

Design

Screen Hierarchy

This section will describe the structure of the app's navigation, that is, how the different activities are connected and what flow users follow when using the app.

The Meal app has a well-defined and user-friendly screen hierarchy. Screen hierarchy had been made to ensure intuitive navigation and ease of access to all core functions. The center of the whole structure is `MainActivity`, which acts as the home dashboard. From there, users can access all major functions of the app using clearly labelled menu icons. Below is the hierarchy:

`LoginActivity` → `MainActivity`

`SignupActivity` → `MainActivity`

`MainActivity`

 |
 | └── `MealRecipeActivity`

 |
 | | └── `AddRecipeActivity`
 | | └── `RecipeDetailActivity`

 |
 | └── `MealPlannerActivity`

 |
 | └── `ItemDelegationActivity`

 └── `Logout` → `LoginActivity`

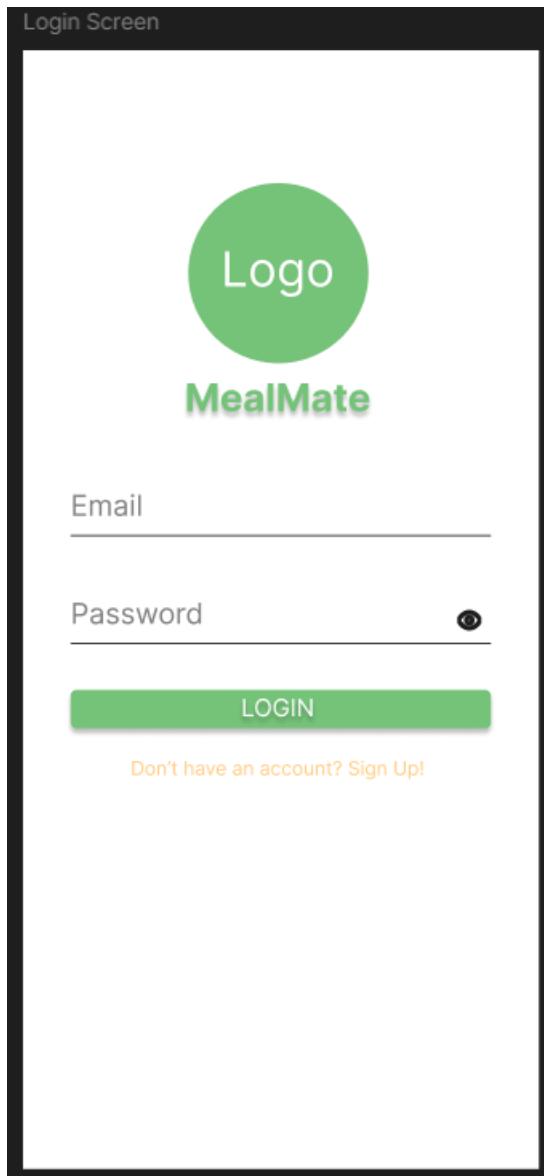
The first point of interaction is the `LoginActivity`. When the app launches, it checks whether a user is already logged in using Firebase Authentication. If the user is not logged in, they are directed to the `LoginActivity`. It will also have a link to the `Signup` screen for new users. Once authenticated, users are directed to the Main screen.

From the Main screen, users can navigate to four main sections: Meal Recipes, Meal Planner, Share Shopping List, and Logout. The Meal Recipes section leads to a list of all recipes, with the ability to add or view detailed information. Meal Planner shows all tracked meals and allows ingredient management. Share Shopping List lets users send ingredients of selected meals via SMS. The Logout option signs the user out and returns them to the Login screen.

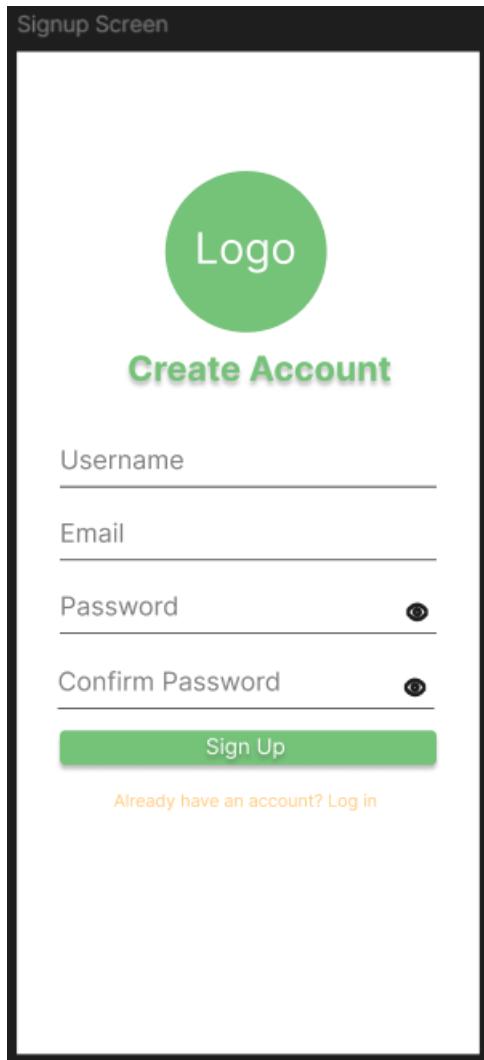
Each feature is only one or two taps away from the main dashboard. This means, this structure provides a natural and consistent user experience. It also supports both linear and hub-and-spoke patterns which make it efficient for users who want to focus on one task at a time or quickly switch between them.

Wireframes

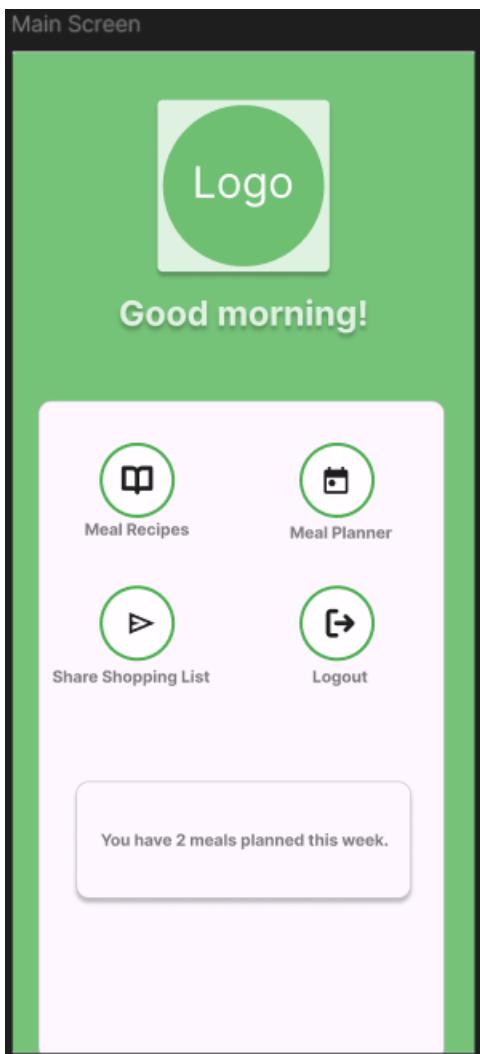
The visual design of MealMate follows a user-centered approach. Each screen was first made as a wireframe before development to focus on clarity, and intuitive layout. The goal was to make the app have a streamlined user experience across all functions.



The Login screen of the app is simple and direct. It has a centered logo at the top followed by two input fields: one for the user's email and another for the password. Both fields are styled with contrasting text and hints for accessibility. Below the login button is a text link for new users to navigate directly to the Signup screen. The layout is vertical scanning, making it easier for first-time users.

