MANIPAL UNIVERSITY
JAIPUR

Minor Project
Recipe app - Synopsis

BCA – V Sem
Subject Code: CA3170

Submitted By

Student Name-1 :- Yuvraj Saini
Student Registration-no :- 23FS10BCA00057

Student Name-2 :- Aditya Gandhi
Student Registration-no :- 23FS10BCA00009

DEPARTMENT OF COMPUTER APPLICATIONS

Title of the Project:

MealScale: Smart Recipe & Cooking Guide

Team Details:

Member Name	Registration No.	Contribution / Work Done
Yuvraj Saini	23FS10BCA00057	Project Planning, Database Design (SQLite), Ingredient Scaling Logic, and Step-by-Step Guide Implementation
Aditya Gandhi	23FS10BCA00057	UI/UX Design (Layouts & Wireframes), Recipe Data Collection, Timer Feature Implementation, and Testing

Introduction:

In today's fast-paced world, many people find cooking challenging because recipe instructions are often generic and set for a fixed serving size. Most recipe apps only provide a list of ingredients and basic steps. They don't let users adjust recipes based on the exact amount of ingredients they have available. This creates confusion about how much to cook, when to add ingredients, and how to time each step correctly.

Our app, "MealScale: Smart Recipe & Cooking Guide," solves this problem by offering a personalized recipe experience. Users can choose a recipe and enter how much of each ingredient they have. The app then automatically adjusts all the other ingredients. It also provides a step-by-step cooking guide, showing exactly when to add each ingredient and how much to use at each step. To make things easier for users, there is a built-in timer that helps schedule cooking reminders throughout the process.

This project was developed as part of our minor project, aiming to combine mobile app development, database management, and user experience design. We use SQLite to save recipes, ingredient information, and cooking procedures locally so that users can access it offline. The app is created with Android Studio (Kotlin) for both the front end and back end, featuring a simple and clean user interface.

Major Features:

- Recipe Selection: View recipes stored in the local database.
- Ingredient Scaling: Adjust ingredients based on user input.
- Step-by-Step Cooking Guide: Follow each step one by one with clear instructions.
- Timer Integration: Set a timer for boiling, frying, or any cooking stage, and receive reminders.
- Offline Functionality: No internet needed, as data is stored locally using SQLite.

This project stands out because it not only helps users learn recipes but also offers a personalized and interactive cooking experience. Its ingredient scaling, live step guidance, and built-in timer set it apart from typical recipe apps that only provide static information.

Objectives

The main goal of this project is to create an interactive and user-friendly recipe application for Android that gives tailored cooking help to users. The project aims to:

1: Develop a Mobile Recipe Application: Build an easy-to-use Android app using Kotlin with a clean and simple user interface.

2: Implement Recipe Storage and Offline Access: Use SQLite to store recipes, ingredients, and steps locally so the app can function without internet access.

3: Provide a Step-by-Step Cooking Guide: Show step-by-step instructions to help users follow the recipe in the right order and avoid errors.

4: Integrate a Cooking Timer: Allow users to set timers for boiling, frying, or other cooking tasks and send notifications when time is up.

5: Ensure Personalization and Accuracy: Create instructions that fit the user's available ingredients, guiding them on how much to add and when to add it.

6: Improve the User Experience: Offer a smooth flow between recipe selection, ingredient input, cooking steps, and timer setup.

Project Timeline:

Week 1 Planning and Requirement Gathering
- Understand the problem statement
- Finalize features (ingredient scaling, step-by-step guide, timer)
- Decide on the tech stack (Kotlin, SQLite)
- Prepare the initial project synopsis

Week 2 UI/UX Design and Wireframing
- Create wireframes for all screens (Home, Recipe Details, Ingredient Input, Steps, Timer)
- Design the color scheme and layout
- Get approval from faculty for the design

Week 3 Database Design and Setup
- Design the SQLite database (Tables: Recipes, Ingredients, Steps)
- Populate it with sample recipes and steps
- Write database helper functions

Week 4 Frontend Development – Part 1
- Implement the Home screen with RecyclerView
- Implement the Recipe Details screen
- Integrate the database fetch functionality

Week 5 Frontend Development – Part 2
- Implement Ingredient Input and Scaling logic
- Implement the Step-by-Step screen
- Build navigation between all screens

Week 6 Feature Implementation
- Add the Timer feature with CountDownTimer
- Add notifications (optional)
- Improve UI/UX (buttons, layouts, spacing)

Week 7 Testing and Debugging
- Test the app on Emulator and Real Device
- Fix UI bugs, navigation issues, and crashes
- Optimize database queries

