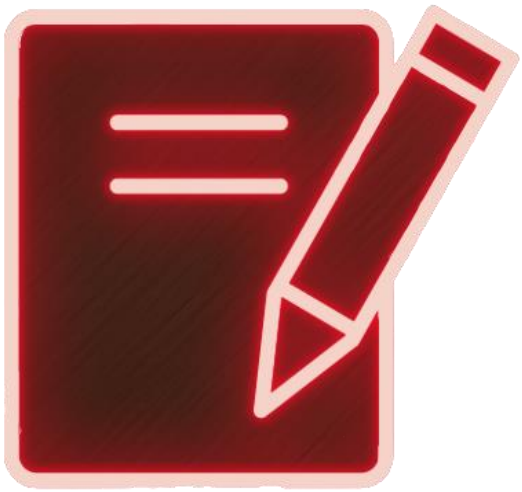
Keep Notes App Documentation



Submitted to:

Mr. Khurruam Yousaf Tehseen

Group Members:

Ali Raza Khan BSCS-SP-2021-094
Adnan Naeem BSCS-SP-2021-079
Zain-ul-Abidin BSCS-SP-2021-067

Developed Using:

Keep Notes App Documentation

1. Introduction

Welcome to the documentation for the Keep Notes app developed using Android Studio and Kotlin. This document provides a comprehensive guide to understanding the purpose, functionality, and implementation details of the Keep Notes application.

2. Abstract

Keep Notes is a simple yet powerful note-taking application designed for Android devices. The app aims to provide users with an intuitive and efficient way to create, organize, and manage their notes. Built using Kotlin programming language and Android Studio, Keep Notes leverages the latest technologies to deliver a seamless and responsive user experience.

3. Problem Statement

In the digital age, individuals often find themselves in need of a reliable and convenient tool for jotting down thoughts, ideas, and important information on the go. Traditional note-taking methods are often cumbersome, leading to disorganization and difficulty in retrieval. Keep Notes addresses these challenges by offering a modern solution that combines simplicity with advanced features.

4. Solution

4.1 Features

Note Creation: Users can create text-based notes with ease, allowing them to capture thoughts quickly.

Prioritization: The app supports prioritization of notes, enabling users to organize their content based on priority of **Low, Medium** or **High.**

Search Functionality: A powerful search feature helps users locate specific notes efficiently, enhancing accessibility.

Timestamp: Keep Notes allows users to set reminders for their notes by associating a timestamp with each note.

4.2 Architecture

The application follows the **Model-View-ViewModel (MVVM)** architecture, promoting a clean and modular codebase. This architecture enhances maintainability, testability, and scalability.

4.3 Technologies Used

Kotlin: The app is developed using the Kotlin programming language, known for its conciseness and expressiveness.

Android Studio: The official integrated development environment (IDE) for Android app development is used for building and debugging the application.

SQLite: Keep Notes relies on **SQLite** for local data storage, ensuring efficient data synchronization and secure note storage on the device without the need for external cloud services.

5. Scope

Keep Notes is designed for users who require a straightforward and efficient note-taking application. Its features cater to a broad audience, from students and professionals to anyone looking for a digital tool to capture and organize their thoughts. The app's clean interface ensures that it is accessible to users of all technical levels.

6. Scalability

The application is built with scalability in mind, both in terms of functionality and user base. Future updates can introduce additional features, such as multimedia note support, collaboration features, or integration with third-party services. The modular architecture facilitates the seamless incorporation of new modules and enhancements.

