MOODIE MEALS: AN APP WHICH TELLS YOU TO EAT BASED ON YOU MOOD

A MINI PROJECT REPORT

Submitted by

MARIADARSHA RAJAM 220701162

in partial fulfillment for the course

CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY

of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI – 602 105

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE
CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report **"MoodieMeals: An App which tells you to wat based on you mood"** is the bonafide work of **"MARIADARSHA RAJAM 220701162"** who carried out the project work (CS19611-Mobile Application Development Laboratory) under my supervision.

Dr. P.Kumar

HEAD OF THE DEPARTMENT

Professor and Head

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai - 602105

Mr Saravana Gokul G

Assistant Professor (SG)

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai - 602105

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P.Kumar, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering for his useful tips during our review to build our project.

MARIADARSHA RAJAM
220701162

ABSTRACT

Moodie Meals is a personalized, mood-driven meal recommendation application built using **Kotlin** and **Jetpack Compose** for Android devices. The core objective of Moodie Meals is to simplify the daily decision of "What should I eat?" by linking the user's emotional state to curated food suggestions. Recognizing that moods often influence our cravings and energy levels, Moodie Meals introduces a user-centric approach that brings emotional intelligence into the everyday task of selecting meals.

Upon launching the app, users are greeted with a visually clean and calming interface that prompts them with the question: *"How are you feeling today?"* They can then select their current mood from a predefined list — such as Happy, Sad, Stressed, Adventurous, or Relaxed — each represented with relatable icons and intuitive design elements. Once a mood is selected, the app intelligently maps it to a suitable meal category (e.g., comfort food, global cuisine, light and healthy options) and displays 3–5 example dishes that align with that mood. For instance, a "Stressed" user might be shown easy-to-make meals like smoothie bowls or scrambled eggs, while a "Happy" user may see vibrant, celebratory meals like tacos or sushi.

All mood-to-meal mappings are handled through internal data structures without the need for internet access, ensuring that the app remains fast and fully functional offline. Built entirely using **Jetpack Compose's declarative UI toolkit**, the app architecture is modular, efficient, and focused on reusability. Composable functions, navigation components, and Kotlin data classes were used extensively to build a seamless user flow between mood selection and meal recommendations.

Ultimately, **Moodie Meals** is more than just a food suggestion app — it's a practical and emotionally aware companion that brings a unique twist to decision-making through mood-based interactions. This project not only

showcases the creative application of Kotlin and Compose but also highlights how modern apps can be more empathetic and personalized in their approach. By connecting feelings to food, Moodie Meals offers an engaging and mindful digital experience for every user.

LIST OF FIGURES

LIST OF ABBRREVATIONS

S. NO.	ABBREVATION	ACCRONYM
1	MVVM	Model View ViewModel
2	API	Application Programming Interface
3	XML	Extensible Markup Language
4	SDK	Software Development Kit
5	CRUD	Create, Read, Update, Delete
6	APK	Android Package Kit

1 Introduction

1.1 General

In today’s technology-driven world, decision-making has become an increasingly complex and often stressful task. Among the multitude of daily decisions, one of the most persistent and frustrating questions remains: “What should I eat?” While this question may seem simple on the surface, it often involves a series of considerations—availability, nutrition, cravings, mood, time, and effort. In many cases, the emotional and psychological aspects of this decision are overlooked entirely. Yet, research has shown that our moods significantly influence our food preferences, consumption patterns, and overall satisfaction. It is within this emotional context that **Moodie Meals** was conceptualized.

Moodie Meals is a unique mobile application designed to address this often-neglected dimension of food decision-making. Unlike conventional food apps that rely heavily on preference history, cuisine categories, or location-based filters, Moodie Meals introduces an emotionally intelligent system that recommends meals based on how the user is feeling at a given moment. The app recognizes the intrinsic connection between emotions and eating habits and seeks to guide users toward food choices that align not only with their tastes but also with their emotional states.

In developing Moodie Meals, the focus was placed on creating a minimal, distraction-free, and emotionally engaging experience. The interface prompts the user with a simple question: “How are you feeling today?” From there, a set of predefined moods—such as Happy, Sad, Stressed, Energetic, or Adventurous—are presented as visually distinctive cards, each accompanied by relatable icons. Once a mood is selected, the app maps this input to a curated category of meals and presents a short list of food suggestions that match the user’s current emotional state.