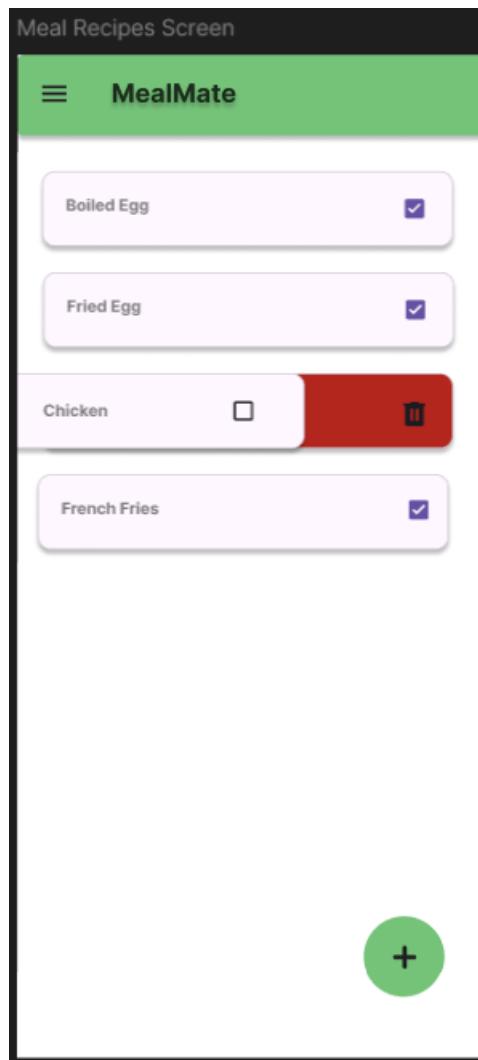


The Signup screen builds upon the login screen design by adding more fields for the username and password confirmation, arranged similar in vertical structure. The password fields have password toggle icons which improve usability by allowing the user to verify their input. The buttons are styled and positioned with spacing to avoid accidental presses. There is also another navigation link directly to the login screen.



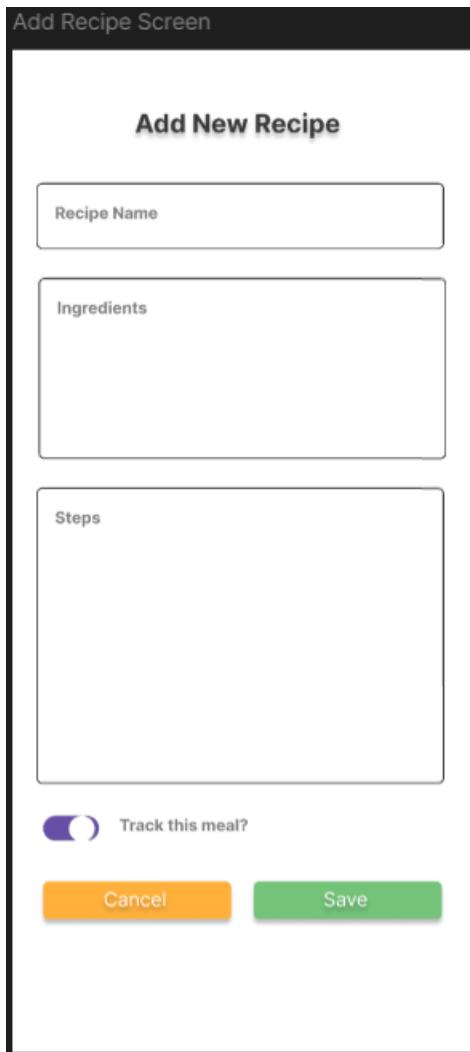
The Main screen is the user's entry point to the app after login. It is split into two contrasting-colored sections. The upper green section features a welcoming greeting that adapts to the time of day and an app logo. This screen uses the app's colors to create visually appealing looks. The lower section, styled with a white card made to look like floating, has a grid layout with four menu items. Menu items are Meal Recipes, Meal Planner, Share Shopping List, and Logout. Each menu item is represented by a circular

space with an icon and a label. The grid has easily tappable spacing. Below the grid is a static card. It displays a random meal-related fact, either how many meals are planned, completed, or still remaining this week. The card adds a sense of feedback and encourage user to regularly engage with the app.

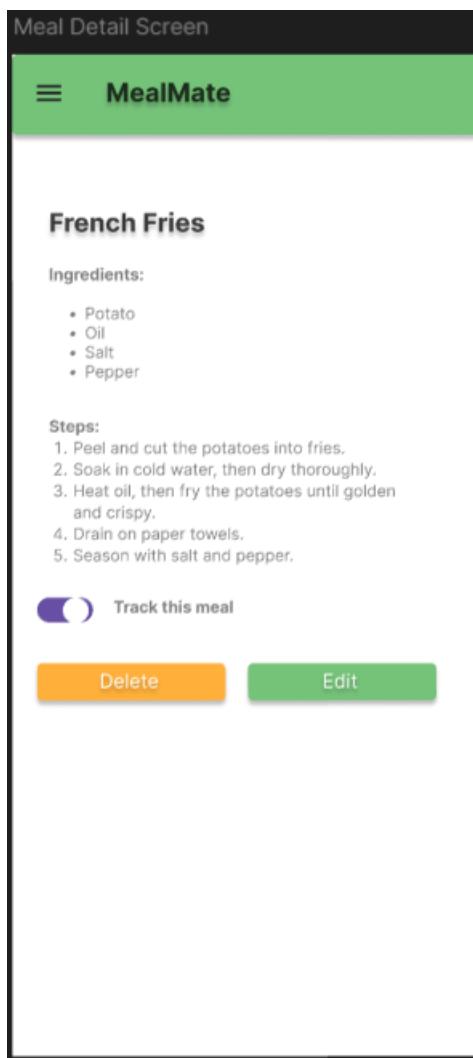


The Meal Recipes screen uses a top Material Toolbar and scrollable list layout. Each meal appears as a rectangular card, showing the meal name and a tracking checkbox to the far right. The swipe-to-delete gesture is supported. It brings a modern and intuitive feel of managing items to the app. There is a floating

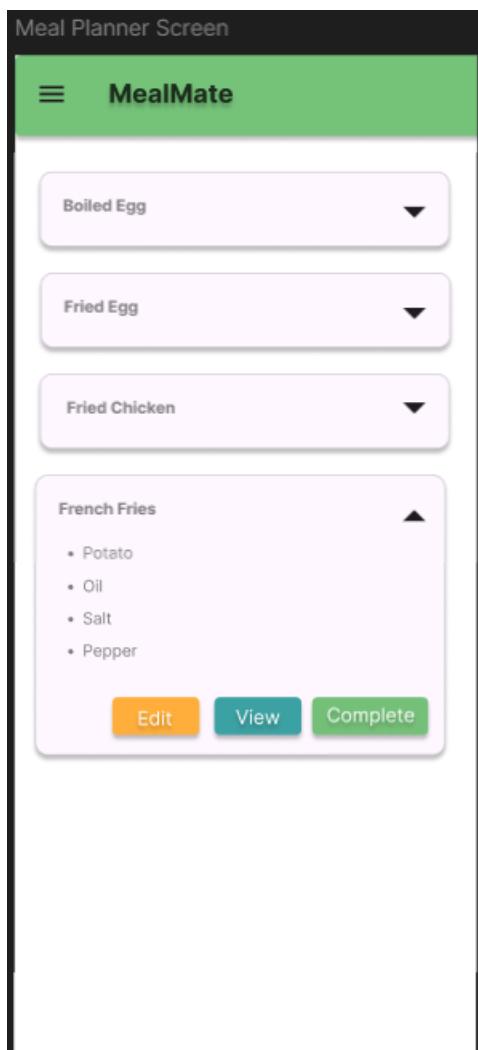
action button at the bottom right. It allows the user to add a new recipe. It also follows Google's Material guidance on prominence and accessibility for primary actions. The recipe list updates in real time based on Firestore changes.



The Add Recipe screen combines structured input with visual clarity. It includes three text fields, recipe name, ingredients, and steps. They are styled to creates a clear distinction from app background. A “Track this meal?” switch is provided. It is used to determine whether the meal should appear in the planner. The switch uses color cues, green for on, grey for off, to visually communicate its state. There are the save and cancel buttons at the bottom in a side-by-side layout. They have strong color differentiation and style to separate them from the form.

