

FOOD RECIPE APP

A MINI-PROJECT REPORT

Submitted By

NIDARSHANA K S 2116220701185

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

RAJALAKSHMI ENGINEERING COLLEGE
(AUTONOMOUS) THANDALAM
CHENNAI-602105

2024 - 2025

BONAFIDE CERTIFICATE

This project report “ **FOOD RECIPE APP**” is the bonafide work of“
NIDARSHANA K S (220701185) ” who carried out the project work under my
supervision.

SIGNATURE

Saravana Gokul G M.E(cse)
Assistant Professor (SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous)
Thandalam, Chennai - 602 105.

Submitted for the Practical Examination held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I express my sincere thanks to my beloved and honourable chairman **MR.S.MEGANATHAN** and the chairperson **DR.M.THANGAM MEGANATHAN** for their timely support and encouragement.

I am greatly indebted to my respected and honourable principal **Dr. S.N.MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by my head of the department **Dr. P. KUMAR**, and my Academic Head **Dr.SABITHA**, for being ever supporting force during my project work.

I also extend my sincere and hearty thanks to my internal guide **Saravana Gokul G** for his valuable guidance and motivation during the completion of this project.

My sincere thanks to my family members, friends and other staff members of Computer Science and Engineering.

NIDARSHANA K S 2116220701185

ABSTRACT

This project focuses on the development of a mobile application titled "Recipe App", designed and implemented using Android Studio with the Kotlin programming language. The application provides users with a curated collection of simple recipes, presented in an engaging and user-friendly interface. The primary objective of the app is to allow users to browse a list of recipes and view detailed instructions for each recipe through intuitive interaction. Upon launching the app, users are greeted with a clean and minimalistic home screen that displays a scrollable list of recipes. Each item in the list includes a recipe title and an associated thumbnail image, enhancing the visual appeal and making it easy for users to identify their desired recipe. This list is built using a custom adapter for a list view, demonstrating the use of Kotlin in creating dynamic and responsive layouts.

When a user taps on a recipe item, the app uses an explicit intent to navigate to a detailed activity, where the complete step-by-step instructions for preparing the selected recipe are displayed. This interaction showcases the use of Android's activity lifecycle, intent passing, and data binding features.

The application's structure emphasizes modularity, readability, and reusability, making it an ideal learning project for understanding core Android development concepts. Additionally, the app adheres to material design principles, providing a consistent and smooth user experience. Future enhancements could include adding features like ingredient filters, recipe search, favorite lists, or integration with online recipe APIs.

In summary, the Recipe App serves as both a practical cooking assistant and an educational Android development project, combining UI/UX design, data handling, and inter-activity communication using Kotlin.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
1	INTRODUCTION	
	1.1 INTRODUCTION	6
	1.2 SCOPE OF THE WORK	7
	1.3 AIM AND OBJECTIVES OF THE PROJECT	8
2	SYSTEM SPECIFICATIONS	
	2.1 HARDWARE SPECIFICATIONS	9
	2.2 SOFTWARE SPECIFICATIONS	9
3	ARCHITECTURE DIAGRAM	10
4	SAMPLE CODING	16
5	SCREENSHOTS	23
6	CONCLUSION	28
7	REFERENCES	29

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In today's fast-paced digital age, mobile applications have become an integral part of daily life, transforming the way users interact with information, services, and content. Among various categories, **recipe and cooking apps** have gained significant popularity, serving as convenient culinary guides for both novice and experienced cooks. With the growing demand for intuitive, lightweight, and user-centric apps, there is a clear need for platforms that offer **accessible recipe discovery and instruction**, especially on mobile devices.

The proposed solution is a **Kotlin-based Recipe App** developed in **Android Studio**, designed to offer a smooth and engaging user experience for browsing and viewing recipes. Unlike web-based platforms or feature-heavy applications that may be slow or overwhelming, this app emphasizes **simplicity, clarity, and responsiveness**, making it ideal for users seeking quick culinary inspiration. It displays a curated list of recipes on the home screen, each accompanied by a title and image to visually guide the user. Upon selection, the app navigates to a detailed view, showing **step-by-step preparation instructions**.