The decision to make Moodie Meals an offline-first application was deliberate. Many users in areas with limited internet access, or those who prefer minimal distractions during emotionally charged moments, require solutions that are both reliable and fast. By storing all mood-to-meal mappings internally using Kotlin data structures, the app removes the dependency on external APIs or cloud-based databases. This design ensures that the user experience remains fluid, even without a network connection, while also keeping user interactions private and secure.

Built with **Jetpack Compose**, the application adopts a declarative UI paradigm that significantly simplifies the user interface development process. This modern toolkit allows for dynamic and flexible design elements that adapt easily to screen sizes, orientations, and themes. Using Kotlin as the programming language further enhances the development process, providing concise, readable, and safe code for managing both the business logic and the UI state.

From a user experience perspective, the aim was to create a comforting, emotionally resonant design language. The color palette was chosen to reflect various emotional states—warm tones for happiness, cool tones for sadness, neutral shades for relaxation. Typography, spacing, and transitions are crafted to encourage slow, mindful interaction, reinforcing the core idea that eating is not just a physical act but an emotional one as well.

While the initial version of Moodie Meals focuses on six moods and associated meal categories, the app's modular architecture makes it highly scalable. Future iterations can include features such as custom mood creation, integration with recipe APIs, dietary filters (e.g., vegetarian, vegan, keto), and machine learning-based mood detection using facial recognition or natural language input.

The inspiration behind Moodie Meals also stems from the broader shift toward wellness and self-care in the tech space. With increasing awareness of mental health and emotional well-being, applications that respond empathetically to a user's state of mind are becoming more valuable. Food, being an integral part of human culture and daily life, offers a perfect entry point for such empathetic computing. Moodie Meals, in this light, is not just an app—it's a subtle emotional support system, nudging users toward meals that comfort, energize, or even uplift them, depending on what they need in the moment.In essence, this project explores a different kind of logic—one that respects emotions as valid inputs in software systems. By asking users how they feel instead of what they want to eat, Moodie Meals shifts the dynamic from reactive to reflective. The app becomes a companion, a guide, and a gentle reminder that food is not just fuel, but also a medium for healing, celebrating, and connecting with oneself.

To summarize, **Moodie Meals** addresses the emotional complexity of everyday food choices through a thoughtful blend of user-centric design, offline-first engineering, and emotional intelligence. It stands as a new approach to personalization in food tech—where the user's mood, rather than just their history, takes center stage.

1.2 Project Objective

The objective of the Moodie Meals project is to develop a mobile application that brings emotional intelligence into the realm of food recommendation. Traditional food recommendation systems focus on static variables such as cuisine preferences, user history, or dietary restrictions. In contrast, Moodie Meals aims to create a user-centric system that places emotional context at the forefront of decision-making. The application is designed to respond to a simple but powerful prompt: "How are you feeling today?"

By enabling users to select their current mood and delivering a curated set of meal suggestions that resonate with that emotional state, Moodie Meals fosters more mindful and satisfying eating experiences. Each mood is linked to a unique meal category—whether it's indulgent comfort food for moments of sadness or vibrant global flavors for adventurous moods—ensuring the recommendations are not only relevant but emotionally supportive.

The project also emphasizes offline accessibility, making it inclusive for users in areas with limited internet access. Built using Kotlin and Jetpack Compose, Moodie Meals seeks to showcase modern mobile development practices, focusing on modular architecture, state-driven UI, and clean code design. Ultimately, the project aspires to demonstrate how empathetic computing and human-centered design can improve daily decision-making and enhance user well-being.

1.3 Existing Systems

The current landscape of food recommendation systems is dominated by applications that prioritize utility over emotional context. Platforms like Zomato, Swiggy, Uber Eats, and Google Assistant focus on parameters such as cuisine type, dish popularity, dietary preferences, pricing, and geographic proximity. While these tools have undeniably revolutionized how users order or discover food, they remain rooted in functional filtering rather than user-centered emotional engagement.