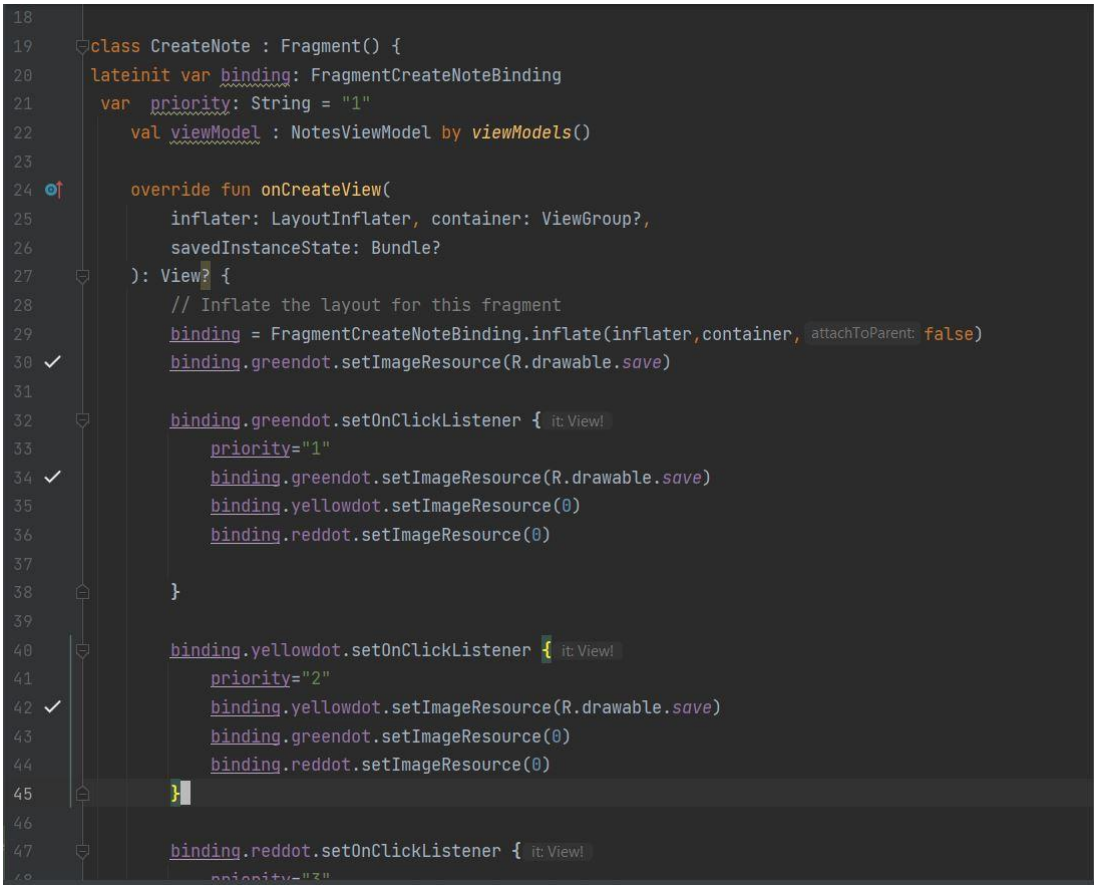
7. Conclusion

Keep Notes represents a significant step forward in addressing the note-taking needs of modern users. By combining simplicity with powerful features, the app aims to enhance productivity and organization in users' lives. The use of Kotlin and the MVVM architecture ensures a robust and maintainable codebase, laying the foundation for future enhancements and improvements.

