INFO 6350 – Smartphones-based Web Development Final Project

Global Kitchen – Recipe Management Application

Nagarjun Mallesh
NUID – 002788601

Table of Contents

Introduction

As a passionate food enthusiast and an international student with a love for cooking, I have often found myself struggling to keep track of my favourite recipes. Whether it's a cherished family recipe shared by my mother or an exciting new dish discovered online, the process of organizing and managing these culinary gems has always been a challenge.

Problem Statement: Like many others in my position, I have experienced the frustration of sifting through scattered notes, bookmarks, and physical cookbooks, trying to find that one special recipe I wanted to recreate. There are multiple source of recipes on the internet, but it is hard to find the perfect recipe with good description and instruction for the user to follow. This is a tedium process in cooking.

What Motivated Me: This personal struggle inspired me to develop the Recipe Management System—an iOS application designed to revolutionize the way food lovers like myself manage their recipe collections. While I was working on the project and while using the GitHub, I got inspiration to create an application that would be a repository of the recipes. Born out of a genuine need for a more efficient and user-friendly solution, this application aims to provide a centralized hub where users can store, categorize, search, and manage their recipes with ease. One thing, I observed is I always found myself lost in the options over the web while finding the right recipes and going through tons of bookmarks, Capsizing the list into an application would make the life extremely simple and convenient.

Objective: Global Kitchen - The Recipe Management System is not just another recipe app; it is a thoughtfully crafted tool that understands the unique challenges faced by busy individuals who are passionate about cooking. This application seeks to bridge that gap by offering a seamless and intuitive platform that simplifies recipe management, making it easier for users to access and utilize their favourite recipes whenever the craving strikes. This application is just like a GitHub repository, wherein instead of adding new coding projects we add the projects taken up in the kitchen. This could be the multi-user platform which could be used by multiple user across the globe, accessing the tons of recipes from different part of the world. One pit stop to all cuisines in the world.

Design /Approach

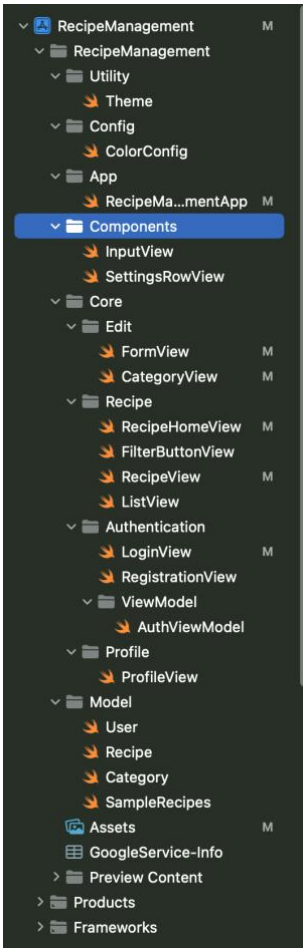
The global Kitchen uses a feature-based modular architecture in the development of the Recipe Management System. Let's explore the application's architecture aligns with feature-based modular approach:

1. **Feature-Based Modular Architecture:**
    - In a feature-based modular architecture, the application is organized into modules based on specific features or functional areas.
    - Each module encapsulates the related components, views, models, and services required for a particular feature.
    - This approach promotes a clear separation of concerns and allows for focused development and maintenance of individual features.
2. **Core Module:**
    - Contains the core functionality and foundation of the application.
    - Includes the Authentication module, which handles user authentication using Firebase.
    - Houses the `AuthViewModel`, responsible for managing user authentication state and interactions with Firebase Authentication.
    - Encapsulates the core services, utilities, and managers that are shared across the application.
3. **Recipe Module:**
    - Encapsulates all the components, views, and models related to the recipe feature.
    - Contains the major pages of the application, such as the recipe list, recipe detail, and recipe form views.
    - Includes the necessary models, such as `Recipe` and `Category`, to represent the data structure of recipes and categories.
    - Interacts with the core module for authentication and shared functionality.