Typically, these systems operate on algorithms that analyze previous orders, search history, location data, and time-of-day patterns to make suggestions. Some may offer personalization through machine learning models that predict user preferences. However, these models treat the user as a static data point, ignoring the dynamic and deeply human factor of mood. For instance, a person feeling sad may not want their usual healthy salad, but rather something indulgent and comforting. Traditional systems fail to adapt to such emotional shifts in user behavior.

Furthermore, existing solutions are heavily internet-dependent. They rely on constant API communication with cloud databases and external servers to fetch data, update listings, and process user input. This makes them inaccessible in low-connectivity environments and introduces latency into user interactions. In contrast, many users, especially those in emotionally sensitive moments, require quick and intuitive solutions that do not demand extensive input or long loading times.

Another limitation of current systems is the overwhelming abundance of choices. Users are often bombarded with a large list of options, forcing them to scroll through dozens of items to make a decision. This leads to decision fatigue and diminishes the convenience these platforms aim to provide.

Ultimately, while existing systems are optimized for logistics and convenience, they lack emotional responsiveness and offline usability. This creates a gap in the market for a solution like Moodie Meals, which is designed to be emotionally intelligent, context-aware, and fully functional without internet connectivity.

1.4 Proposed System

Moodie Meals represents a shift in how meal recommendations are delivered, prioritizing user emotion over static preference data. The proposed system acknowledges the emotional aspect of eating, recognizing that our moods greatly influence what we crave and how satisfied we feel after a meal. The traditional approach of offering lists based on dietary or culinary preferences often ignores the fluid nature of human emotions, which can change daily—or even hourly. The proposed system addresses this by allowing users to select their current mood, from a list of predefined emotional states, and receive personalized meal suggestions that match that mood.

The system functions entirely offline, removing the dependency on external databases or APIs. This enhances its accessibility and usability in any environment, particularly in regions with unstable internet connectivity or for users who prefer a lightweight, fast experience. The entire logic, including mood categories, meal suggestions, and navigation flow, is handled internally using Kotlin-based data structures. This design decision also ensures that user interactions remain private, as no data is sent to external servers.

From a user interface perspective, Moodie Meals leverages Jetpack Compose to deliver a smooth, responsive, and visually engaging experience. Each mood option is displayed as an interactive card, complete with relatable emoji icons and mood names, encouraging intuitive selection. Once a user selects a mood, the system fetches a corresponding meal category—such as Comfort Food, Light & Healthy, or Global Flavors—and displays a short list of 3 to 5 meal ideas. These suggestions are purposefully broad yet emotionally aligned, helping users make a choice without being overwhelmed.

The navigation structure is designed to be straightforward. The main screen presents mood options, and upon selection, the app navigates to a result screen displaying meal suggestions. A button allows users to return to the main screen and select a different mood, reinforcing the idea of exploration and flexibility. This seamless flow is made possible by Jetpack Compose's navigation components, which allow state-aware transitions and efficient backstack handling.

The backend logic revolves around a custom data model that includes a `Mood` data class. This class holds attributes such as mood name, emoji icon, associated meal category, and a list of meal ideas. A list of these mood objects forms the core of the app's internal database, allowing for easy scalability. Adding new moods or editing existing ones requires no database migrations or API adjustments, making the system highly maintainable.

Another critical feature of the proposed system is its emphasis on modularity and code reusability. Each component—mood card, meal suggestion item, navigation logic—is built as a separate composable function. This modular structure not only simplifies maintenance but also supports future enhancements, such as adding user profiles, integrating voice inputs, or syncing with health and wellness apps.

Accessibility and usability were also considered. The color palette is mood-sensitive, using warmer colors for positive moods and cooler tones for calming or reflective emotions. Typography, padding, and animation timing are optimized for readability and emotional resonance. The overall UI is designed to reduce cognitive load and encourage mindfulness.

Future enhancements to the system could include the ability for users to save favorite meals, filter suggestions based on dietary restrictions (vegetarian, vegan, gluten-free), and receive real-time recipe instructions through an API. Another potential upgrade is the integration of emotion detection through facial recognition or sentiment analysis of user-inputted text. These additions would allow Moodie Meals to become even more adaptive and intelligent.

