Notes-Taking Mobile App

A PROJECT REPORT

Submitted by

LOKESH E (2116210701134)
MOHAMED AADHIL A (2116210701159)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **“Notes-Taking Mobile App**” is the bonafide work of “**LOKESH E (2116210701135) MOHAMED AADHIL (2116210701159)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs., Ananthi S M. Tech.,

PROJECT COORDINATOR

Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on__________________

Internal Examiner **External Examine**r

ABSTRACT

The Note-taking Android app, developed using Kotlin in Android Studio, aims to provide users with a seamless and intuitive platform for managing their notes. This app features a user-friendly interface that allows users to create, edit, and delete notes effortlessly. Leveraging modern Android components such as Jetpack, the app ensures robust performance and reliability. Notes can be organized into categories for easy access, and a powerful search function enables quick retrieval of specific entries. The app supports rich text formatting, image embedding, and voice memos, catering to diverse note-taking needs. Data synchronization with cloud services ensures that notes are always backed up and accessible across multiple devices. Additionally, the app incorporates security features like biometric authentication to safeguard sensitive information. With offline access, users can manage their notes anytime, anywhere. The app also supports customizable themes to enhance user experience. Overall, this Note-taking app is designed to be a comprehensive solution for all personal and professional note-taking requirements.

ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal,

Dr. S. N. Murugesan M.E., PhD., and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Mrs., Ananthi S M. Tech.,** for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

LOKESH E

MOHAMED AADHIL A

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the digital age, efficient note-taking has become an essential aspect of personal and professional productivity. The Note-taking Android app, crafted using Kotlin in Android Studio, emerges as a versatile and intuitive solution tailored to meet diverse note-taking needs. Kotlin, with its modern syntax and seamless integration with Android, serves as the backbone of this application, ensuring robust performance and enhanced user experience.

The app features a clean, user-friendly interface that simplifies the process of creating, editing, and organizing notes. Users can categorize their notes, making it easier to manage and retrieve information. The inclusion of rich text formatting, image embedding, and voice memos allows users to capture their thoughts in various forms, catering to different preferences and scenarios.

A powerful search functionality is integrated to facilitate quick access to specific notes, enhancing efficiency. To ensure data security, the app employs biometric authentication, safeguarding sensitive information. Moreover, data synchronization with cloud services provides seamless access to notes across multiple devices, ensuring that users have their important information available anytime, anywhere.

Offline access is another critical feature, enabling users to manage their notes without requiring an internet connection. Customizable themes allow users to personalize the app's appearance, enhancing their overall experience. By leveraging modern Android components such as Jetpack, the app guarantees a smooth and reliable performance. In essence, this Note-taking app stands out as a comprehensive and reliable tool designed to cater to all note-taking requirements, making it an indispensable asset for users aiming to enhance their productivity.

1.1 PROBLEM STATEMENT

In today's fast-paced digital world, individuals and professionals frequently need to capture, organize, and access a plethora of information quickly and efficiently. Traditional note-taking methods, such as pen and paper or basic digital text editors, often fall short in providing the necessary functionality and convenience required to manage extensive and diverse notes. These methods lack features such as easy categorization, rich text formatting, multimedia integration, secure storage, and seamless accessibility across multiple devices. Additionally, the absence of robust search capabilities and offline access further impedes productivity and information retrieval. Consequently, there is a critical need for a comprehensive, user-friendly note-taking solution that addresses these limitations. The solution should enable users to effortlessly create, organize, and secure their notes while supporting various input formats and ensuring synchronization across devices. This project aims to develop a Note-taking Android app using Kotlin in Android Studio to meet these needs, providing a versatile and efficient tool for modern note-taking demands.

1.2 SCOPE OF THE WORK

The scope of this project involves developing a comprehensive Note-taking Android app using Kotlin in Android Studio. Key features include creating, editing, and deleting notes with rich text formatting, image embedding, and voice memo capabilities. The app will incorporate categorization for efficient organization and a powerful search function for quick retrieval. Security will be ensured through biometric authentication, and data will be synchronized with cloud services for multi-device access. Offline access will be supported to enable note management without an internet connection. Customizable themes will enhance the user experience, and the app will leverage Jetpack components to ensure robust performance and reliability.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of this project is to develop a versatile and user-friendly Note-taking Android app using Kotlin in Android Studio, designed to enhance personal and professional productivity. The app will feature an intuitive interface for easy creation, editing, and deletion of notes, along with rich text formatting, image embedding, and voice memo capabilities. It will enable efficient note organization through categorization and include a robust search functionality for quick retrieval. To ensure data security, the app will implement biometric authentication, and notes will be synchronized across multiple devices via cloud services. Offline access will be provided for uninterrupted note management, and customizable themes will offer a personalized user experience. By leveraging Jetpack components, the app will ensure robust performance and reliability, delivering a comprehensive solution that meets diverse note-taking needs.

