

THE FOOD MEMO RECIPE APP:

Your Culinary Companion

By Team FlavorLab

Tasneem Banu and Syed Sahir Hussain

College of Engineering, Design, and Computing

University of Colorado Denver

Denver, Colorado, USA

tasneem.banu@ucdenver.edu

sahirhussain.syed@ucdenver.edu

Index

1. Project Description

1.1 Introduction

1.2 Key Features

2. Requirements Specifications

3. Technical Description

1. Project Description

1.1. Introduction

The Food Memo Recipe App is a remarkable milestone in the culinary world, born from the success of The Food Memo Instagram page, which has been delighting food enthusiasts for over five years. One of our team members developed this page with the aim to provide users the flexibility of exploring a variety of recipes to satisfy their own food cravings. Building on the tremendous support and love received on the Instagram platform, we are excited to take The Food Memo to new heights by introducing a feature-rich mobile application.

From humble beginnings to becoming a renowned food destination on Instagram, The Food Memo has captivated audiences with an enticing array of recipes, food inspiration, and culinary tips. Now, with the development of The Food Memo Recipe App, we aim to provide our dedicated followers and new users alike with an even more convenient, personalized, and immersive culinary experience.

1.2.Key Features

- a. *Vast Recipe Collection:* The app boasts a vast collection of mouthwatering recipes, ranging from delectable vegetarian and non-vegetarian dishes to refreshing drinks and indulgent desserts. Users can browse through an extensive array of recipes, each thoughtfully categorized for easy navigation.
- b. *Intuitive Categorization:* Our app's intuitive categorization system allows users to effortlessly find recipes that suit their preferences. Categories include Main Course Breakfast , Desserts and Drink ,with plans to introduce Vegan and Keto categories in future updates.
- c. *Personalized User Experience:* The Food Memo Recipe App offers a personalized user experience through a user-friendly login page. Once logged in, users can save their favorite recipes, create custom recipe collections.
- d. *Splash Screen with Loading Bar:* Upon launching the app, users will be greeted with a visually appealing splash screen featuring The Food Memo logo and a loading bar. This ensures a smooth and engaging start to the app experience.
- e. *Efficient Firebase Database:* With Firebase integration, we are harnessing the full potential of cloud-based technology to enhance the app's performance and user experience. Firebase provides real-time data synchronization, scalability, and seamless backend support, ensuring a smooth and efficient operation of the app.
- f. *Easy-to-Use Interface:* The app features an elegant and easy-to-navigate interface, allowing users of all skill levels to access recipes, read instructions, and view ingredient lists effortlessly.
- g. *Regular Updates and New Content:* Our development team is committed to providing regular updates to The Food Memo Recipe App. These updates will include the addition of new recipes, enhanced search features, and improvements based on user feedback.

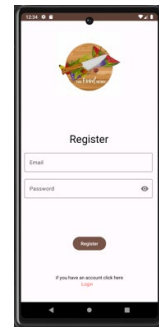
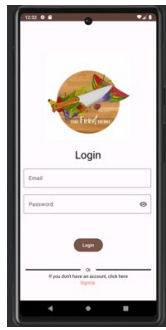
2. Requirements Specifications

- a. *Splash Screen Activity:* Upon launching the app for the first time, users will be greeted with a visually appealing splash screen featuring The Food Memo's logo, app name, and version number. The splash screen will provide a seamless and engaging introduction to the app.

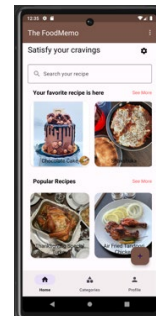
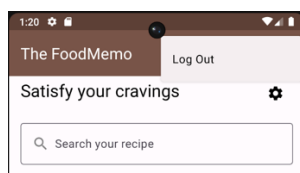
- b. *Login and Registration Activities*: The app will have a dedicated LoginActivity that will prompt users to enter their registered email and password. If users are new to the app or haven't registered, they can click on the "Sign Up button, which will redirect them to the RegisterActivity. In the RegisterActivity, users can provide their email and password for account creation. Upon successful registration, the credentials will be securely stored in the firebase database.
- c. *Database Validation and Toast Messages*: When users attempt to log in, the app will validate the entered credentials with the stored database records. If the provided email and password match, the user will be granted access to the main screen. Otherwise, an appropriate toast message will inform the user of invalid credentials or the need to register.
- d. *Firebase Realtime Database*: It provides cloud-based data storage, allowing us to store recipe data securely on the cloud. This means that the app users can access the same data from different devices, and any updates to the database will be reflected in real-time across all connected devices. Firebase Authentication is utilized as it offers an easy-to-implement authentication methods for email/password, Google sign-in, Facebook sign-in, etc., ensuring secure access to the app's personalized features. From the backend only the author id and username are visible, and the password is hidden.
- e. *Main Screen*: The app's main screen/ home page will feature a search engine along with favorite and popular recipes for that profile. It will also have a floating action button to create new recipes and publish in the app.
- f. *Recipe Category Fragment*: Upon selecting a specific recipe category, the app will redirect users to a CategoriesFragment. This fragment will display all the recipes within the chosen category, presented in an organized manner using Image Views and Table Layout. Users can browse through the recipes and select the one they find interesting. These visually appealing representations will entice users to explore each category.
- g. *Profiles Fragment*: Here the users profile photo and user email will be visible along with the option to edit their details. And all the recipes added by the user to the app will also be visible in one pace for easy navigation later.
- h. *Recipe Details Activity*: When users click on a recipe from the Recipe Category Activity, they will be directed to a new page called RecipeDetailsActivity. This page will provide comprehensive details about the selected recipe, such as, cooking steps, preparation time, calories, and cooking time. The recipe details will be obtained from the database and presented as a series of steps with a scroll view, making it easy for users to follow the cooking process.