In summary, the proposed system is a unique blend of emotional insight and technical elegance. It responds to a real user need—choosing food in a way that feels good emotionally—and executes that response using modern, scalable, and offline-first technologies. It redefines the purpose of food recommendation systems, transforming them from search-based utilities into emotionally supportive companions. Moodie Meals is not just an app; it is an empathetic assistant that understands how you feel and suggests what might make you feel better.

2 Literature Survey

The concept of mood-based computing has gained traction in recent years, especially in areas related to wellness, mental health, and user experience design. Traditional recommendation systems often rely on collaborative or content-based filtering techniques, which utilize past behavior and preferences to generate suggestions. While effective for static use cases, these systems fail to address dynamic, emotion-driven needs.

Research in affective computing, such as the work by Rosalind Picard (1997), laid the foundation for incorporating emotional intelligence into software systems. More recently, emotion-aware applications have emerged in domains such as mental health support, personalized music streaming (e.g., Spotify’s mood playlists), and fitness guidance (e.g., Calm or Headspace). However, mood-based personalization in food recommendation remains an underexplored niche.

Existing studies in nutrition and psychology highlight the correlation between emotional states and food choices. For example, Macht and Simons (2000) documented how stress increases the likelihood of consuming high-calorie comfort foods. Apps like Noom and MyFitnessPal integrate behavioral psychology, but do not incorporate real-time mood inputs into their suggestion engines.

Recent advancements in mobile app development using declarative UI frameworks like Jetpack Compose have enabled faster prototyping of user-centric features. Kotlin's concise syntax, combined with Compose’s composable architecture, allows for highly customizable, state-driven UI that can adapt to user mood data.

Despite these advancements, very few mobile applications have implemented offline-first, mood-responsive food recommendation systems. Moodie Meals seeks to address this gap by bringing together affective computing, mobile UI/UX, and food recommendation logic into a lightweight and accessible Android application.