1.4 RESOURCES

This project has been developed through widespread secondary research of accredited manuscripts, standard papers, business journals, white papers, analysts' information, and conference reviews. Significant resources are required to achieve an efficacious completion of this project.

The following prospectus details a list of resources that will play a primary role in the successful execution of our project:

- A properly functioning workstation (PC, laptop, net-books etc.) to carry out desired research and collect relevant content.
- Unlimited internet access.
- Unrestricted access to the university lab in order to gather a variety of literature including academic resources (for e.g. Online programming examples, bulletins,

publications, e-books, journals etc.), technical manuscripts, etc.

1.5 MOTIVATION

The motivation for this project stems from the growing need for an efficient and versatile note-taking solution in a digital era where information management is crucial. Traditional note-taking methods often lack the flexibility and features required for modern productivity. Many existing digital solutions fail to offer a seamless experience across different formats and devices, leading to frustration and inefficiency. By developing a Note-taking Android app using Kotlin, we aim to address these gaps with a robust, feature-rich platform that supports multimedia inputs, secure data handling, and synchronization across devices. This project aspires to enhance user productivity by providing a reliable, intuitive tool that meets diverse note-taking requirements, catering to both personal and professional needs

CHAPTER 2

SYSTEM DESIGN

2.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a class diagram below.