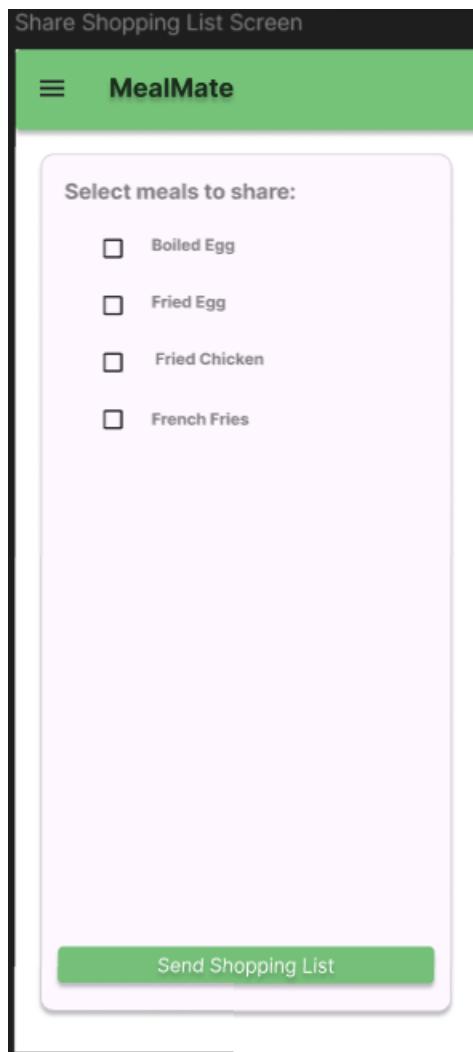


The Meal Detail screen is made to present full meal information clearly. It starts with the meal name in large bold text, followed by labeled lists for ingredients and steps. These are presented using vertical spacing and clear section headers. This makes the screen easy to scan. Below the content is a toggle switch like before. At the bottom are two buttons, Edit and Delete, presented side by side. With these buttons, the user can review and take action on a recipe without navigating elsewhere.



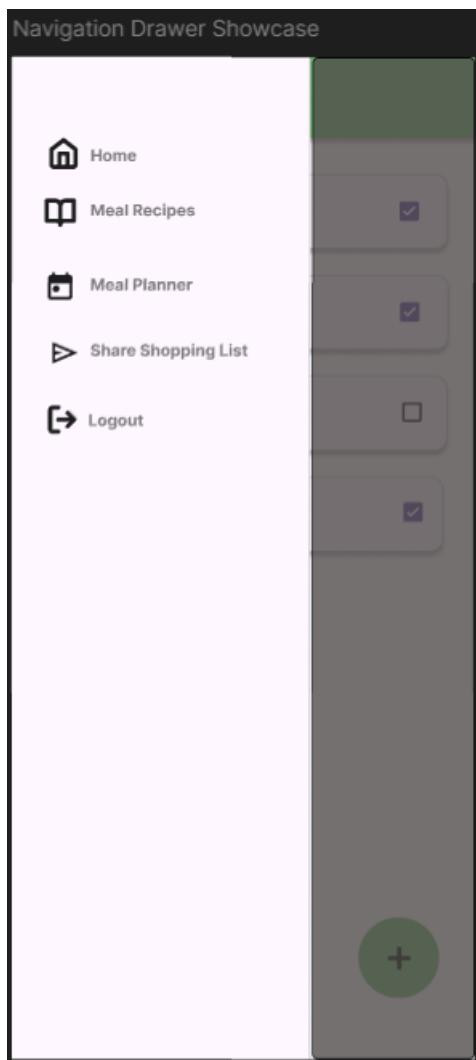
The Meal Planner screen builds on the vertical card list pattern but introduces interactivity on multiple layers. Each tracked meal is displayed in an expandable card that shows its ingredient list. The card has the same visual styles with the recipe cards in the Meal Recipes screen but includes additional buttons: a “View” button to open the full recipe, an “Edit” button to modify ingredients directly in the place, and a “Complete” button that checks if all ingredients are marked as purchased. The ingredient list uses checkboxes to track purchase status, with strikethroughs applied for purchased items. All of this is in

synced with Firebase. Meaning there is a continuity across sessions and devices. The design prioritizes efficiency for meal planning.



The Share Shopping List screen follows a simpler checklist pattern. A top Material Toolbar with navigation and title also exist here for consistency across the app. Below it, the screen prompts the user to select meals to share. Tracked meals are listed with checkboxes. The user can select multiple meals and tap

the “Send Shopping List” button at the bottom to share ingredients via SMS. The minimal layout support quick action without distractions.



The Navigation Drawer is present in all major activity screens using the `NavigationView` component. It includes links to Home, Meal Recipes, Meal Planner, Share Shopping List, and Logout. Each item is clearly labeled and uses appropriate icons for recognition. The drawer opens with a hamburger icon in

the top app bar and can be close by tapping on the negative space or by simply swiping left. By using a consistent drawer across the app, the user will always know how to navigate.

Functionality

The MealMate is built using native Android development with Java and the Firebase suite for user authentication and cloud-base data storage. The main functions include login, signup, recipe management, ingredient tracking, and shopping list sharing. The main functions also made use of real-time UI updates with firebase and gesture-based interactions. Each of these features were implemented to be responsive, consistent with Material Design, and optimized for real world usage.

User Authentication

User authentication is handled with Firebase Authentication, using email and password login. The app check whether the user is signed in upon launching, and redirect them to the right screen accordingly. A session persists as long as the user does not log out. This way the user will have uninterrupted experience across uses and does not have to log in every time they open the app.

Meal Recipes

In the Meal Recipes screen, users can add, view, edit, or delete recipes. Each recipe has a name, a list of ingredients (which are input as newline-separated strings), and step-by-step instructions. There is also a toggle for whether the meal should be tracked for planning. Data from form inputs is gathered and uploaded as a Firestore document.

Firestore's NoSQL structure mean storage and retrieval are efficient. Each recipe document contains fields for "name", "ingredients" (an array), "steps", and "tracked" (a boolean). Editing a recipe preloads the form with existing data, and updates are applied to the existing Firestore document. The Firestore document IDs allows for accurate updates and deletions. For the UI, **RecyclerView** cards are used and users can swipe left to delete an item or tap to view full details.

Meal Planner

When a recipe is marked as tracked, it appears in the Meal Planner screen as expandable cards. Each card shows its list of ingredients and includes buttons like View, Edit, and Complete. View button opens the recipe detail. Edit allows editing the list of ingredients inline and Complete checks if all ingredients are purchased and displays a congratulation message. Ingredients can be edited inline when users tap "Edit" button.

I also used the device's accelerometer via **SensorManager** and a **SensorEventListener** make a shake gesture function. Once a shake is detected and the user confirms via a dialog, the app will reset the

purchased state of the ingredients. This gesture simplifies list management for users who have completed shopping and want to keep the meal tracked for the next time.

Share Shopping List

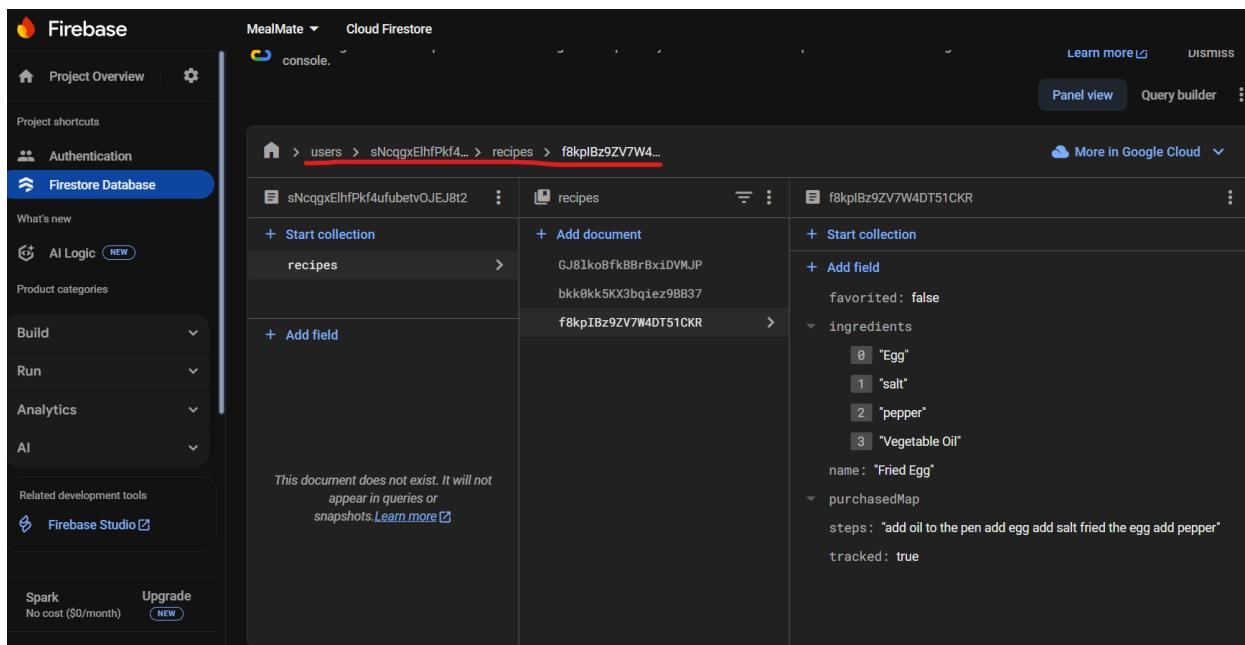
With this feature, users can select one or more tracked meals and send a combined list of their ingredients. This feature uses Android's SMS intents allow users to send the shopping list to a phone number.

