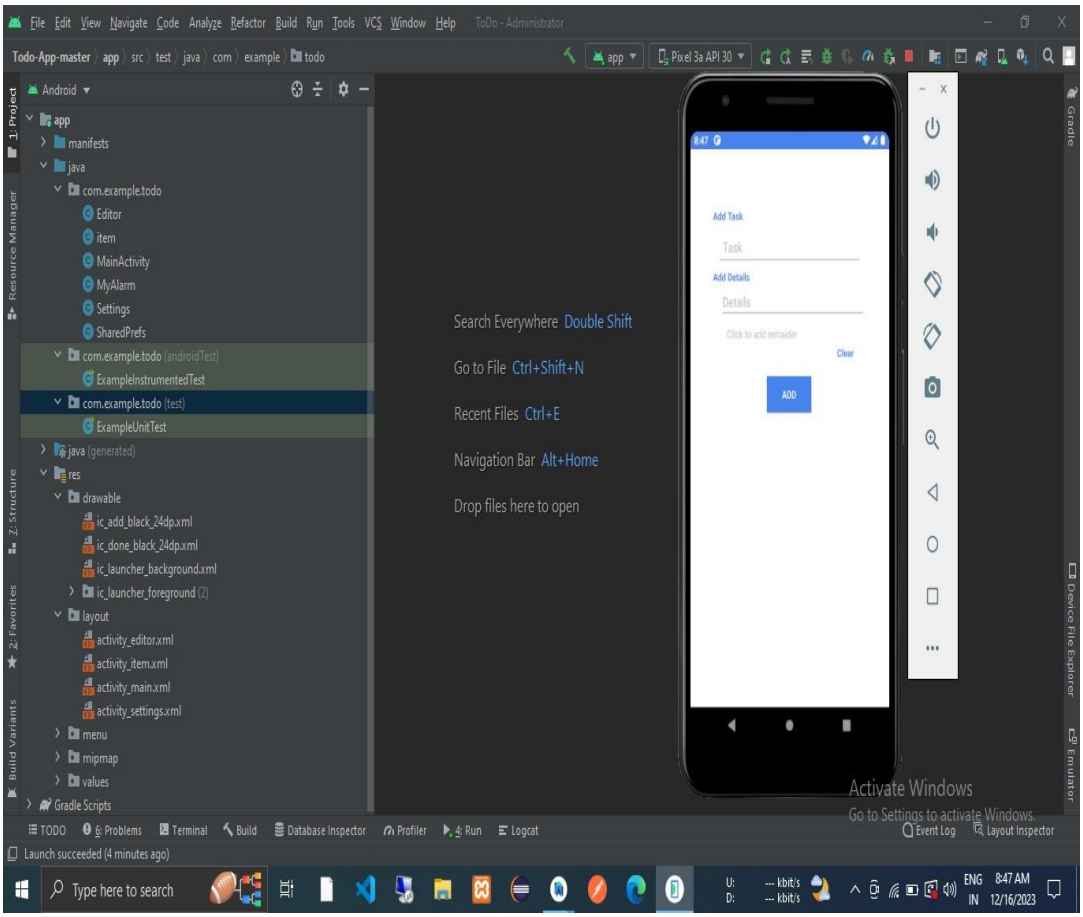
Field Name	Data type	Constraint	Description
Attachment ID	Int	Primary key	Unique identifier foe each attachment
Note Id	Int	Foreign Key	From note table
File Name	Varchar(25)	Not Null	
Color	Varchar(5)	Not Null	

4.7) User Interface Design and report:

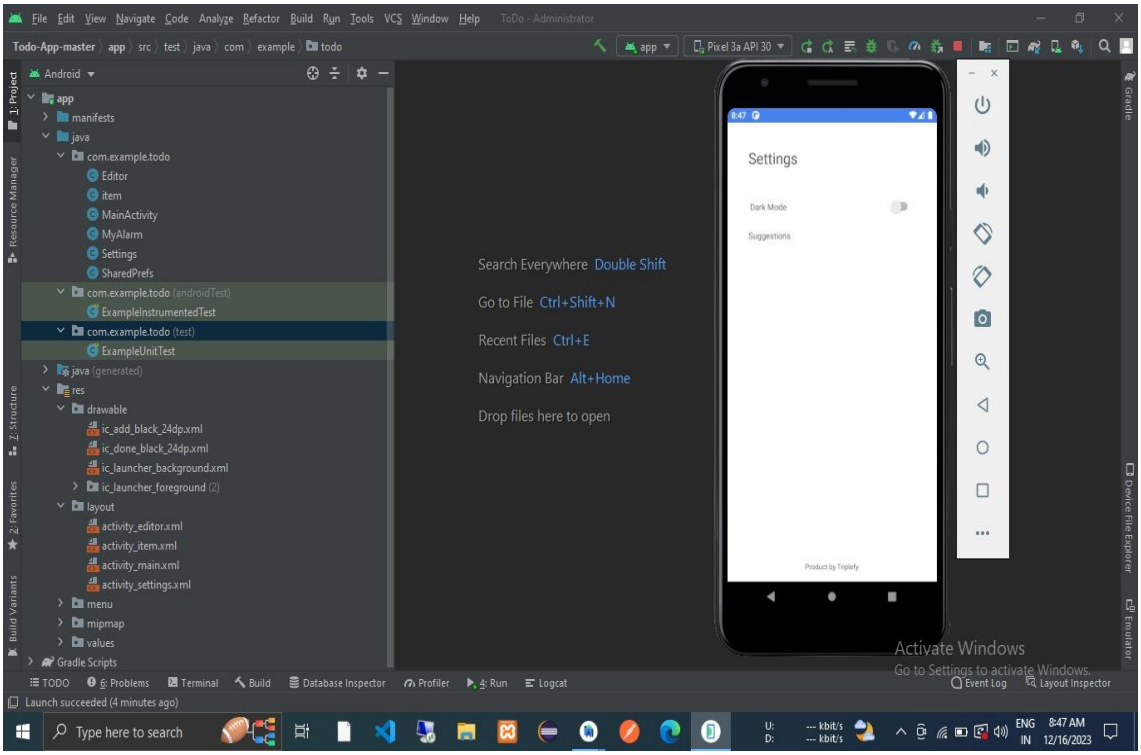
4.7 1) HOME PAGE



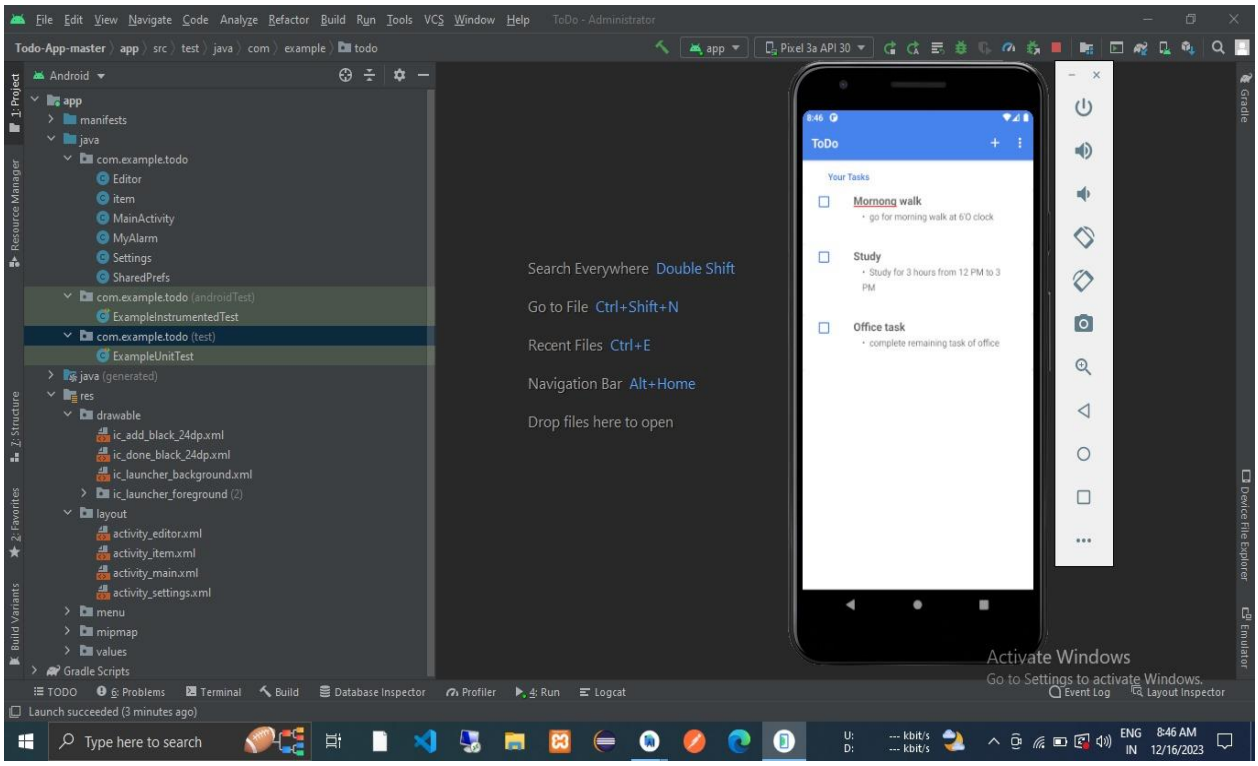
4.7.2) ADD TASK PAGE



4.7.3) Settings



4.7.4) View page



5.Test case

Test Case Name	Todo List App Home Screen Test
Test Case Type	Functional Testing
Module	Home Screen
Status	XYZ
Severity	Critical
Release	Beta
Version	1
Pre-Condition	User is logged in
Test Data	User Credentials: Valid | Job Details: Title, Description, Location, Type, etc.
Summary	To test the functionality of the home screen in a Todo List app.

Sr No	Description/Action	Inputs	Expected Result	Actual Result	Status
1	Launch the Todo List app		App successfully launches to the home screen	Launched to Home Screen	Pass
2	Display existing tasks on home screen		Home screen displays existing tasks, if any.	Tasks displayed correctly	Pass
3	Show "Add Task" button prominently		Add Task" button is visible and accessible	Button displayed correctly	Pass
4	Allow marking tasks as completed	Mark a task as completed	Task is visually marked as completed	Task marked as completed	Pass

6. Drawback and limitation

1. Limited Offline Functionality

This notes app may have limitations in offline functionality, making it challenging for users to access their notes an internet connection.

2 .Limited Formatting Options

This notes app offer basic formatting options , but they may lack advanced features such as rich text formatting , complex tables, or the ability to embed multimedia elements

7. Proposed Enhancement

Enhancing a notes app on Android can involve a variety of improvements to enhance functionality , user experience and overall usability.

1. Advanced Formatting Options:

Introduce more advanced formatting options, such as rich text editing , markdown support and the ability to embed multimedia elements like images, audio clips and videos.

2 Smart Reminders:

Implement intelligent reminder features, allowing users to set reminders based on the content their notes or specific keywords

8.Conclusion