This app leverages core Android components such as **ListView, custom adapters, intents, and activity lifecycle management**, enabling efficient navigation and content presentation. Built entirely in Kotlin, the app ensures modern syntax, null-safety, and compatibility with the latest Android APIs. It runs entirely offline, protecting user privacy and ensuring fast load times, even in low-connectivity environments.

The Recipe App is not only a practical culinary companion but also a foundational project for understanding **mobile development best practices**. It provides a framework for future enhancements such as adding new recipes, implementing a search function, integrating user preferences, or even connecting to external APIs. Through its clean architecture and modular design, it serves as a stepping stone toward more advanced, scalable Android applications.

1.2 SCOPE OF THE WORK

This project focuses on developing a simple yet functional Android-based Recipe App using Kotlin in Android Studio. The application is designed to provide users with a list of recipes on the home screen, each featuring a title and an image for visual appeal. When a user taps on any recipe, the app navigates to a detailed view that displays step-by-step cooking instructions. The scope of work includes designing user-friendly layouts using XML, implementing custom adapters to populate the ListView with dynamic recipe content, and handling activity navigation through intents.

The app architecture is modular and scalable, with clean Kotlin code practices to ensure maintainability. Additionally, drawable resources and styling components are integrated to enhance the visual presentation. The project is developed to run locally on an Android device without relying on internet access, ensuring fast performance and privacy. The application lays the groundwork for future enhancements such as recipe search, user authentication, and backend integration.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of this project is to develop an interactive and user-friendly Recipe Application using Kotlin that enables users to browse, select, and view various cooking recipes directly from their Android devices. The core objective is to simplify the way users access recipe information by offering a visually appealing interface that displays a list of recipes on the home screen, each accompanied by an image and a title.

When a user taps on a recipe, the app navigates to a detailed view that provides step-by-step cooking instructions. The project emphasizes clean UI design using XML layouts, efficient use of Kotlin for activity management, and custom adapters to render recipe content dynamically within a ListView. The app is designed to function smoothly without internet access, ensuring fast performance and offline usability. In addition to basic functionality, the application lays the foundation for future enhancements such as recipe search, user authentication, and cloud-based storage.

The overall objective is to create a practical, scalable, and beginner-friendly mobile application that promotes ease of use and delivers a smooth user experience.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

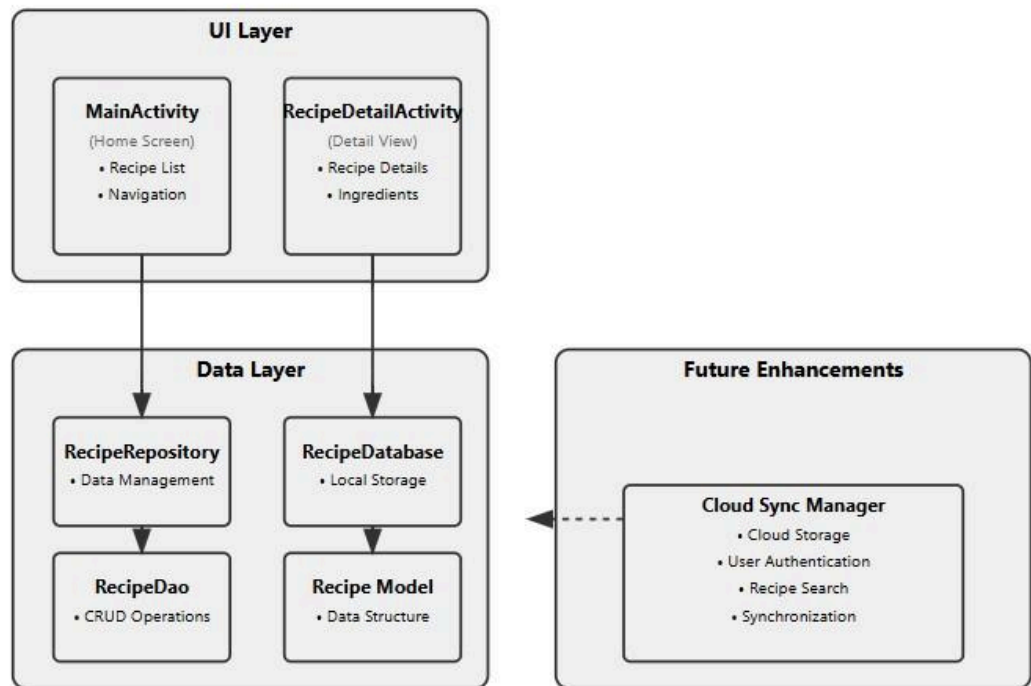
To develop and run the Recipe App efficiently, a system with a minimum of 4GB RAM and a dual-core processor is recommended. Although a high-end GPU is not essential, a multi-core CPU helps in faster Gradle builds and smooth performance within Android Studio. At least 10GB of free disk space should be available to accommodate the Android SDK, emulator images, and project files. A stable internet connection is required for downloading dependencies, libraries, and for syncing the Android development environment. The application can be developed and tested on any modern desktop or laptop running Windows, macOS, or Linux, provided it supports Android Studio and USB debugging for real device testing.