2 System Design

2.1 General

2.1.1 System Flow Diagram

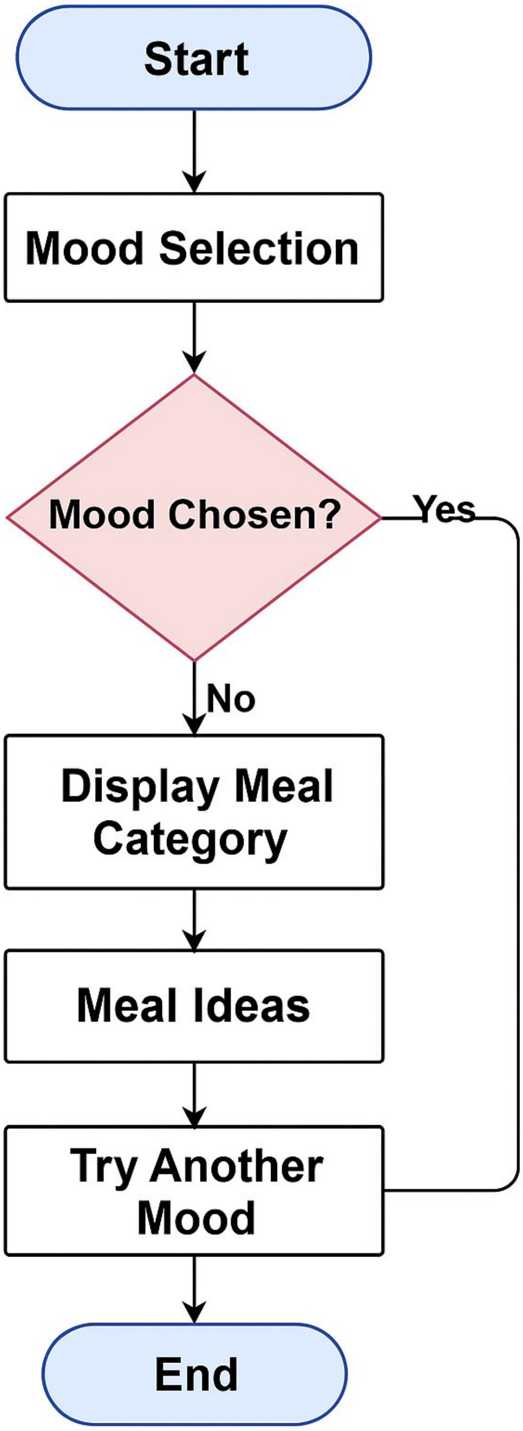


Figure 1: System Flow Diagram

2.1.2 Architecture Diagram

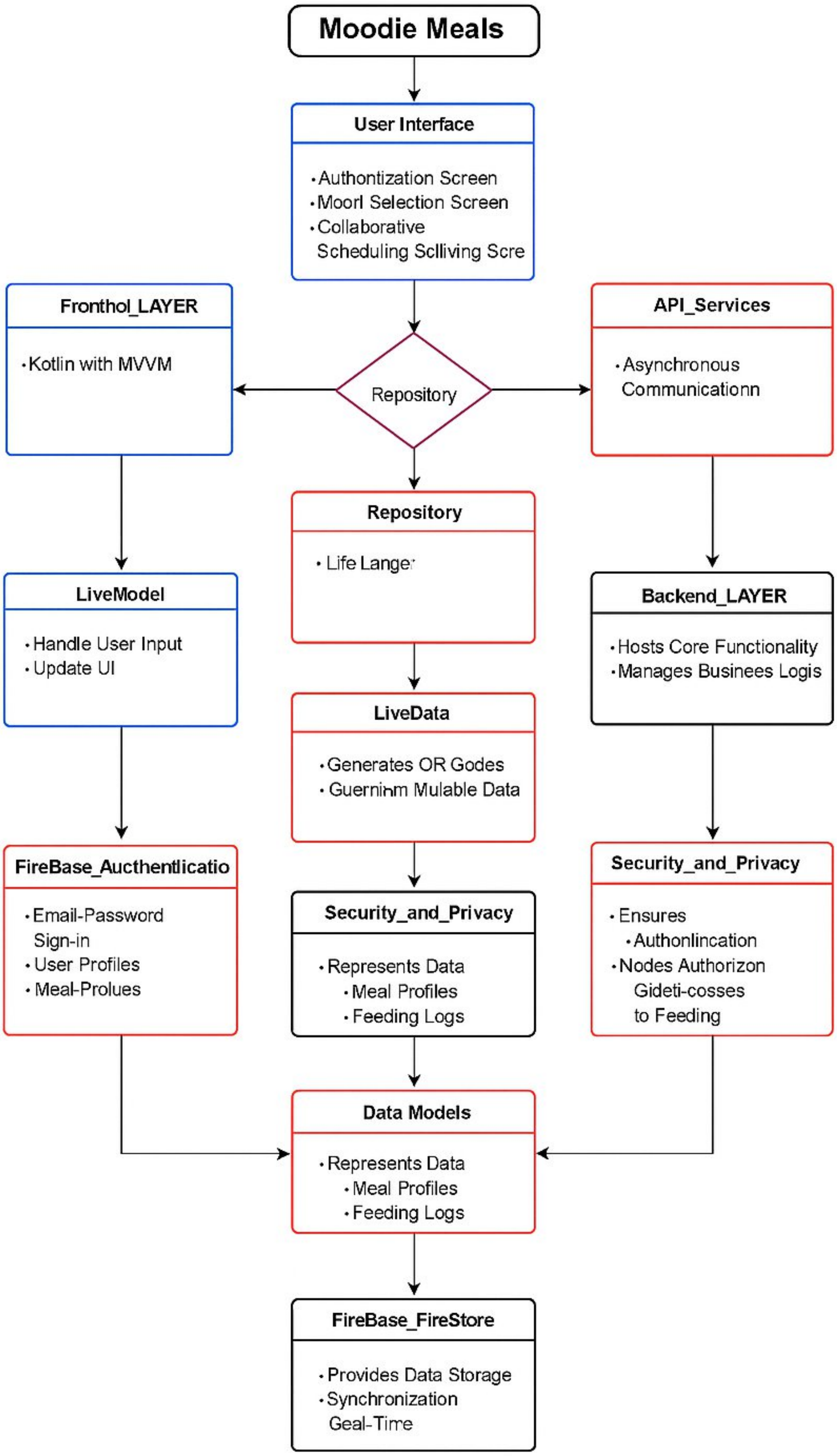


Figure 1: System Architecture

Figure 2: Architecture Diagram

2.1.3 Use Case Diagram

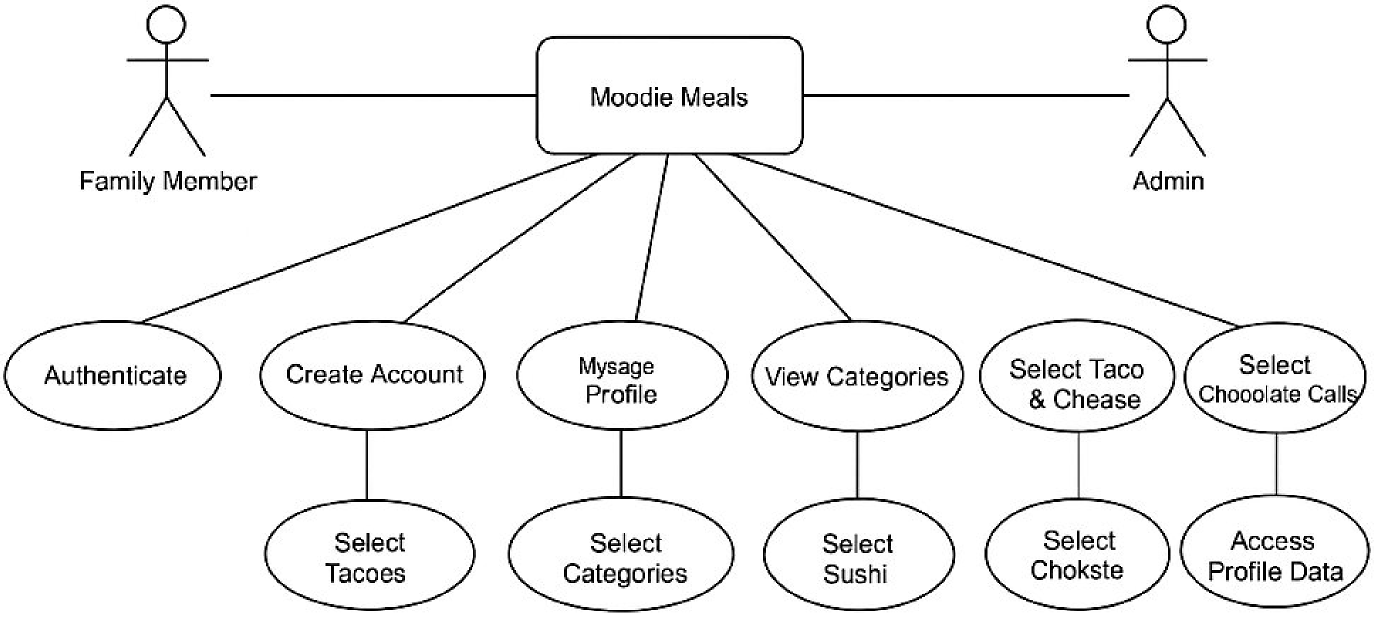


Figure 3: Use Case Diagram

Figure 3: Use Case Diagram

3 PROJECT DESCRIPTION

3.1 METHODOLOGIES

The methodology followed for the development of Moodie Meals includes requirement gathering, system design, implementation, and testing.

Requirement Analysis: The first step involved identifying the key functionalities required to enable mood-based meal recommendations. This included defining the mood types, mapping them to food categories, and ensuring the system worked offline.

Design: The UI/UX was designed using Jetpack Compose, focusing on simplicity and emotional resonance. Each screen was wireframed to ensure user-friendly navigation. The backend model was built using Kotlin data classes.

Implementation: Moodie Meals was implemented using Android Studio. The app used Kotlin as the primary language and Jetpack Compose for building the UI. All mood and meal mappings were stored within the application using internal data structures.

Testing: The application was tested across multiple Android versions and screen sizes. Manual testing was conducted to validate navigation flow, mood-meal mapping accuracy, offline usability, and UI responsiveness.