8. Code Snippets

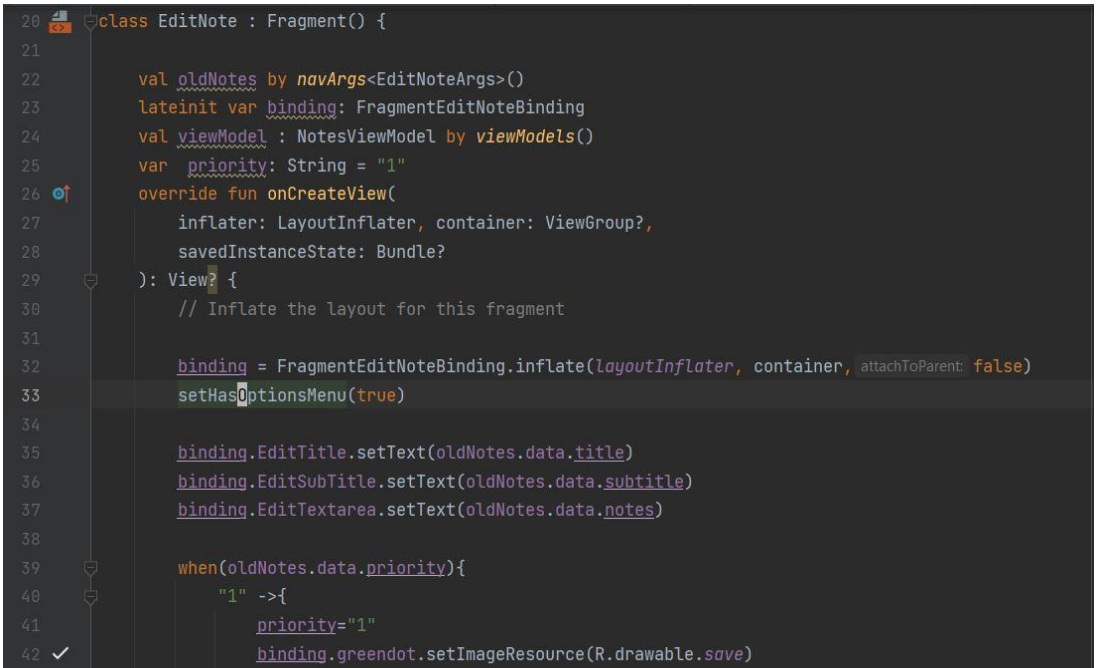
8.1 Backend

8.11 Example of Note Creation

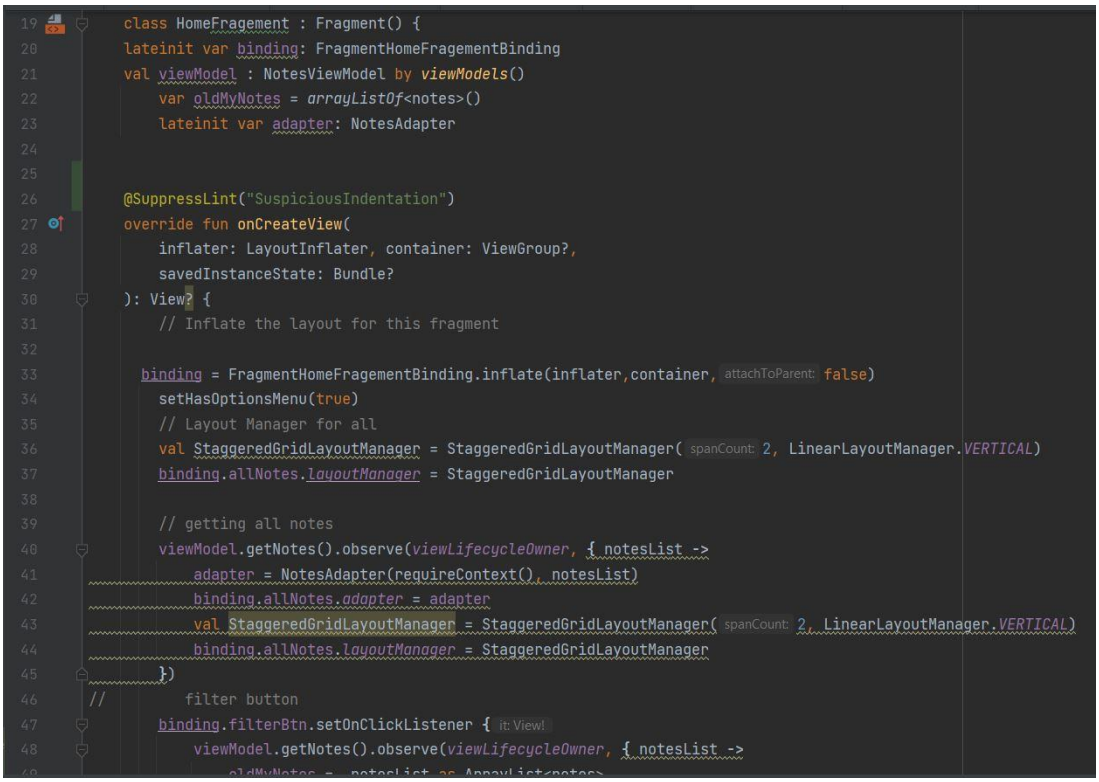


```
class CreateNote : Fragment() {
lateinit var binding: FragmentCreateNoteBinding
 var  priority: String = "1"
    val viewModel : NotesViewModel by viewModels()

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        binding = FragmentCreateNoteBinding.inflate(inflater,container, attachToParent: false)
        binding.greendot.setImageResource(R.drawable.save)

        binding.greendot.setOnClickListener { it: View!
            priority="1"
            binding.greendot.setImageResource(R.drawable.save)
            binding.yellowdot.setImageResource(0)
            binding.reddot.setImageResource(0)

        }

        binding.yellowdot.setOnClickListener { it: View!
            priority="2"
            binding.yellowdot.setImageResource(R.drawable.save)
            binding.greendot.setImageResource(0)
            binding.reddot.setImageResource(0)
        }

        binding.reddot.setOnClickListener { it: View!
```