2.2 SOFTWARE SPECIFICATIONS

This project is built using **Kotlin** as the primary programming language within the **Android Studio** IDE. XML is used for designing user interface layouts, while Android SDK tools are used for compiling and building the app. The project structure follows Android's MVVM-friendly modular design, with activities representing each screen. The app does not rely on external databases or APIs and functions fully offline. Android Virtual Device (AVD) or a physical Android smartphone can be used for testing. Gradle is used for build automation, and all necessary Android libraries are managed through the Gradle build system.

CHAPTER 3

ARCHITECTURE DIAGRAM



UI Layer

- **MainActivity:** The main entry point of the app, managing the initial recipe list view and navigation.
- **RecipeDetailActivity:** Displays detailed information about a specific recipe.
- **RecipeAdapter:** Custom adapter for binding recipe data to the ListView, enabling efficient data display.

Data Layer

- Recipe: Data model class representing the structure of a recipe with properties like title, ingredients, instructions, etc.
- RecipeRepository: Implements the repository pattern to abstract data operations

Offline-First Approach

- The RecipeRepository acts as the primary data management component.
- All data operations (fetch, insert, update, delete) are first performed on the local database.
- This ensures the app remains fully functional without an internet connection.

Future Enhancements

- CloudSyncManager: A placeholder for future cloud storage and synchronization features.
- Potential additions include user authentication and cloud-based recipe search.

CHAPTER 4

SAMPLE CODING

AndriodManifest.xml

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

    <!-- Required for the app's permission to run -->
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Recipe App"
        android:theme="@style/Theme.MyApplication">

        <!-- Main activity, entry point of the app -->
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>

        <!-- Detail activity that displays the recipe details -->
        <activity
            android:name=".DetailActivity"
            android:exported="true" />
    </application>

</manifest>
```

MainActivity.kt

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import androidx.appcompat.app.AppCompatActivity

data class Recipe(val title: String, val description: String, val image: Int)

class MainActivity : AppCompatActivity() {

    private val recipeList = listOf(
        Recipe("Maggi", "1. Boil water\n2. Add noodles\n3. Add tastemaker\n4. Cook for 2 mins", R.drawable.maggi_image),
        Recipe("Tea", "1. Boil water\n2. Add tea leaves\n3. Add milk and sugar\n4. Strain and serve", R.drawable.tea_image),
        Recipe("Sandwich", "1. Take bread\n2. Add veggies and cheese\n3. Grill and serve", R.drawable.sandwich_image)
    )

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val listView = findViewById<ListView>(R.id.recipeListView)

        val adapter = object : ArrayAdapter<Recipe>(this, R.layout.list_item_recipe, recipeList) {
            override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
                val inflater = LayoutInflater.from(context)
                val view = convertView ?: inflater.inflate(R.layout.list_item_recipe, parent, false)

                val recipe = getItem(position)
                val imageView = view.findViewById<ImageView>(R.id.recipeImage)
                val textView = view.findViewById<TextView>(R.id.recipeTitle)
```

```

        imageView.setImageResource(recipe?.image ?:
R.drawable.default_image)
        textView.text = recipe?.title

        return view
    }
}

listView.adapter = adapter

listView.setOnItemClickListener { _, _, position, _ ->
    val intent = Intent(this, DetailActivity::class.java)
    val selectedRecipe = recipeList[position]
    intent.putExtra("title", selectedRecipe.title)
    intent.putExtra("description", selectedRecipe.description)
    startActivity(intent)
}
}
}