In summary the android notes app project aims to create a feature -rich , user - friendly application .Proposed enhancement include advanced formatting, real-time collaboration,improved security , and cross platform syncing. These addition will make the app versatile, accessible and aligned with modern user expectation. "In conclusion, a well-organized and regularly updated to-do list is a powerful tool for enhancing productivity and managing tasks efficiently. By prioritizing and breaking down tasks into actionable steps, individuals can stay focused, reduce stress, and accomplish their goals in a more structured manner. It's important to review and update the to-do list regularly, adapting it to changing priorities and deadlines. Additionally, the sense of accomplishment derived from crossing off completed tasks provides motivation and encourages continued productivity. Overall, incorporating a to-do list into daily routines can lead to increased efficiency and a more organized approach to both work and personal responsibilities."

9.BIBILOGRAPHY

Android Developer Documentation:

URL: https://developer.android.com/docs

Description: The official Android Developer Documentation provides detailed information about Android app development, UI design, and bestpractices.

SQLiteDatabase - Android Developers:

URL: https://developer.android.com/reference/android/database/sqlite/SQLiteD atabase

Description: The official documentation for SQLiteDatabase in Android,which is commonly used for local data storage.

SQLite Documentation:

URL: https://www.sqlite.org/docs.html

Description: The official documentation for SQLite, the relational database management system used in Android.

10. ANNEXURES

Document Organization:

Create a system for organizing your main to-do list and associated annexures. This could involve creating folders or sections for different projects or types of tasks.
File Naming Conventions:
Establish a consistent naming convention for your annexures to make them easy to identify and associate with specific tasks on your to-do list.