8.12 Example of Note Editing

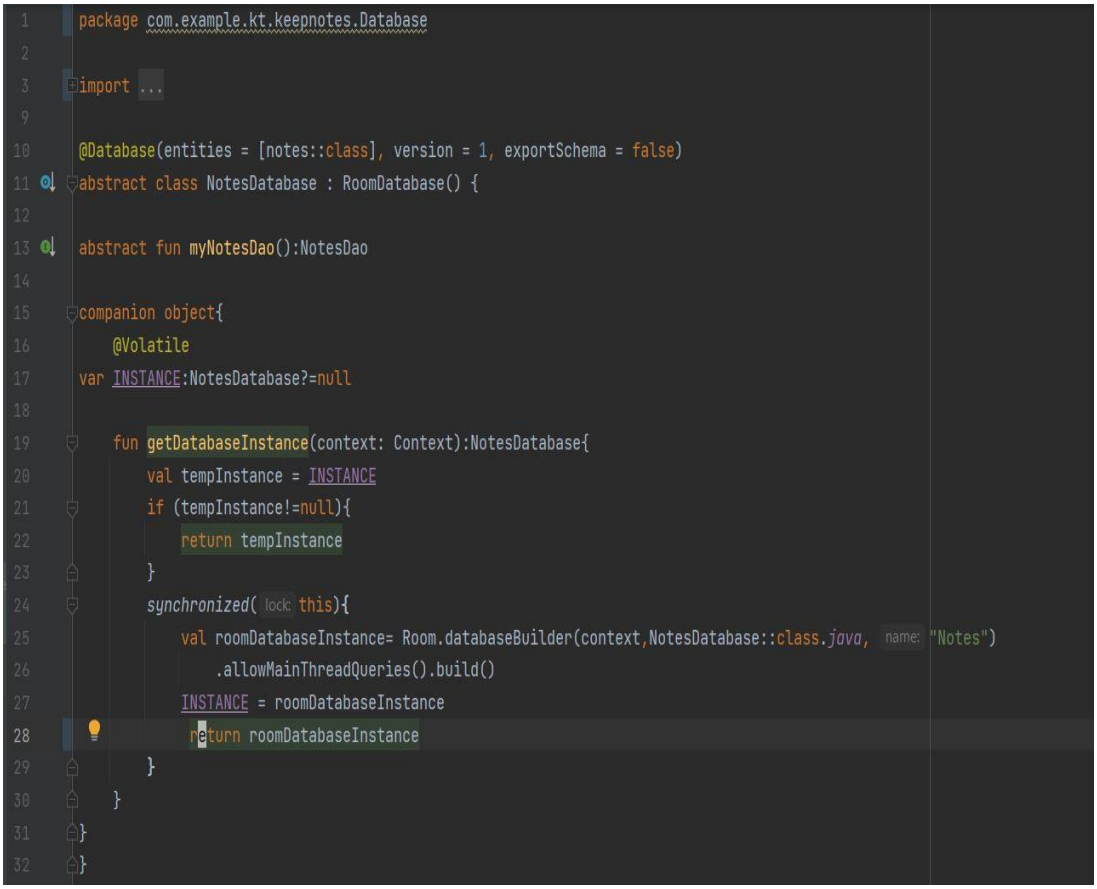


```
class EditNote : Fragment() {

    val oldNotes by navArgs<EditNoteArgs>()
    lateinit var binding: FragmentEditNoteBinding
    val viewModel : NotesViewModel by viewModels()
    var  priority: String = "1"
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment

        binding = FragmentEditNoteBinding.inflate(layoutInflater, container, attachToParent: false)
        setHasOptionsMenu(true)

        binding.EditTitle.setText(oldNotes.data.title)
        binding.EditSubTitle.setText(oldNotes.data.subtitle)
        binding.EditTextarea.setText(oldNotes.data.notes)

        when(oldNotes.data.priority){
            "1" ->{
                priority="1"
                binding.greendot.setImageResource(R.drawable.save)
```