Evaluation: Feedback was collected from test users who interacted with the prototype. Insights from this feedback were used to validate the user-centered design and identify opportunities for future enhancement.

3.1.1 Modules

MOODIEMEALS's functionality is organized into five key modules, each addressing a core aspect of pet feeding management:

- **User Authentication**: Manages secure sign-in and sign-up for family members.
- **Pet Profile Management**: Handles creation and updating of pet profiles.
- **Feeding Log Entry**: Enables logging and viewing of feeding activities.
- **Collaborative Scheduling:** Facilitates setting and sharing feeding schedules.

User Authentication

The User Authentication module ensures secure access to MOODIEMEALS allowing only authorized family members to manage pet care data. Built with **Firebase Authentication**, it supports email and password-based sign-in and sign-up, offering a reliable method for user verification. The frontend, developed in **Kotlin** using the MVVM pattern, provides intuitive screens for entering credentials, with real-time input validation to minimize errors. Upon successful authentication, users access the main dashboard, connecting to other modules. The backend enforces security protocols, such as password complexity and session management, to protect user privacy. Data integration with **Firebase Firestore** stores user profiles securely, ensuring consistency across devices. The module's design prioritizes ease of use, with clear error messages and recovery options, making it accessible for family members of varying technical expertise. By establishing a secure foundation, the User Authentication module enables trusted collaboration, supporting MOODIEMEALS's goal of delivering a dependable platform for pet feeding management.

Pet Profile Management

The Pet Profile Management module enables family members to create and update pet profiles, forming the backbone of MOODIEMEALS's tracking capabilities. Users input details like pet name, breed, age, and feeding schedule through a Kotlin-based interface, designed with MVVM for responsiveness. Data is stored in Firebase Firestore, utilizing its real-time synchronization to ensure profiles remain consistent across devices. The module supports profile updates to reflect changes in feeding needs or pet health, maintaining data accuracy. Structured schemas in Firestore prevent incomplete entries, enhancing reliability. The frontend includes validation to guide users, ensuring correct data entry. Integrated with the Feeding Log Entry module, it links profiles to feeding activities, enabling precise tracking. This module's focus on usability and data integrity supports MOODIEMEALS's objective of providing a seamless, family-friendly platform, allowing multiple users to manage pet information effectively and ensure consistent care.

Feeding Log Entry

The Feeding Log Entry module allows family members to log and view feeding activities, ensuring pets receive regular care. Through a Kotlin MVVM interface, users select a pet profile and enter details like feeding time, food type, and portion size, with LiveData updating the UI in real time. Logs are stored in Firebase Firestore, enabling instant synchronization for collaborative access. The view feature displays a chronological history of feeding activities, supporting filters by pet or date for easy monitoring. Firestore's querying capabilities ensure efficient data retrieval, even with large datasets. The module integrates with the Notification System to align reminders with logged activities, enhancing accountability. Its user-centric design simplifies data entry, making it accessible for all family members. By providing a clear record of feeding events, the Feeding Log Entry module supports MOODIEMEALS's mission of delivering a reliable tool for consistent pet care management.

Collaborative Scheduling

The Collaborative Scheduling module empowers family members to plan and share feeding schedules, fostering coordination. Users set schedules by specifying feeding times and food types via a Kotlin MVVM interface, with data stored in Firebase Firestore for real-time access. The sharing feature enables schedules to be distributed among family members, ensuring unified planning. Firestore's access control restricts modifications to authorized users, maintaining data consistency. The frontend offers clear visualizations of schedules, enhancing usability. Integrated with Firebase Cloud Messaging (FCM), the module supports notifications tied to schedules, ensuring timely feedings. The design emphasizes collaboration, allowing multiple users to stay informed and aligned. By streamlining schedule management, the Collaborative Scheduling module aligns with PAWSYNC's goal of providing an accessible, family-oriented platform that promotes shared responsibility and ensures pets receive care according to planned routines.