Digital Storage:

If your annexures are digital documents, consider using cloud storage services or a shared drive for easy access and collaboration. This ensures that all team members have access to the relevant materials.

Linking to Tasks:

In your to-do list, you can include links or references to the corresponding annexures. This helps you quickly access additional information when working on a specific task.

Version Control:

If annexures are subject to updates or revisions, implement a version control system to keep track of changes. This helps maintain the accuracy of information.

Collaboration and Sharing:

If you're working in a team, ensure that team members have access to the annexures they need. This could involve sharing access to folders or sending updated documents as tasks progress.

11.Sample Code

```
package com.developerdepository.noted;

import static android.content.ContentValues.TAG;

import android.annotation.SuppressLint;

import android.content.ActivityNotFoundException;

import android.content.Intent;

import android.content.IntentSender;

import android.database.Cursor;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.drawable.ColorDrawable;

import android.net.Uri;

import android.os.AsyncTask;

import android.os.Bundle;

import android.provider.MediaStore;

import android.provider.Settings;

import android.speech.RecognizerIntent;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.util.Patterns;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;
```

```
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.view.ActionMode;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.core.content.ContextCompat;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;

import com.developerdepository.noted.adapters.NotesAdapter;
import  com.developerdepository.noted.database.NotesDatabase;
import com.developerdepository.noted.entities.Note;
import  com.developerdepository.noted.listeners.NotesListener;
import com.github.dhaval2404.imagepicker.ImagePicker;
import com.google.android.material.bottomappbar.BottomAppBar;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.navigation.NavigationView;
```

```
import com.google.android.material.snackbar.Snackbar;
import com.google.android.play.core.appupdate.AppUpdateManager;
import com.google.android.play.core.appupdate.AppUpdateManagerFactory;
import com.google.android.play.core.install.InstallStateUpdatedListener;
import com.google.android.play.core.install.model.AppUpdateType;
import com.google.android.play.core.install.model.InstallStatus;
import com.google.android.play.core.install.model.UpdateAvailability;
import com.tapadoo.alerter.Alerter;

import net.yslibrary.android.keyboardvisibilityevent.KeyboardVisibilityEvent;
import net.yslibrary.android.keyboardvisibilityevent.util.UIUtil;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import dev.shreyaspatil.MaterialDialog.MaterialDialog;
import hotchemi.android.rate.AppRate;
import maes.tech.intentanim.CustomIntent;

public class MainActivity extends AppCompatActivity implements
NotesListener, NavigationView.OnNavigationItemSelectedListener {

    private DrawerLayout drawerLayout;
    private NavigationView navigationView;
```

```
private ConstraintLayout contentView;
private ImageView imageEmpty;
private ImageButton navigationMenu;
private TextView textEmpty;
private EditText inputSearch;
private RecyclerView notesRecyclerView;
private BottomAppBar bottomAppBar;
private FloatingActionButton addNoteFloatingBtn;

private List<Note>  noteList;
private NotesAdapter notesAdapter;

public static final int REQUEST_CODE_ADD_NOTE = 1;
public static final int REQUEST_CODE_UPDATE_NOTE = 2;
public static final int REQUEST_CODE_SHOW_NOTES = 3;
public static final int REQUEST_CODE_TAKE_PHOTO = 4;
public static final int REQUEST_CODE_SELECT_IMAGE = 5;
public static final int REQUEST_CODE_VOICE_NOTE = 6;

private int noteClickedPosition = -1;
private androidx.appcompat.view.ActionMode actionMode;

private AlertDialog dialogAddURL;
private AlertDialog dialogAddImage;

private static final float END_SCALE = 0.8f;
```

```
    private AppUpdateManager mAppUpdateManager;
    private InstallStateUpdatedListener installStateUpdatedListener;
    private int RC_APP_UPDATE = 123;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

getWindow().setNavigationBarColor(ContextCompat.getColor(MainActivity.this
, R.color.colorQuickActionsBackground));

        AppRate.with(MainActivity.this)
                .setInstallDays(1)
                .setLaunchTimes(3)
                .setRemindInterval(1)
                .setShowLaterButton(true)
                .setShowNeverButton(false)
                .monitor();
        AppRate.showRateDialogIfMeetsConditions(MainActivity.this);

        initViews();
        setNavigationMenu();
        setActionOnViews();

        getNotes(REQUEST_CODE_SHOW_NOTES, false);
    }
```