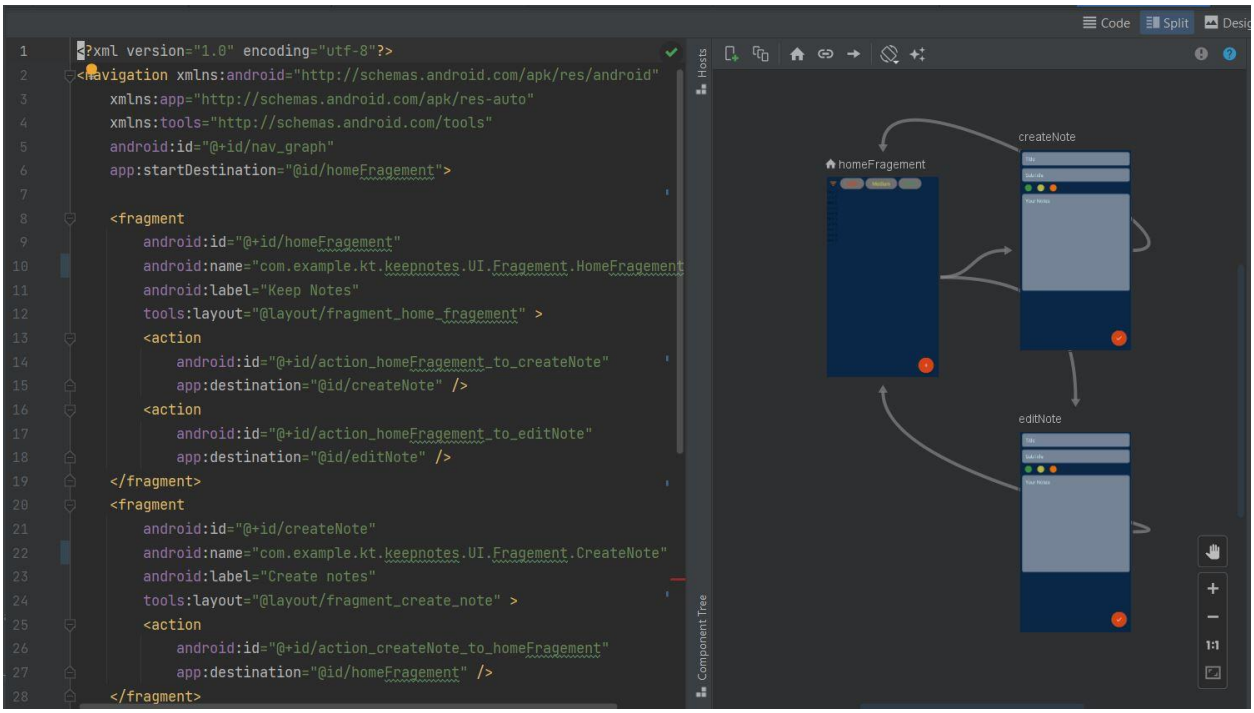
```
class HomeFragement : Fragment() {
lateinit var binding: FragmentHomeFragementBinding
val viewModel : NotesViewModel by viewModels()
    var oldMyNotes = arrayListOf<notes>()
    lateinit var adapter: NotesAdapter


@SuppressLint("SuspiciousIndentation")
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    // Inflate the layout for this fragment

  binding = FragmentHomeFragementBinding.inflate(inflater,container, attachToParent: false)
    setHasOptionsMenu(true)
    // Layout Manager for all
    val StaggeredGridLayoutManager = StaggeredGridLayoutManager( spanCount: 2, LinearLayoutManager.VERTICAL)
    binding.allNotes.layoutManager = StaggeredGridLayoutManager

    // getting all notes
    viewModel.getNotes().observe(viewLifecycleOwner, { notesList ->
        adapter = NotesAdapter(requireContext(), notesList)
        binding.allNotes.adapter = adapter
        val StaggeredGridLayoutManager = StaggeredGridLayoutManager( spanCount: 2, LinearLayoutManager.VERTICAL)
        binding.allNotes.layoutManager = StaggeredGridLayoutManager
    })
//          filter button
    binding.filterBtn.setOnClickListener { it: View!
        viewModel.getNotes().observe(viewLifecycleOwner, { notesList ->
```