The UI presents a checklist of tracked meals using a **RecyclerView**. When the user taps Send Shopping List, the app aggregates the meal names and their ingredients and creates a formatted string for SMS. This feature requires no external SMS API or permission beyond intent launching. It also makes use of Android's implicit intent to open the user's preferred messaging app.

UI and Storage

From a UI perspective, key functionalities such as swipe-to-delete, switch toggles, expandable cards, and form validation were implemented using standard Android widgets with help with Material Components like **MaterialToolbar**, **CardView**, and **TextInputLayout**. Gestures use **ItemTouchHelper** and custom **RecyclerView.Adapter** logic.

The backbone of the app's functionality is Cloud Firestore. The app is fully integrated with it for data persistence, structured per user. It is a NoSQL document-based database provided by Firebase. All user data, including recipes, tracked meals, ingredients, and purchase states, is stored under a user-specific path (**users/{userId}/recipes**).



The screenshot shows the Firebase Cloud Firestore interface. On the left, the sidebar includes Project Overview, Authentication, Firestore Database (selected), AI Logic (NEW), Product categories, Build, Run, Analytics, AI, and Related development tools (Firebase Studio). The main area displays a hierarchical view of a document structure:

- Path: users > sNcqgxElhfPkf4ufubetvOJEJ8t2 > recipes > f8kpIBz9ZV7W4DT51CKR
- Collection "recipes" contains documents: sNcqgxElhfPkf4ufubetvOJEJ8t2, GJ81koBfkBBxBxiDVMJP, bkk0kk5KX3bqiez9BB37, and f8kpIBz9ZV7W4DT51CKR.
- Document "f8kpIBz9ZV7W4DT51CKR" contains fields:
 - favorited: false
 - ingredients
 - 0 "Egg"
 - 1 "salt"
 - 2 "pepper"
 - 3 "Vegetable Oil"
 - name: "Fried Egg"
 - purchasedMap
 - steps: "add oil to the pen add egg add salt fried the egg add pepper"
 - tracked: true

This way each user only accesses only their own data which is good for both privacy and scalability. Firestore's real-time sync capability makes it easy to reflect updates immediately after operations like adding a meal, editing ingredients, or toggling tracking. Firestore was chosen for its real-time sync, scalability, and easy **FirebaseAuth** integration.

Overall, MealMate use Firebase services and Android's material components to make a structured, efficient, and user-friendly experience. The functionalities are built to be modular and scalable. This way it is easy to add more features in future.

Testing

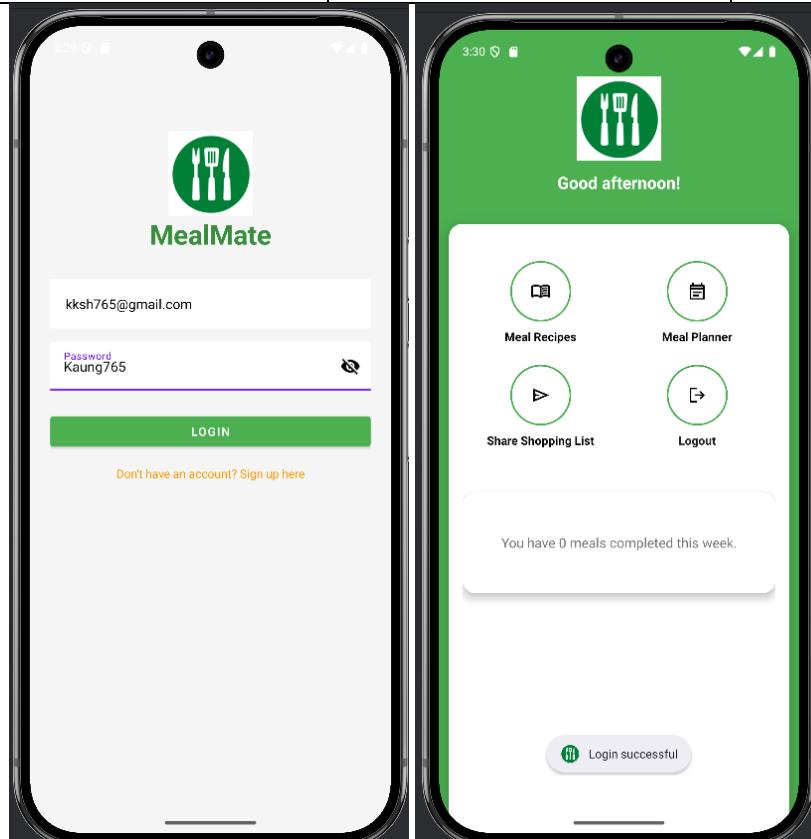
Black-box testing was the main testing strategy for evaluating the functionality of the app. The testing was from a user perspective, focusing on input and output behavior without the knowledge of the internal logic. This is good for testing features the rely on UI such as login, adding recipe, and ingredient tracking. Some Gray-box testing was used to verify whether Firestore stored and retrieved recipe and meal planning data as intended.

Functional testing was performed to confirm the function of key app components such as user authentication, recipe create, read, update, delete operations, meal planning, ingredient tracking and SMS integration. Non-functional tests were also included, they are UI responsiveness and usability, across different Android devices. For compatibility both an emulator and a real Android device was used for testing.

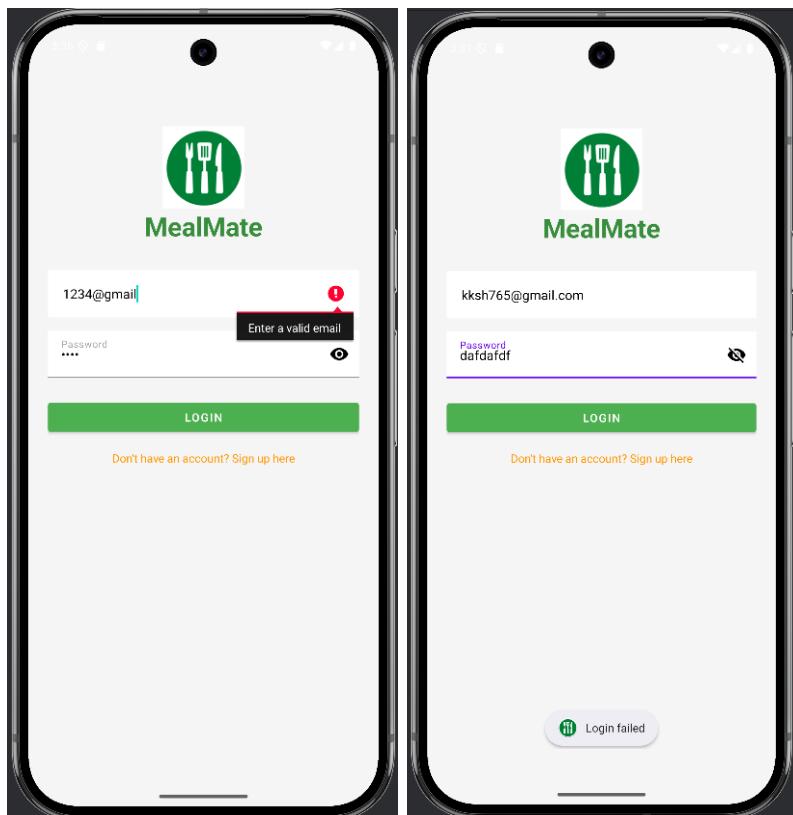
Functional Test

No.	Test Description	Expected Result	Performed as Expected?	Issue & Status
1	Login with correct credentials	User is logged in and taken to the main screen	Yes	N/A
2	Login with wrong credentials	Error message appears and login is prevented	Yes	N/A
3	Sign up with valid details	Account created and taken to the main screen	Yes	N/A
4	Sign up with missing or invalid details	Error shown and sign up failed	Yes	N/A
5	Logout by tapping logout button	Returns to Login screen	Yes	N/A
6	Add a new recipe with all fields	Recipe is saved in Firebase and shown in list	Yes	N/A
7	Edit a recipe	Recipe is updated in Firebase and update reflected in app	Yes	N/A
8	Delete a recipe with swipe left motion	Recipe is removed from Firebase and list	Yes	N/A