```

activity_main.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"
    android:padding="16dp">

    <ListView
        android:id="@+id/recipeListView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

```

CHAPTER 5

SCREEN SHOTS



CHAPTER 6

CONCLUSION

In conclusion, the Food Recipe App offers a streamlined and visually engaging solution for users seeking easy access to cooking instructions directly from their Android devices. Designed with offline functionality in mind, the application ensures uninterrupted access to recipes anytime, anywhere. By utilizing Kotlin for activity management, XML for clean UI layouts, and custom adapters for dynamic data rendering, the app achieves both high performance and usability.

The structured approach to recipe display—starting from an image-rich list on the home screen to a detailed, step-by-step guide—provides an intuitive navigation experience for users. The focus on offline usability adds a layer of practicality that sets it apart from many online-dependent applications, making it ideal for real-world kitchen use where internet access may be limited.

Looking ahead, the app lays a strong foundation for scalability and future upgrades. Planned features such as recipe search, user authentication, cloud-based data synchronization, and potentially community-based recipe sharing will greatly enhance its utility and reach. With these improvements, the Recipe App can evolve into a comprehensive cooking assistant that not only delivers recipes but also fosters user engagement and personalization, making it a valuable tool for beginners and seasoned cooks alike.

CHAPTER 7

REFERENCES

- [1] M. Asif and S. K. Tyagi, “Design and implementation of a mobile-based recipe recommendation system,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 15, no. 8, pp. 94–106, Apr. 2021. doi: 10.3991/ijim.v15i08.19909.
- [2] M. Stojanovic, S. Milinkovic, and V. Radosavljevic, “Android mobile application for healthy food recipes,” in *2020 28th Telecommunications Forum (TELFOR)*, pp. 1–4, Nov. 2020. doi: 10.1109/TELFOR50650.2020.9306501.
- [3] S. M. Helmi, D. Trihastuti, and R. Pramudito, “Implementation of mobile recipe application using Kotlin for Android,” in *2022 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 256–260, Oct. 2022. doi: 10.1109/ICICoS56016.2022.10032171.
- [4] F. I. Farisi and R. M. Syamsuddin, “Offline-first architecture for Android applications using Room and ViewModel,” *International Journal of Computer Applications*, vol. 179, no. 16, pp. 1–6, Feb. 2018. doi: 10.5120/ijca2018916246.
- [5] T. Raharja, A. Ramadhan, and B. Arfianto, “Designing user-friendly mobile applications for culinary purposes,” in *2020 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, pp. 179–184, Sept. 2020. doi: 10.1109/CENIM51130.2020.9263135.
- [6] S. Widyanti, A. Suhartono, and H. Santoso, “Mobile food recipe application with voice recognition for interactive cooking,” *Procedia Computer Science*, vol. 161, pp. 110–117, 2019. doi: 10.1016/j.procs.2019.11.106.
- [7] Y. Xu and J. Zhang, “An Android-based personalized recipe recommendation system using collaborative filtering,” *Journal of Physics: Conference Series*, vol. 1992, no. 2, p. 022077, 2021. doi: 10.1088/1742-6596/1992/2/022077.
- [8] M. L. Jain and P. A. Gade, “Smart recipe application using machine learning,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 7, no. 6, pp. 2748–2754, Jun. 2019.

- [9] A. I. Vinaya, M. D. Amale, and S. Patil, "Android-based smart food recipe and nutrition app," in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 545–549, Mar. 2021. doi: 10.1109/ESCI50559.2021.9396893.
- [10] N. S. Kolhe and R. R. Deshmukh, "Android application development using Kotlin: A case study," *International Journal of Scientific Research in Science and Technology (IJSRST)*, vol. 7, no. 2, pp. 440–444, Mar. 2020.