8.14 Example of Notes Database



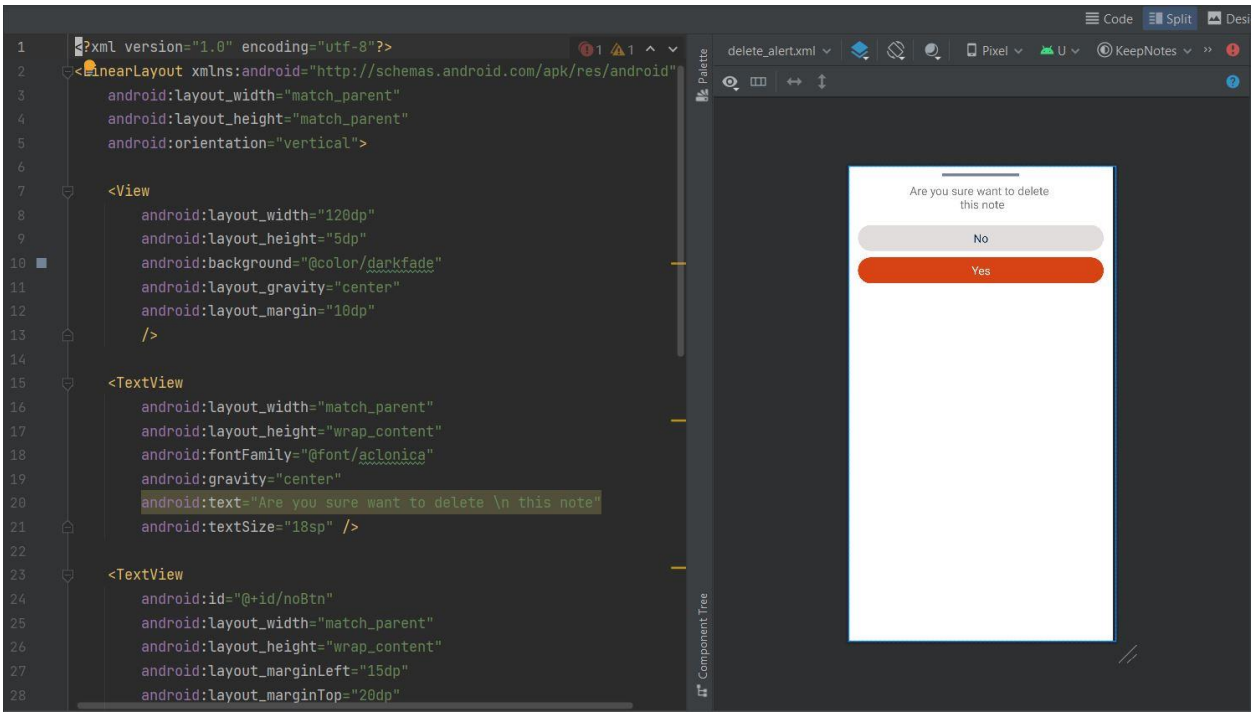
```
package com.example.kt.keepnotes.Database

import ...

@Database(entities = [notes::class], version = 1, exportSchema = false)
abstract class NotesDatabase : RoomDatabase() {

abstract fun myNotesDao():NotesDao

companion object{
    @Volatile
var INSTANCE:NotesDatabase?=null

    fun getDatabaseInstance(context: Context):NotesDatabase{
        val tempInstance = INSTANCE
        if (tempInstance!=null){
            return tempInstance
        }
        synchronized( lock: this){
            val roomDatabaseInstance= Room.databaseBuilder(context,NotesDatabase::class.java, name: "Notes")
                .allowMainThreadQueries().build()
            INSTANCE = roomDatabaseInstance
            return roomDatabaseInstance
        }
    }
}
}
```