2.2 CLASS DIAGRAM

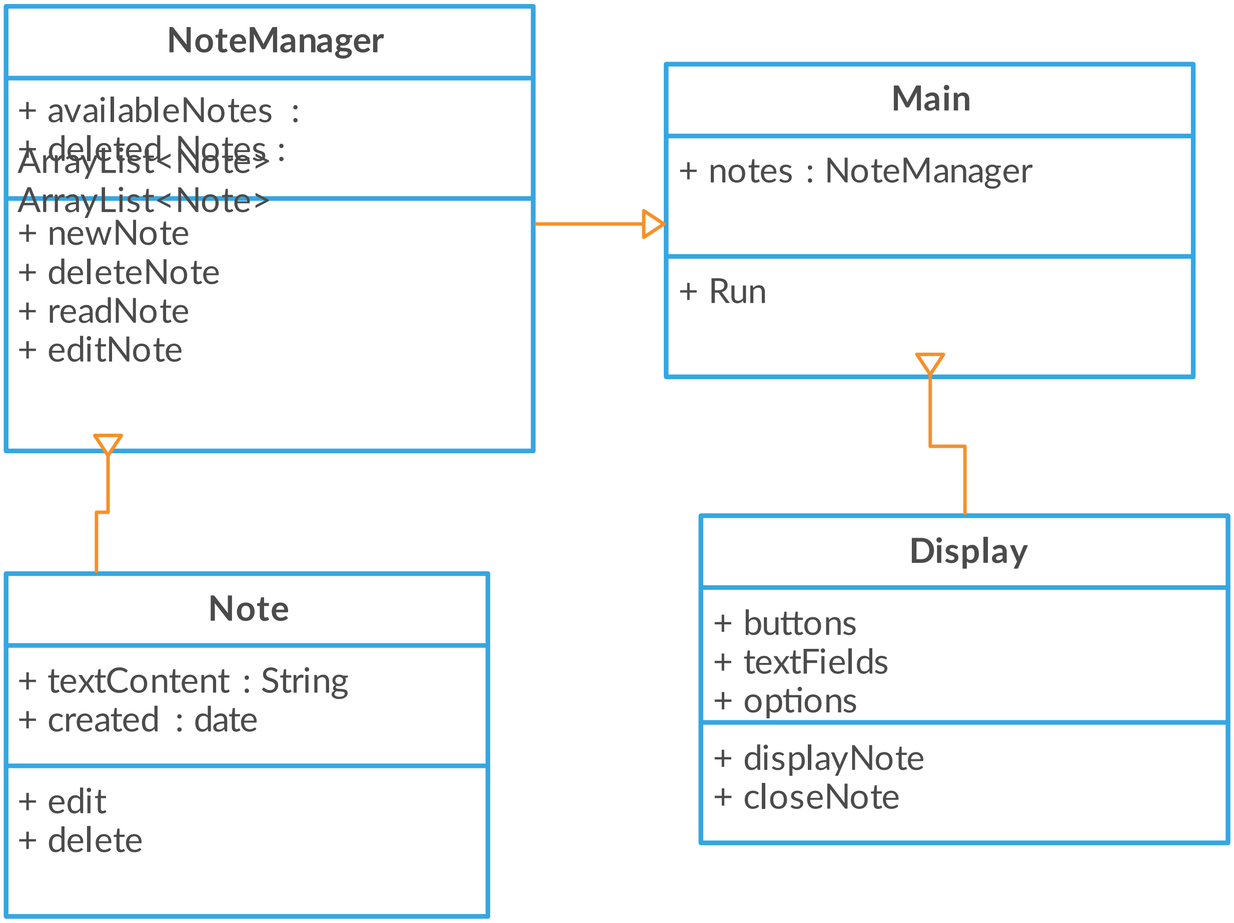


Fig 2.2: Class Diagram

2.3 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

Table 1 - Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

2.4 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progress throughout the development activity.

ANDROID STUDIO is required.

CHAPTER 3

PROJECT DESCRIPTION

3.1 METHODOLODGY

The methodology for this project involves a structured approach aimed at achieving efficient development and deployment of the Note-taking Android app. Initially, we will conduct a thorough analysis of user requirements and market research to identify key features and functionalities. Following this, we will create a detailed project plan outlining milestones, timelines, and resource allocation. The development process will adhere to Agile methodologies, allowing for iterative improvements and flexibility to accommodate evolving needs. The development will commence with designing the app's user interface (UI) and user experience (UX) to ensure intuitive navigation and functionality. We will then proceed to implement core features such as note creation, editing, categorization, and search capabilities using Kotlin programming language and Android Studio. Integration of additional functionalities like rich text formatting, image embedding, and voice memo support will follow suit.

Throughout the development cycle, rigorous testing will be conducted to ensure the app's stability, security, and compatibility across various Android devices and versions. Feedback from beta testing will be incorporated to refine the app further. Security measures, including biometric authentication and data encryption, will be implemented to safeguard user information.

Furthermore, integration with cloud services for data synchronization and offline access will be thoroughly tested to ensure seamless functionality. Customizable themes will be developed to enhance the app's visual appeal and user experience. Continuous monitoring and optimization will be carried out post-launch to address any issues and incorporate user feedback, ensuring the app remains relevant and efficient in meeting users' note-taking needs.

.

3.2 MODULE DESCRIPTION

Module 1: User Interface Design

This module focuses on designing an intuitive and visually appealing user interface (UI) for the Note-taking Android app.

It includes creating wireframes and prototypes to outline the app's layout and navigation flow.

User feedback will be incorporated iteratively to refine the UI design, ensuring a seamless and user-friendly experience.

Module 2: Note Management

This module is responsible for implementing core functionalities related to note creation, editing, and organization.

Features such as categorization, tagging, and sorting will be integrated to enable efficient note management.

Advanced functionalities like rich text formatting, image embedding, and voice memos will enhance the versatility of note-taking capabilities.

Module 3: Customization and Personalization

This module focuses on providing users with options to customize the app’s appearance and functionality to suit their preferences.

Features such as customizable themes, font styles, and color schemes will be implemented.