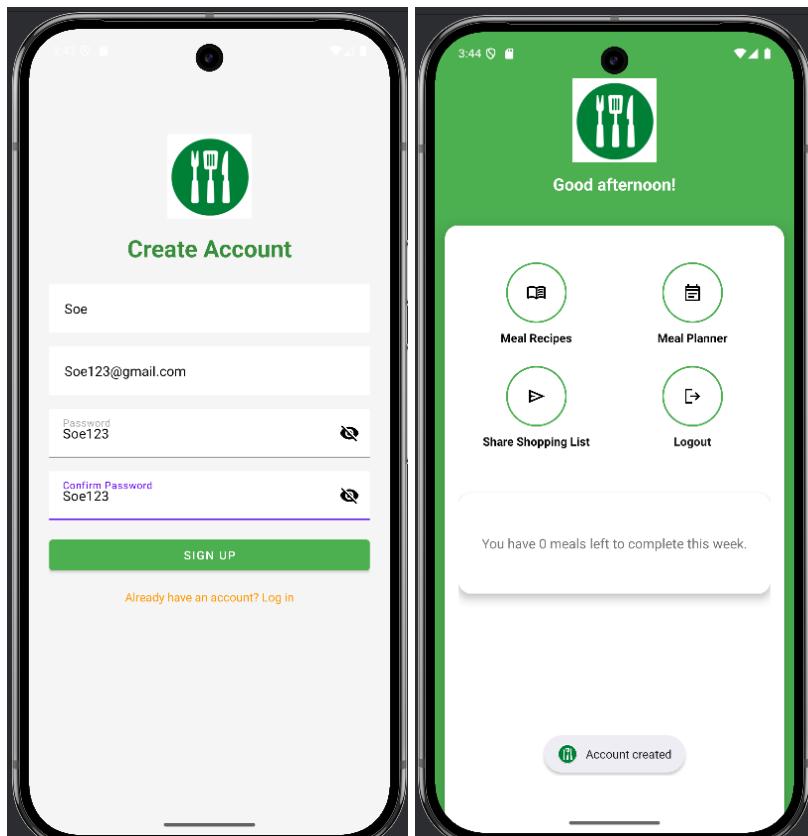
9	Mark meal as favorite with swipe right motion	Star icon appears and item moves to top	Yes	N/A
10	Unmark meal as favorite with swipe right motion	Star icon disappears	Yes	N/A
11	Track a meal	Meal appears in planner	Yes	N/A
12	Open a detailed screen of a meal by tapping on it	Detailed screen opens with data	Yes	N/A
13	Mark ingredient as purchased	Ingredient is marked with strikethroughs and synced to Firebase	Yes	N/A
14	Shake device to clear purchased ingredients	Ingredient marked as purchased are now unmarked	Yes	N/A
15	Complete a meal by marking all ingredients as purchased and tapping Complete button	Success dialog shown and the meal untracked	Yes	N/A
16	Send selected meals via SMS	SMS app opens with formatted list	Yes	N/A
17	Expand and collapse meal card in planner	Ingredients are shown/hidden	Yes	N/A
18	Show weekly meal stats on main screen	Random stat message appears with correct values	Yes	N/A