The following are the benefits for implementing the Feature-based Approach:

- **Encapsulation:** Each feature module encapsulates the related components, views, and models, making the codebase more organized and focused.
- **Separation of Concerns:** The separation of features into distinct modules allows for a clear separation of responsibilities and promotes maintainability.
- **Reusability:** Common functionality and services are placed in the core module, allowing for reuse across different feature modules.
- **Scalability:** New features can be added as separate modules without significantly impacting the existing codebase, enabling easier scalability and extensibility.
- **Parallel Development:** Teams can work on different feature modules independently, enabling parallel development and faster iteration.

The feature-based modular architecture aligns well with the principles of SwiftUI and modern iOS app development practices. It promotes a clean, modular, and maintainable codebase, making it easier to develop, test, and evolve the application over time.



Screenshot depicts the feature-based modular architecture implemented

```swift
import Foundation
import FirebaseAuth

class AuthViewModel: ObservableObject {
    @Published var user: User?

    init() {
        if let firebaseUser = Auth.auth().currentUser {
            self.user = User(id: firebaseUser.uid, fullName: firebaseUser.displayName ?? "", email: firebaseUser.email ??
                "")
        }
    }

    func signIn(withEmail email: String, password: String) async throws {
        do {
            let authResult = try await Auth.auth().signIn(withEmail: email, password: password)
            let firebaseUser = authResult.user
            self.user = User(id: firebaseUser.uid, fullName: firebaseUser.displayName ?? "", email: firebaseUser.email ??
                "")
        } catch {
            print("Error: \(error.localizedDescription)")
            throw error
        }
    }

    func createUser(withEmail email: String, password: String, fullname: String) async throws {
        do {
            let authResult = try await Auth.auth().createUser(withEmail: email, password: password)
            let firebaseUser = authResult.user
            let changeRequest = firebaseUser.createProfileChangeRequest()
            changeRequest.displayName = fullname
            try await changeRequest.commitChanges()
            self.user = User(id: firebaseUser.uid, fullName: fullname, email: email)
        } catch {
            print("Error: \(error.localizedDescription)")
            throw error
        }
    }
```

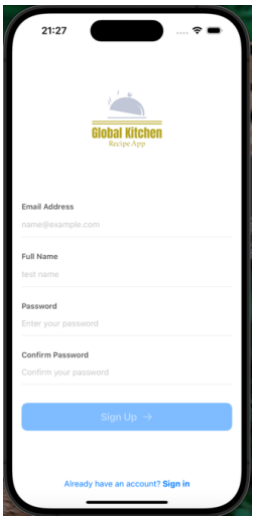
Screenshot 2a - depicts the AuthViewModel class which handles user authentication

```
    func signOut() throws {
        do {
            try Auth.auth().signOut()
            self.user = nil
        } catch {
            print("Error: \(error.localizedDescription)")
            throw error
        }
    }
}
```

Screenshot 2b - depicts the AuthViewModel class which handles user signout methods

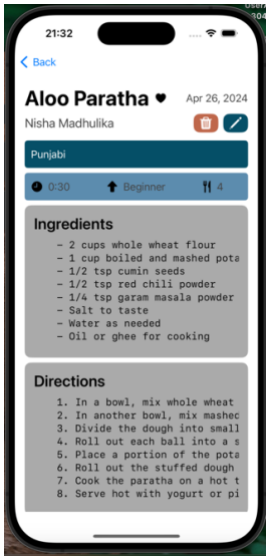
Screenshots – Application User Interface

Login Page

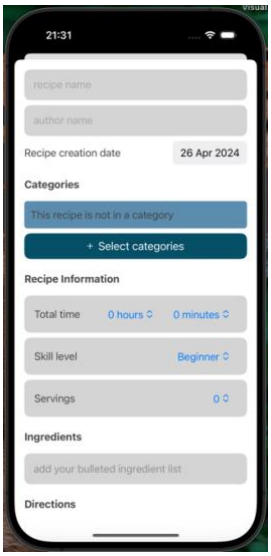


Register Page

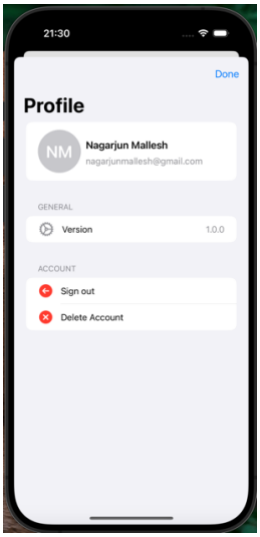
Recipe Home Page



Recipe View Page



Recipe Add Page



User Home Page

Techniques

User Authentication and Data Storage:

To ensure a secure and personalized experience, I have integrated Firebase Authentication, allowing users to create accounts and safeguard their precious recipe collections. For storing the application data, such as recipes and categories, I have leveraged Swift Data, a powerful data storage solution. This approach ensures that users' recipes, categorizations, and preferences are persistently stored and can be seamlessly accessed across devices.