Week 8 Documenting and Final Review
- Prepare the project report (Synopsis, DFDs, ER Diagram, Objectives, Conclusion)
- Conduct the final demo run
- Submit the project before the deadline

Tools, Platforms & Requirements

Platform: Android Studio (Kotlin), XML

Database: SQLite (local recipe storage)

Design Tools: Figma, Canva (for wireframes), Material Design Components

Testing: Android Emulator, Physical Android Device

Version Control: Git, GitHub (optional)

Hardware: Minimum: Intel i3, 4GB RAM, 10GB free storage

Recommended: 8GB RAM for better Android Studio performance

Software:

Android Studio, JDK

SQLite Browser (optional)

Gradle (build tool)

DFD Explanation

The Data Flow Diagram (DFD) shows how the MealScale: Smart Recipe & Cooking Guide application functions.

User: The main external entity interacts with the app by choosing recipes, entering ingredient amounts, and navigating through steps.

Home (1.0): Displays a list of recipes and lets users search and select one.

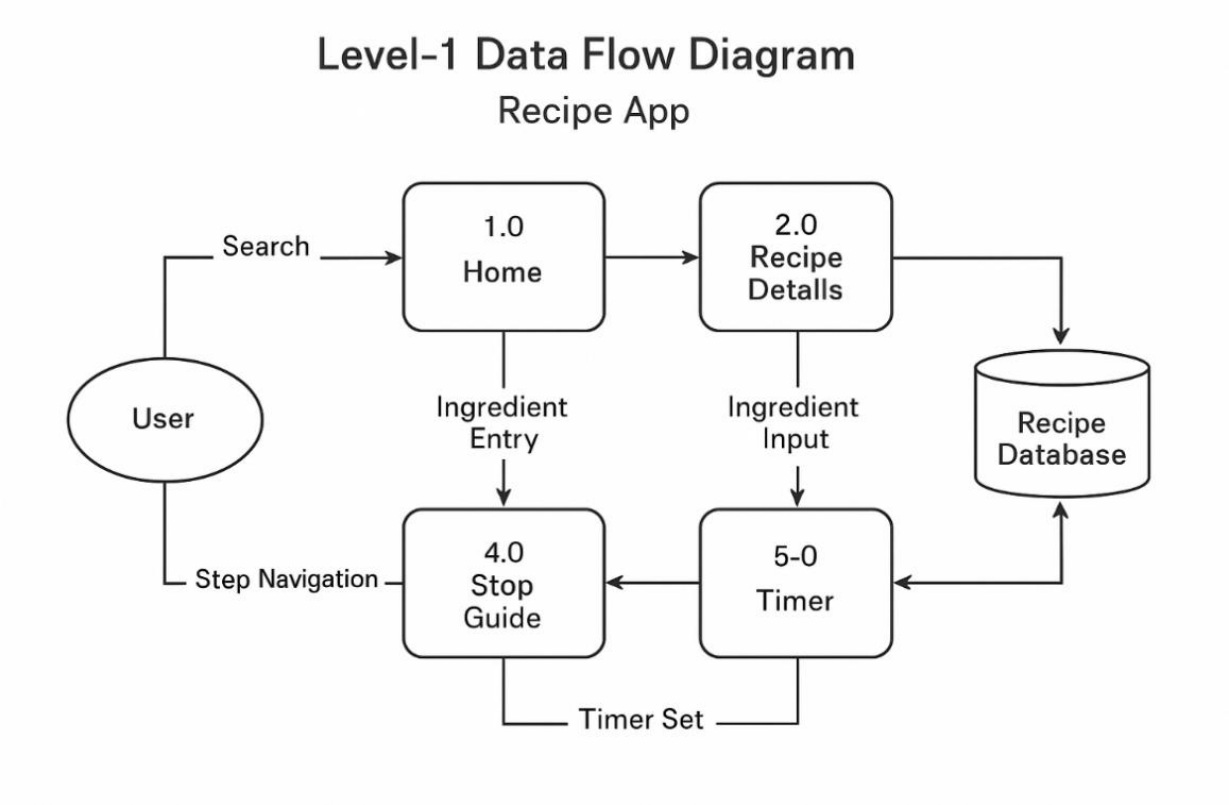
Recipe Details (2.0): Retrieves recipe data such as ingredients and steps from the Recipe Database.

Scale Ingredients (3.0): Adjusts ingredient amounts based on the user's input; this is a unique feature.

Step Guide (4.0): Offers step-by-step cooking instructions to the user.