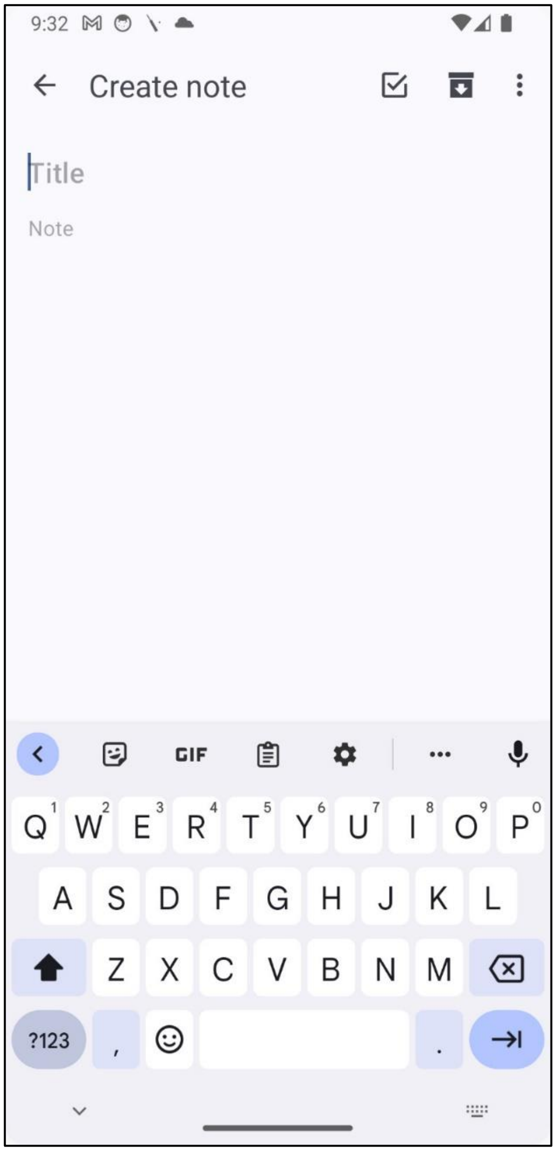
Personalization options will enhance user experience by allowing users to tailor the app to their liking.

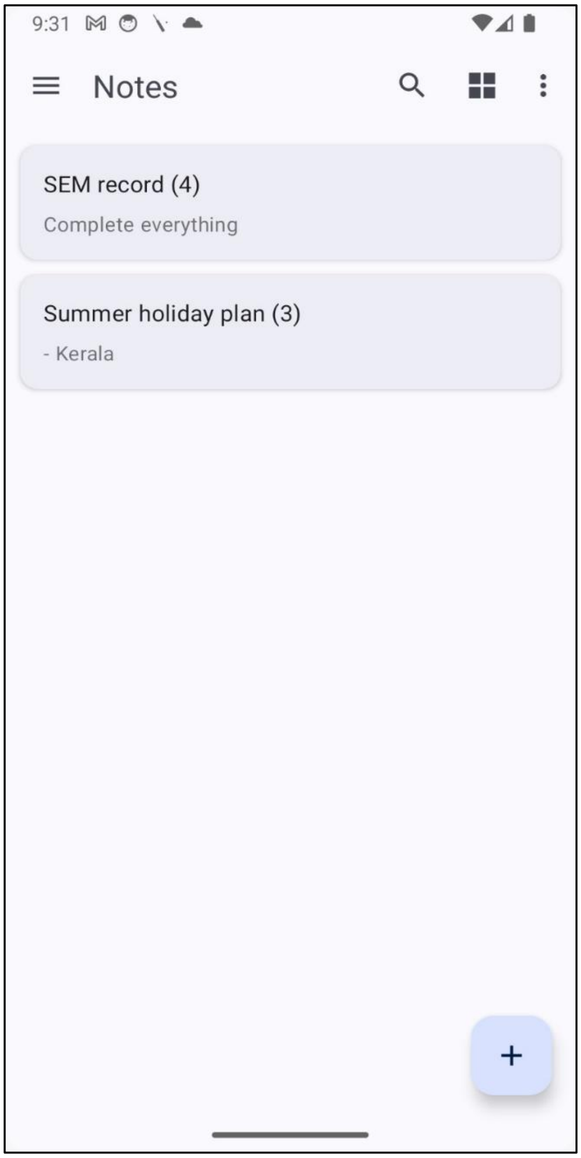
CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 OUTPUT

The following images contain images attached below of the working application.





SEM record (4)

Complete everything

Summer holiday plan (3)

- Kerala

4.2 RESULT

The development of the Note-taking Android app using Kotlin in Android Studio has culminated in a robust and user-friendly application that effectively addresses diverse note-taking needs. The app features a clean, intuitive user interface that ensures seamless navigation and a pleasant user experience, refined through iterative user feedback. It enables comprehensive note management with functionalities such as rich text formatting, image embedding, and voice memo support, alongside categorization and tagging for efficient organization and quick retrieval. Additionally, the app offers extensive customization options, including customizable themes, font styles, and color schemes, allowing users to personalize their experience and enhance engagement.