8.21 Example of Nav Graph



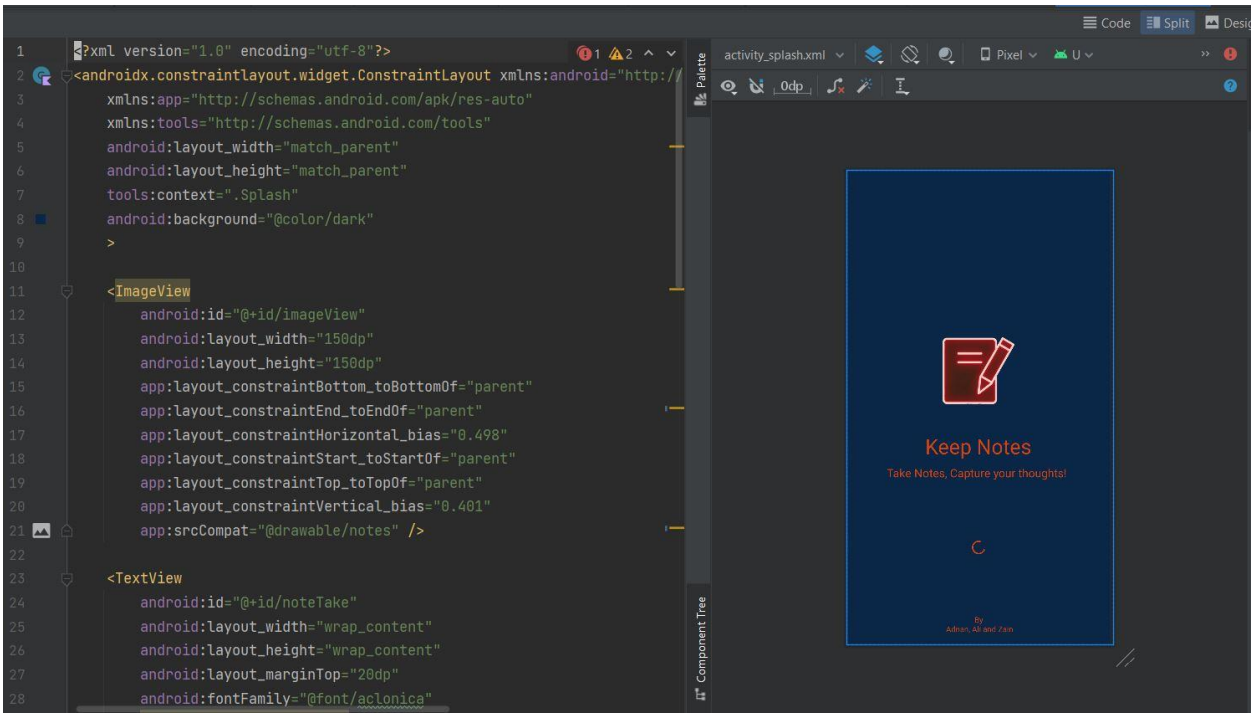

```xml
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nav_graph"
    app:startDestination="@id/homeFragement">

    <fragment
        android:id="@+id/homeFragement"
        android:name="com.example.kt.keepnotes.UI.Fragement.HomeFragement"
        android:label="Keep Notes"
        tools:layout="@layout/fragment_home_fragement" >
        <action
            android:id="@+id/action_homeFragement_to_createNote"
            app:destination="@id/createNote" />
        <action
            android:id="@+id/action_homeFragement_to_editNote"
            app:destination="@id/editNote" />
    </fragment>
    <fragment
        android:id="@+id/createNote"
        android:name="com.example.kt.keepnotes.UI.Fragement.CreateNote"
        android:label="Create notes"
        tools:layout="@layout/fragment_create_note" >
        <action
            android:id="@+id/action_createNote_to_homeFragement"
            app:destination="@id/homeFragement" />
    </fragment>
```

8.22 Example of Delete Function




```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <View
        android:layout_width="120dp"
        android:layout_height="5dp"
        android:background="@color/darkfade"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="@font/aclonica"
        android:gravity="center"
        android:text="Are you sure want to delete \n this note"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/noBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="20dp"
```

8.23 Example of Splash Screen


Code
Split
activity_splash.xml
Pixel
Palette
Component Tree
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Splash"
android:background="@color/dark"
>
<ImageView
android:id="@+id/imageView"
android:layout_width="150dp"
android:layout_height="150dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.401"
app:srcCompat="@drawable/notes" />
<TextView
android:id="@+id/noteTake"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:fontFamily="@font/aclonica"
Keep Notes
Take Notes, Capture your thoughts!
By
Adnan, Ali and Zain