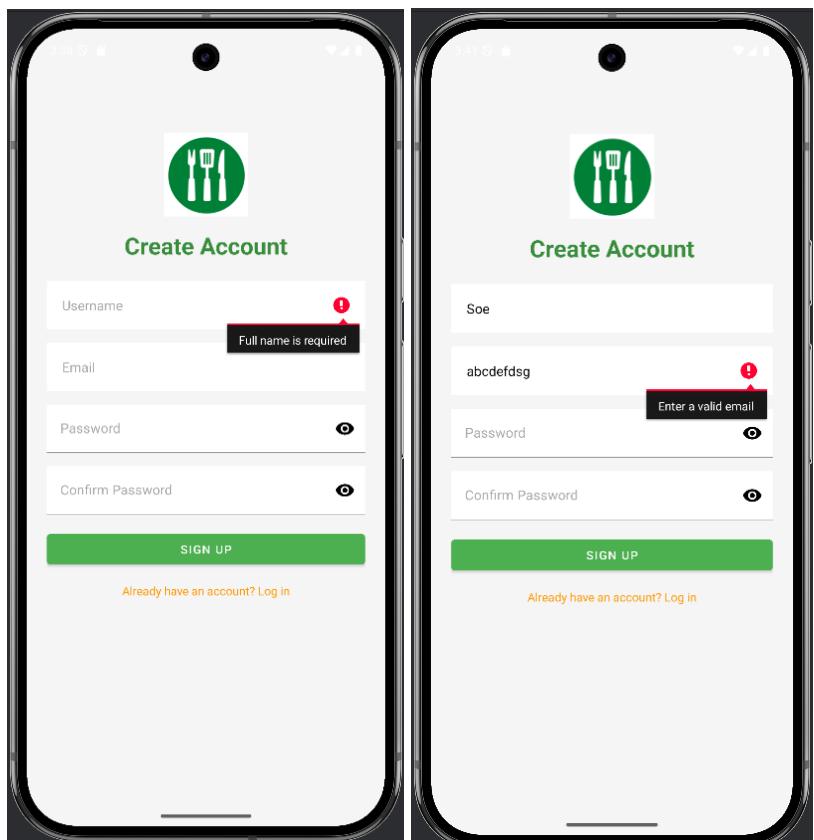
Test case 1



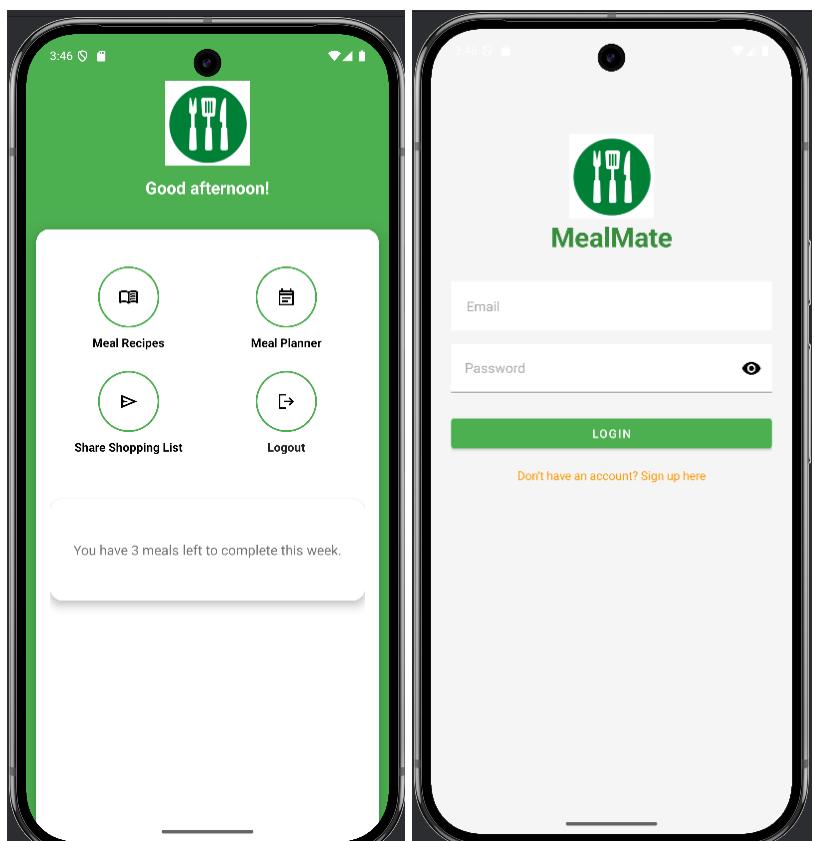
Test case 2



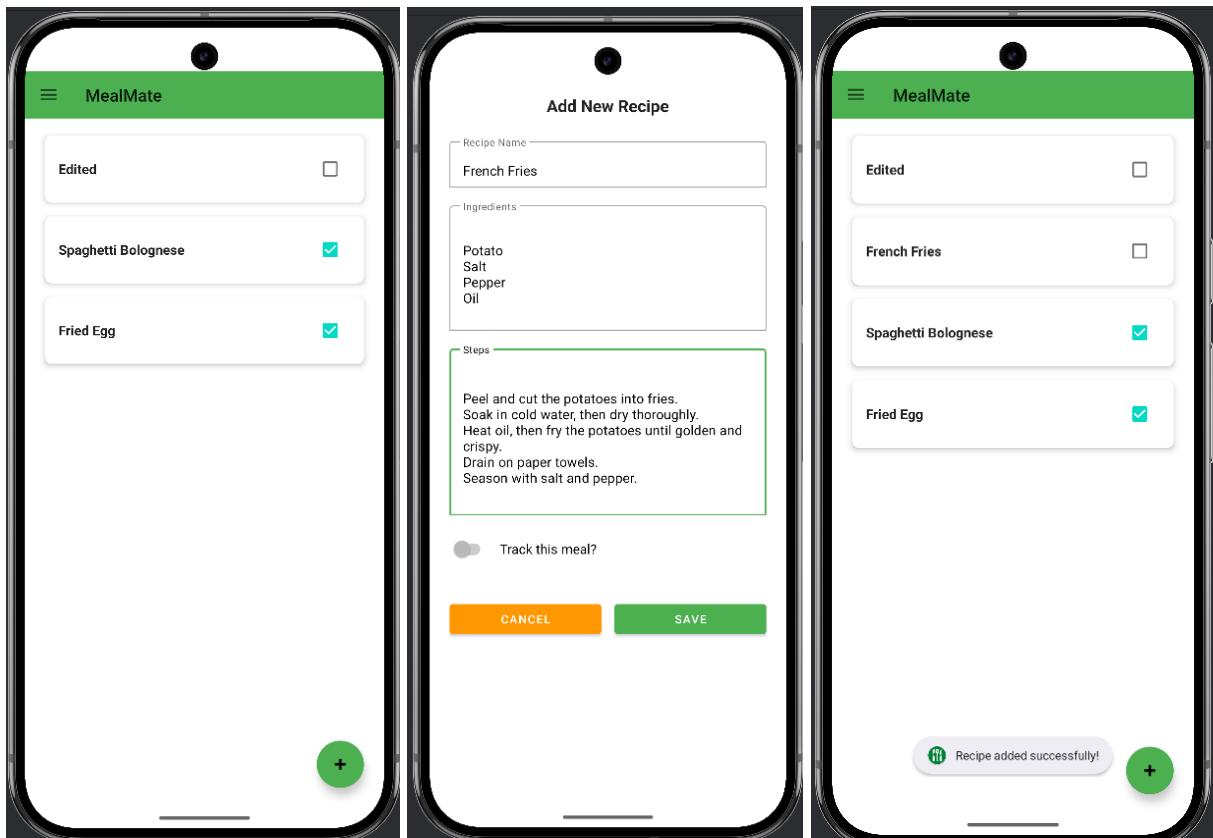
Test case 3



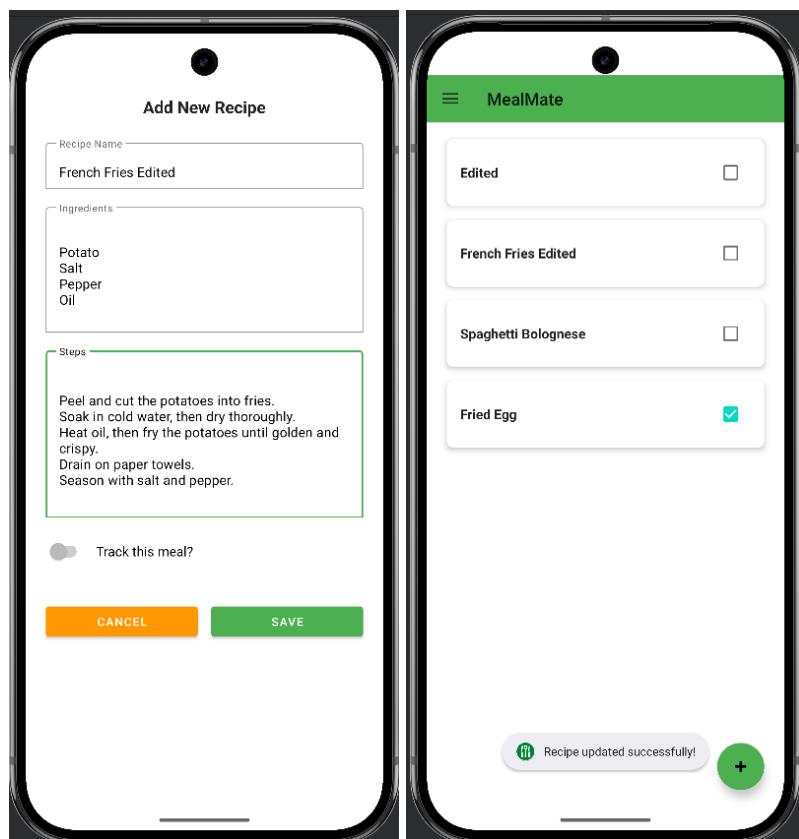
Test case 4