Visually Stunning User Interface:

Recognizing the importance of a delightful user experience, I have meticulously crafted the application's user interface using SwiftUI, Apple's cutting-edge framework for building modern and responsive UIs. With SwiftUI's declarative syntax and powerful features, I have created a visually stunning and intuitive interface that feels natural and seamless to navigate.

Markdown Support for Recipe Content:

To enhance readability and organization, I have implemented markdown support for recipe ingredients, directions, and notes. This innovative feature allows users to format their recipes using rich text editing capabilities, such as bold headings, italicized emphasis, and bullet points, making their culinary creations more visually appealing and easier to follow.

Advanced Search Functionality:

Recognizing the frustration of sifting through cluttered recipe collections, you implemented a robust search functionality that enables users to quickly locate specific recipes based on keywords, ingredients, or categories. The intelligent search algorithm ensures accurate and relevant results, saving users valuable time and effort.

Asynchronous Data Fetching:

To ensure a smooth and responsive user experience, you implemented asynchronous data fetching techniques. This approach ensures that the application remains responsive and does not freeze or lag while fetching data from the backend or Swift Data storage, providing a seamless and enjoyable experience for users.

Conclusion and Future Work

The application's comprehensive features, including recipe creation, editing, deletion, and categorization, empower users to curate their personal recipe library according to their preferences. The incorporation of favourites and markdown support further enhances the user experience, allowing for personalized access to cherished recipes and improved readability.

Future Work

Despite the impressive achievements, the Recipe Management System has immense potential for future improvements and enhancements. The introduction of user profiles would unlock a realm of personalization and sharing features, fostering a vibrant community of food enthusiasts.

Integrating social sharing capabilities would enable users to seamlessly share their favourite recipes and culinary experiences across popular social media platforms. This feature would not only inspire others but also facilitate the discovery of new and exciting recipes from fellow food lovers around the world.

To further elevate the application's utility and convenience, implementing meal planning functionality would be a game-changer. Users could generate personalized weekly meal plans based on their saved recipes, ensuring a well-balanced and diverse menu. The automatic generation of shopping lists derived from the meal plan would streamline grocery shopping, saving time and reducing food waste.

Enhancing the visual appeal and engagement of the application, image support for recipes would allow users to showcase their culinary masterpieces and draw inspiration from visually stunning dishes. High-quality images accompanying each recipe would not only entice users but also provide a more immersive and enjoyable cooking experience.

Moreover, the incorporation of recipe scaling functionality would greatly benefit users by enabling them to easily adapt recipes to different serving sizes. Whether cooking for a small family gathering or a large party, users could effortlessly adjust ingredient quantities and cooking times, ensuring perfectly proportioned meals every time.

As the Recipe Management System continues to evolve and incorporate these future improvements, it has the potential to become an indispensable all-in-one platform for recipe management, meal planning, and social sharing. By constantly iterating and gathering user feedback, the application can align itself with the ever-changing needs and preferences of food enthusiasts worldwide.

Conclusion

In conclusion, the Recipe Management System represents a significant leap forward in the realm of culinary organization and management. With its solid foundation, user-centric design, and powerful features, the application empowers users to explore their culinary

passions, save time, and create delightful meals with ease. The future improvements, including user profiles, social sharing, meal planning, image support, and recipe scaling, will further enhance the application's capabilities and user experience. As the Recipe Management System continues to grow and evolve, it has the potential to revolutionize the way individuals interact with their recipes, plan their meals, and share their love for food with others. It is an exciting journey ahead, filled with possibilities to make cooking an even more enjoyable, convenient, and rewarding experience for all.