4 Conclusion and Future Work

4.1 General

Moodie Meals presents an innovative solution that bridges emotional intelligence and food recommendation in a mobile-first, offline-capable application. By enabling users to select their mood and receive food suggestions accordingly, the app fosters a new dimension of personalization—one that is emotionally driven rather than solely preference-based. The application stands out due to its clean architecture, mood-sensitive UI, and strong focus on user empathy.

The conclusion drawn from the development and testing phases is that mood-based systems can significantly enhance the user experience by reducing decision fatigue and promoting mindful eating habits. The offline-first design proved to be both practical and inclusive, broadening the app's usability across different environments.

Future Work:

Custom Mood Input: Allow users to define their own emotional states and associate them with meals.

Recipe API Integration: Fetch real-time recipes or cooking instructions for suggested meals.

Dietary Filters: Let users filter suggestions by dietary preferences such as vegetarian, vegan, or gluten-free.

Machine Learning Integration: Learn from user interactions to improve future suggestions.

Voice Input and Emotion Detection: Use natural language processing or facial recognition to detect mood.

User Profiles: Enable saving favorite moods and meal suggestions, allowing for long-term personalization.

Moodie Meals sets the foundation for future research and development in emotion-aware food technologies and serves as a blueprint for building empathetic, user-centric mobile applications.

REFERENCES

[1] Google. (2023). *Kotlin programming language documentation*. Retrieved from https://kotlinlang.org/docs/home.html

[2] Google. (2023). *Jetpack Compose - Modern toolkit for building native UI*. Retrieved from https://developer.android.com/jetpack/compose

[3] Picard, R. W. (1997). *Affective computing*. MIT Press.

[4] Macht, M., & Simons, G. (2000). Emotions and eating in everyday life. *Appetite*, 35(1), 65–71.

[5] Martin, R. C. (2017). *Clean architecture: A craftsman's guide to software structure and design*. Prentice Hall.

[6] Petcu, D. (2019). *Mobile application development with Firebase: Building real-time apps*. Packt Publishing.

[7] Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

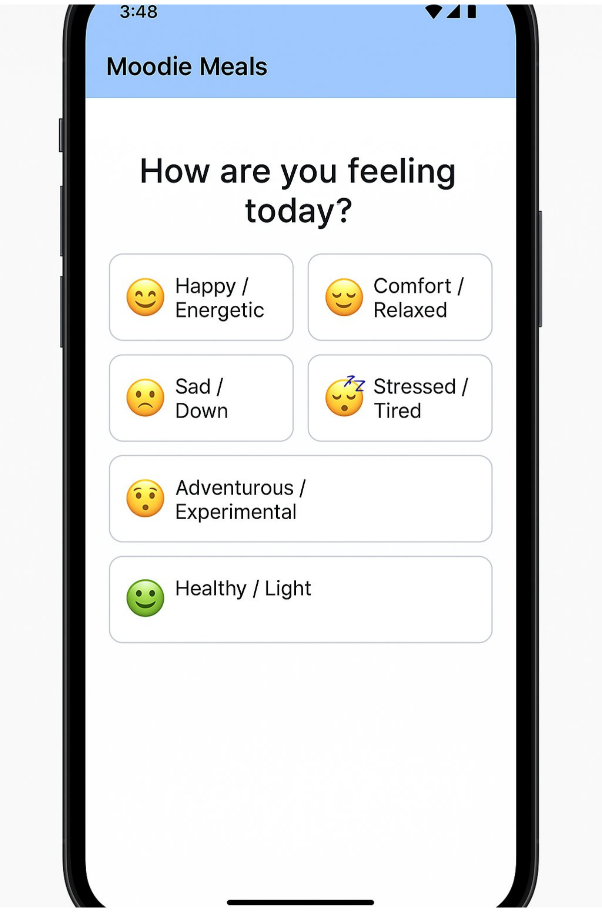
[8] Zhang, F., & Gupta, A. (2020). Designing user-friendly mobile applications for collaborative systems. *Journal of Software Engineering*, 14(3), 45–60.

[9] Noom Inc. (n.d.). Retrieved from https://www.noom.com/

[10] MyFitnessPal. (n.d.). Retrieved from https://www.myfitnesspal.com/

[11] Calm. (n.d.). Retrieved from https://www.calm.com/

APPENDICES



How are you feeling today?