Timer (5.0): Lets users set a timer for each cooking step and alerts them when the time is up.

Recipe Database: Stores all recipes, ingredients, and steps, and sends data back to the system when needed.



ER Diagram

Entities and Attributes

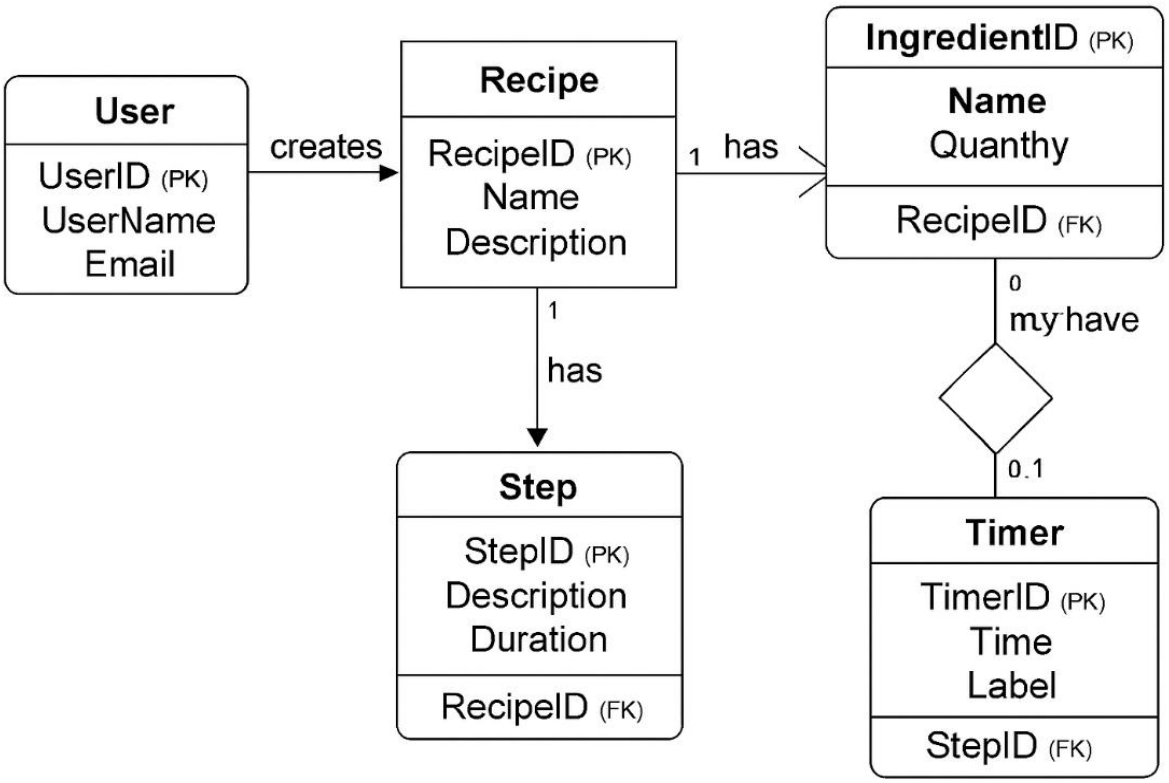
User Entity: This stores basic user details, including UserID, UserName, and Email. A user can create multiple recipes.

Recipe Entity: This holds recipe information, such as RecipeID, Name, and Description.

Ingredient Entity: This stores the ingredients needed for each recipe, including IngredientID, Name, and Quantity. It connects to recipes using RecipeID, creating a one-to-many relationship.

Step Entity: This contains detailed instructions for a recipe, with StepID, Description, and Duration. It connects using RecipeID, creating a one-to-many relationship.

Timer Entity: This links to each step using StepID as a foreign key. It stores timer details like Time and Label, which is optional for steps that require timing.



References

Android Developers Documentation, https://developer.android.com

(For learning Android Studio, Kotlin, and XML layouts)

SQLite Official Documentation, https://www.sqlite.org/docs.html

(For designing and integrating a local database for recipe storage)

Material Design Guidelines, https://m3.material.io

(For following UI/UX best practices and creating a clean design)

TutorialsPoint – Kotlin Programming Language
(Used for understanding basic Kotlin syntax and Android development)

GeeksforGeeks – SQLite in Android with Example
(Helped in implementing SQLite database integration and CRUD operations)

GitHub & Open-Source Recipe Projects
(For reference in structuring recipe data and step-by-step flows)

Stack Overflow
(For debugging issues and getting solutions during development)