3. Technical Description

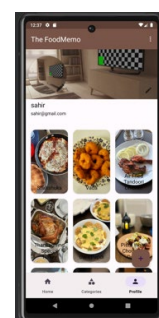
- a. *Login Activity*: Firebase Authentication is used to implement a login activity in an Android application. The *gotomain()* method is called when the user clicks the login button. It initially checks the user input to ensure that both the email and password boxes are not empty, that the email is in an acceptable format, and that the password is at least 6 characters long. If the input is correct, Firebase Authentication is launched, and the user's credentials are validated. When a user successfully logs in, a toast message appears, and the user is sent to the *MainActivity.java*. In the event of a login failure, the user is prompted to create an account. This method handles user authentication efficiently and offers suitable feedback during the login process.



- b. *Register Activity*: Users can create a new account by entering their email address and password in the RegisterActivity. The code validates the input to check that both fields are completed, the email is in a proper format, and the password is at least 6 characters long. If the input is validated, the `createnewuser()` method is invoked to start Firebase Authentication and create a new user account. A toast message indicates whether the account creation was successful. There is also a "Go to Login" button to return users to the login activity *LoginActivity*. Overall, this code allows users to register and create new accounts in the Android app utilizing Firebase Authentication in an easy and safe manner.

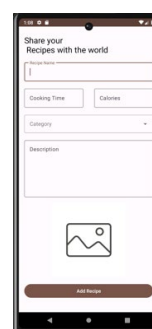


- c. *Main Activity*: This is the activity with a bottom navigation view with menu items, allowing for quick navigation between each fragment. A Fragment serves as a container for navigation via the Navigation Component. The design also has a clickable floating action button with a + icon in the bottom right corner, which is meant for custom actions. For the app's main screen, the combination of bottom navigation and floating action button produces a straightforward and user-friendly design. We have also incorporated an app bar which has an overflow button for the Log Out activity. `menu_main` is the resource file which contains the log out option. The `logout()` method is responsible for handling the log out functionality.



- d. *Home Fragment*: It is made up of a Constraint Layout that contains multiple UI elements. A title (*TextView*), a search bar (*TextInputEditText*), and two Recycler Views for presenting favorite and popular recipes are included in the layout. There are also "See More" links for each section to explore other recipes. Constraints are used in the design to efficiently position pieces within the layout. Overall, the Home Fragment style provides consumers with a user-friendly interface for searching for recipes and seeing popular and preferred recipe items. The code in this section is basically responsible for displaying recipes on the

home screen. The important methods in this class include *performsearch()* which extracts the search query from the *EditText*, *loadRecipes()* to load all recipes from the Firebase Realtime Database using a *ValueEventListener*, *loadpopularRecipes()* to randomly select five popular recipes from the list of all recipes and sets up the *HorizontalRecipeAdapter* to display them in a horizontal RecyclerView and *loadFavoriteRecipes()* to randomly select five favorite recipes from the list of all recipes.



- e. *Category Fragment*: This is designed to display culinary categories using data from the Firebase database. The layout is made up of a 'ConstraintLayout' and two primary components. First, a 'TextView' at the top serves as the headline, displaying the message "Your Culinary Journey Begins Here". A 'RecyclerView' is used below the title to dynamically display the culinary categories acquired from Firebase. The 'RecyclerView' is set to be vertical, and a 'LinearLayoutManager' efficiently arranges the layout. This design delivers an engaging and user-friendly experience for customers to explore various culinary categories, enriching their gastronomic adventure with smooth Firebase data integration. The *loadCategories()* method retrieves the list of categories from the "Categories" node in the Firebase Realtime Database using a *ValueEventListener*.
- f. *Profile Fragment*: This part oversees managing the user's profile information, which includes name, email, and profile image using the *loadprofile()* method. Users can modify their profile photo using *uploadImage()* and cover image using *uploadCoverImages()* by uploading images from their device to the fragment. The uploaded photographs are preserved in Firebase Storage, and the image URLs are saved under the user's profile in the Firebase Realtime Database. The fragment also displays the user's uploaded recipes in a grid layout using a RecyclerView, which is obtained from the Firebase database using the user's unique ID. Overall, by integrating Firebase for data storage and retrieval, ProfileFragment gives users control over their profile details and recipe uploads.
- g. *RecipeDetailsActivity*: This code is responsible for displaying the details of a specific recipe. *init()* method initializes the UI components and sets up the recipe details like name, calories, category, description, and recipe image using the Glide library. If the current user is the author of the recipe, it displays an "Edit" button. *onResume()* method is called when the activity is resumed, and it retrieves the recipe ID from the intent. It then calls the *updateDatawithFireBase()* method to fetch and display the recipe details from Firebase Realtime Database.
- h. *AddRecipeActivity*: It allows users to add or edit an existing recipe. Users can input recipe details like name, description, cooking time, category, and calories. They can also select an image for the recipe, which is then uploaded to Firebase Storage.

The recipe data, along with the image URL, is saved to Firebase Realtime Database. The `retrieveRecipeData()` method fetches a specific recipe from Firebase Realtime Database using the provided recipe ID and populates the UI components with the recipe data and the `saveDataInfirebase()` method saves the recipe data to Firebase Realtime Database, including the image URL obtained from `uploadImage()`. It displays a progress dialog during the upload process and shows a success or error message based on the upload result.

- i. *Firebase*: In this project we have used Firebase instead of SQLite as it has an edge over the other. Firebase is a powerful choice for applications that require real-time updates, seamless synchronization, and cloud-based infrastructure management. Additionally, Firebase provides a scalable infrastructure that can handle a growing user base and offers security measures to protect user data.