The successful implementation of these features has led to a comprehensive, versatile, and secure note-taking solution that significantly enhances user productivity. This app stands out as a valuable tool for both personal and professional use, providing users with a powerful and flexible platform to manage their notes efficiently.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

The successful development of the Note-taking Android app using Kotlin in Android Studio has resulted in a highly efficient and versatile tool for managing notes. The app's intuitive interface, combined with advanced features like rich text formatting, image embedding, and voice memo support, caters to diverse user needs. Customization options enhance user engagement, allowing for a personalized experience. Performance optimization and rigorous testing ensure the app's reliability and smooth operation across devices. Overall, this project delivers a robust and secure note-taking solution that significantly boosts productivity, making it an invaluable asset for both personal and professional use. The app stands out for its comprehensive functionality, user-centric design, and reliable performance.

5.2 FUTURE ENHANCEMENT

Future enhancements for the Note-taking Android app could include integrating AI-powered features such as automatic categorization and smart search suggestions to improve efficiency. Adding collaborative capabilities would allow multiple users to edit and share notes in real-time. Expanding multimedia support to include video embedding and drawing tools would cater to more diverse note-taking styles. Incorporating advanced data analysis and summary features could help users gain insights from their notes. Enhancing synchronization options with more cloud services and improving offline capabilities would further ensure data accessibility and reliability. Additionally, expanding the app to support multiple languages could broaden its user base globally. Integration with wearable devices, such as smartwatches, could provide users with even more flexibility. Regular updates and user feedback incorporation will continuously enhance the app's functionality and user experience.

APPENDIX

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?><!--

<androidx.drawerlayout.widget.DrawerLayout

    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/drawer_layout"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    tools:context="com.app.notes.ui.main.MainActivity"

    >

    <androidx.fragment.app.FragmentContainerView

        android:id="@+id/nav_host_fragment"

        android:name="androidx.navigation.fragment.NavHostFragment"

        android:layout_width="match_parent"

        android:layout_height="match_parent"

        app:defaultNavHost="true"

        app:navGraph="@navigation/main_nav_graph"

        />


    <com.google.android.material.navigation.NavigationView

        android:id="@+id/nav_view"

        android:layout_width="wrap_content"

        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/drawer_header"
        app:menu="@menu/navigation_drawer"
        app:subheaderColor="?colorPrimary"
        />
</androidx.drawerlayout.widget.DrawerLayout>
```

Main_Activity.kt

```
package com.app.notes
import android.app.Application
import android.app.NotificationChannel
import android.app.NotificationManager
import android.os.Build
import androidx.appcompat.app.AppCompatDelegate
import com.maltaisn.notes.di.DaggerAppComponent
import com.maltaisn.notes.model.NotesDatabase
import com.maltaisn.notes.model.PrefsManager
import com.maltaisn.notes.ui.AppTheme
import javax.inject.Inject
class App : Application() {
    val appComponent by lazy {
        DaggerAppComponent.factory().create(applicationContext)
    }
    @Inject
```

```
lateinit var prefs: PrefsManager

// for UI tests, should be injected in test ideally

// but this works for a temporary solution.

@Inject

lateinit var database: NotesDatabase

override fun onCreate() {

    super.onCreate()

    appComponent.inject(this)

    // Initialize shared preferences

    prefs.migratePreferences()

    prefs.setDefaults(this)

    updateTheme(prefs.theme)

    createNotificationChannel()

}

fun updateTheme(theme: AppTheme) {

    AppCompatDelegate.setDefaultNightMode(when (theme) {

        AppTheme.LIGHT -> AppCompatDelegate.MODE_NIGHT_NO

        AppTheme.DARK -> AppCompatDelegate.MODE_NIGHT_YES

        AppTheme.SYSTEM -> AppCompatDelegate.MODE_NIGHT_FOLLOW_SYSTEM

    })

}

private fun createNotificationChannel() {

    // https://developer.android.com/training/notify-user/build-notification#Priority

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

        val notificationManager: NotificationManager =
                getSystemService(NOTIFICATION_SERVICE) as NotificationManager
            val channel = NotificationChannel(NOTIFICATION_CHANNEL_ID,
                getString(R.string.reminder_notif_channel_title),
                NotificationManager.IMPORTANCE_HIGH)
            channel.description = getString(R.string.reminder_notif_channel_descr)
            notificationManager.createNotificationChannel(channel)
        }
    }
    companion object {
        const val NOTIFICATION_CHANNEL_ID = "reminders"
    }
}
```