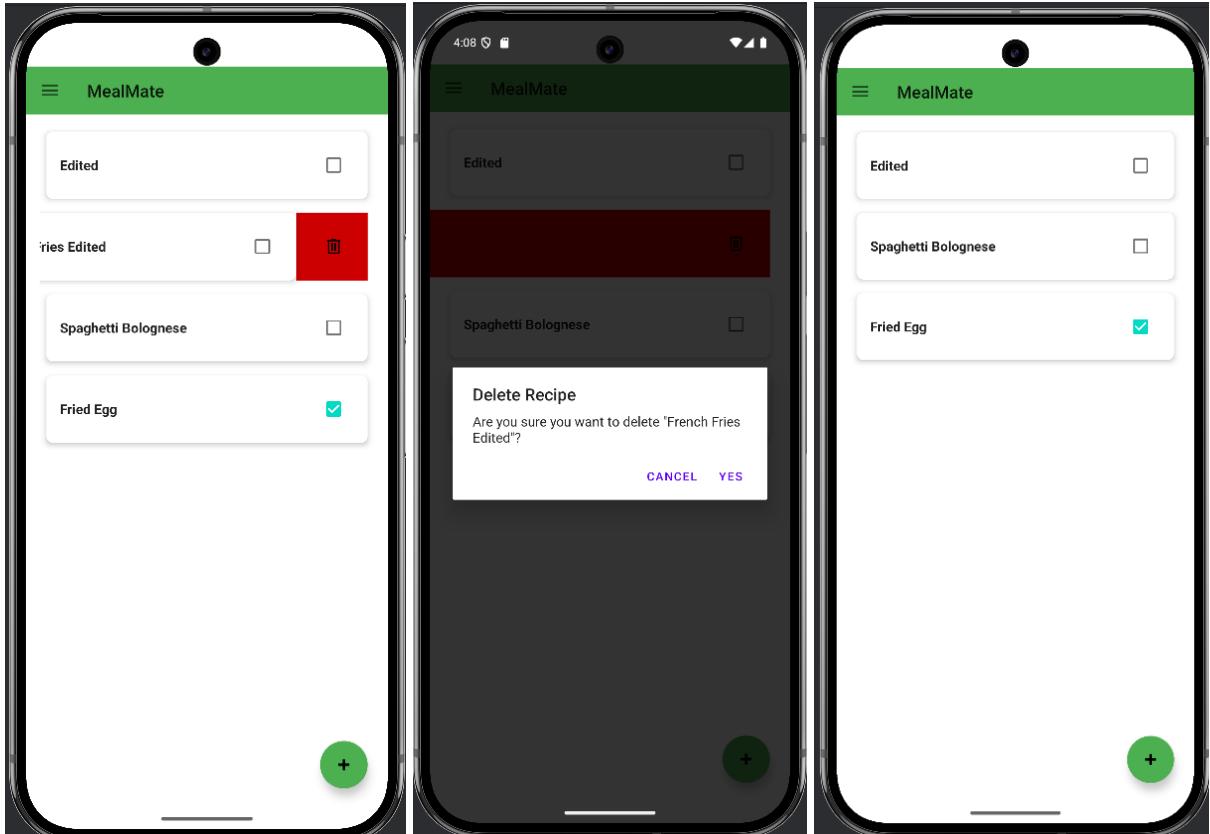
Test case 5



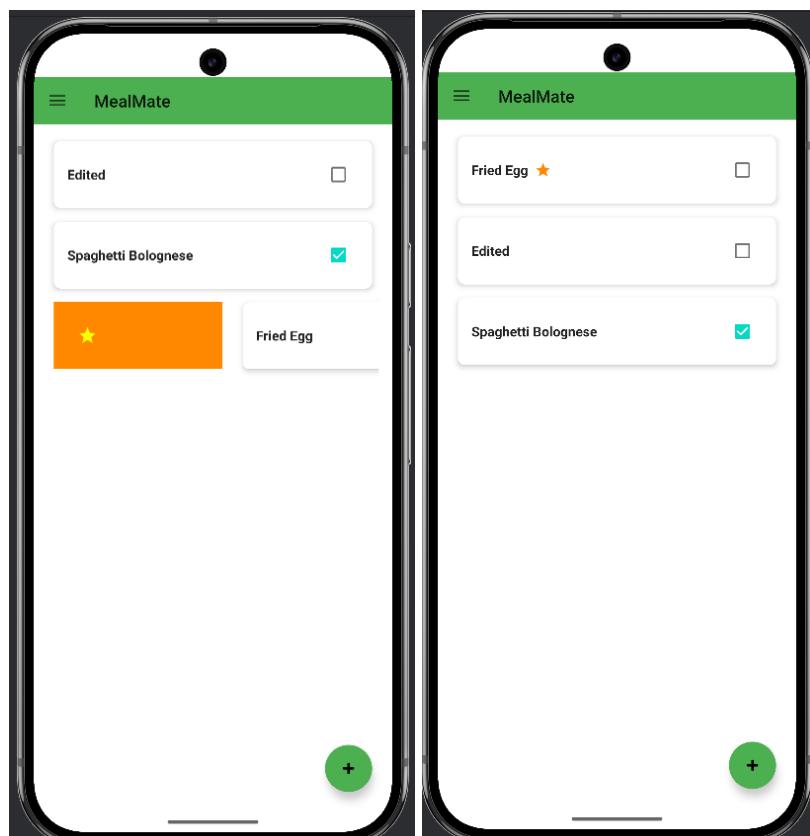
Test case 6



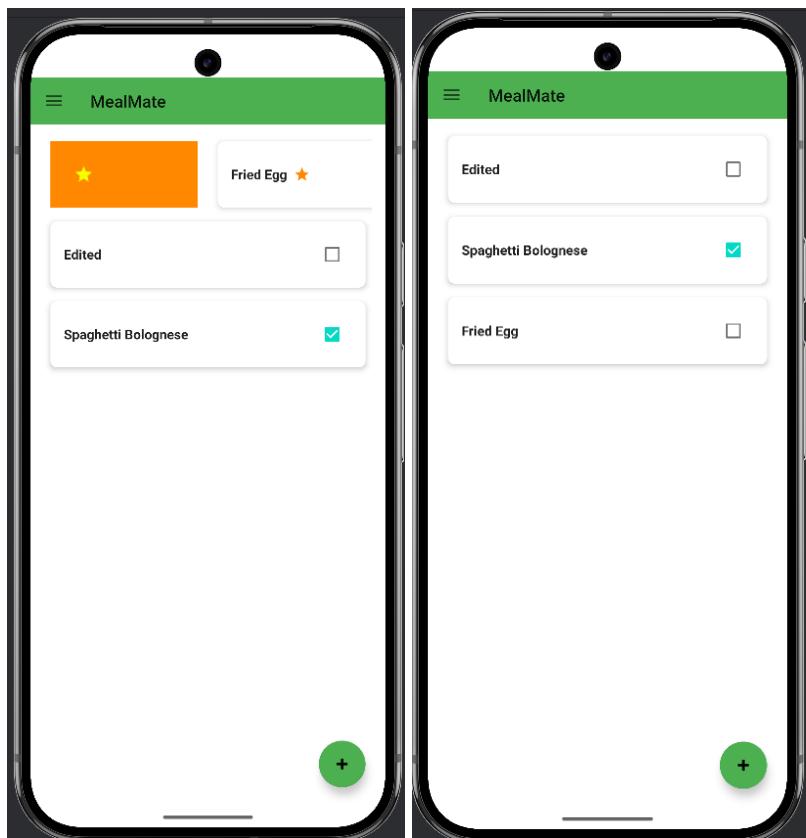
Test case 7



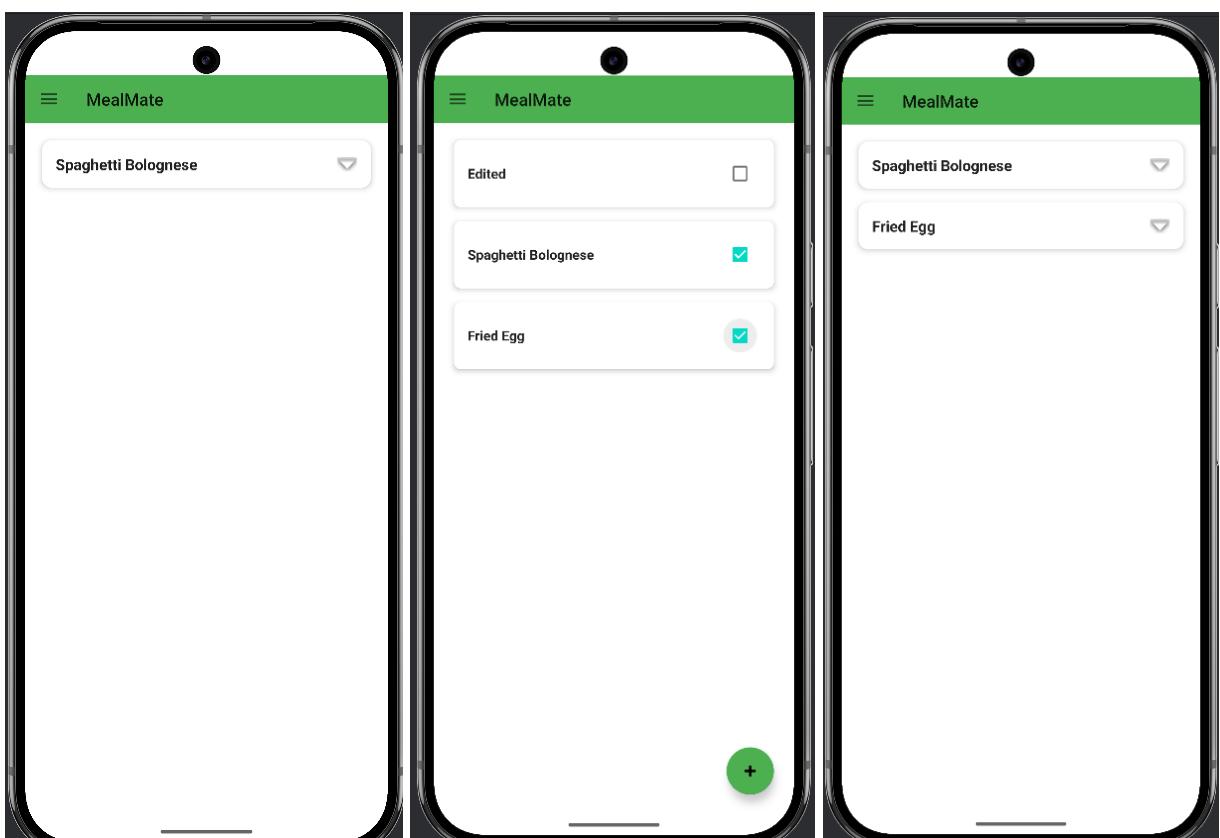
Test case 8



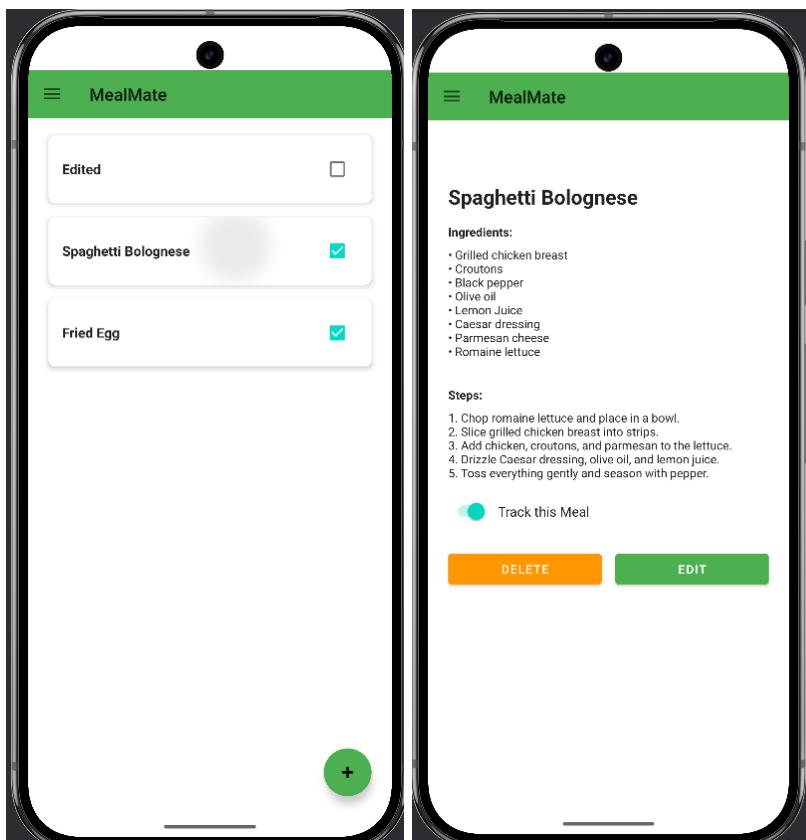
Test case 9



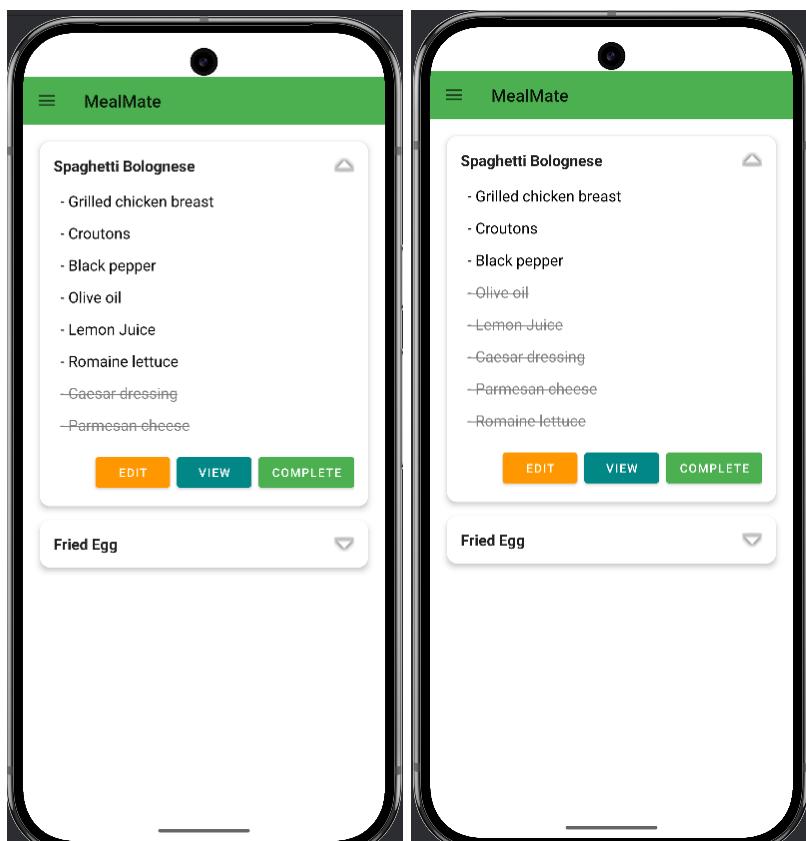
Test case 10



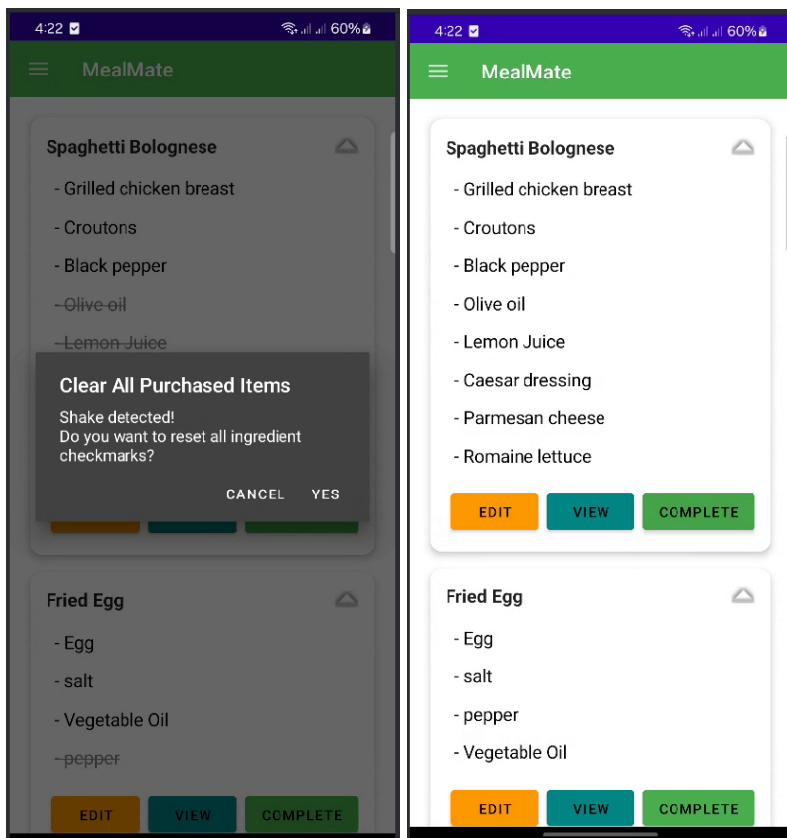
Test case 11



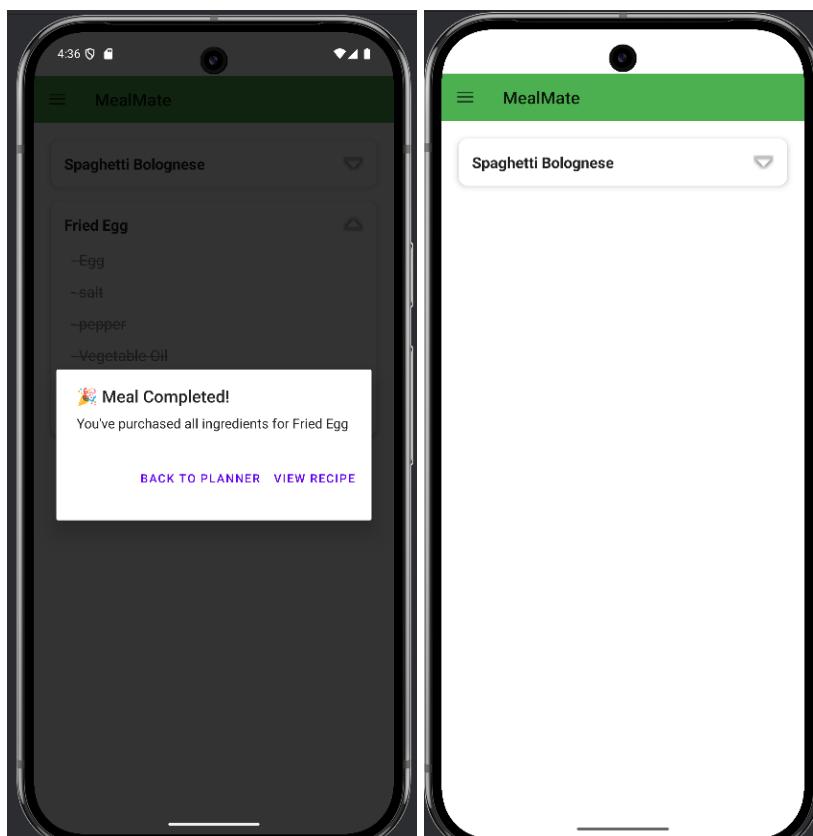
Test case 12



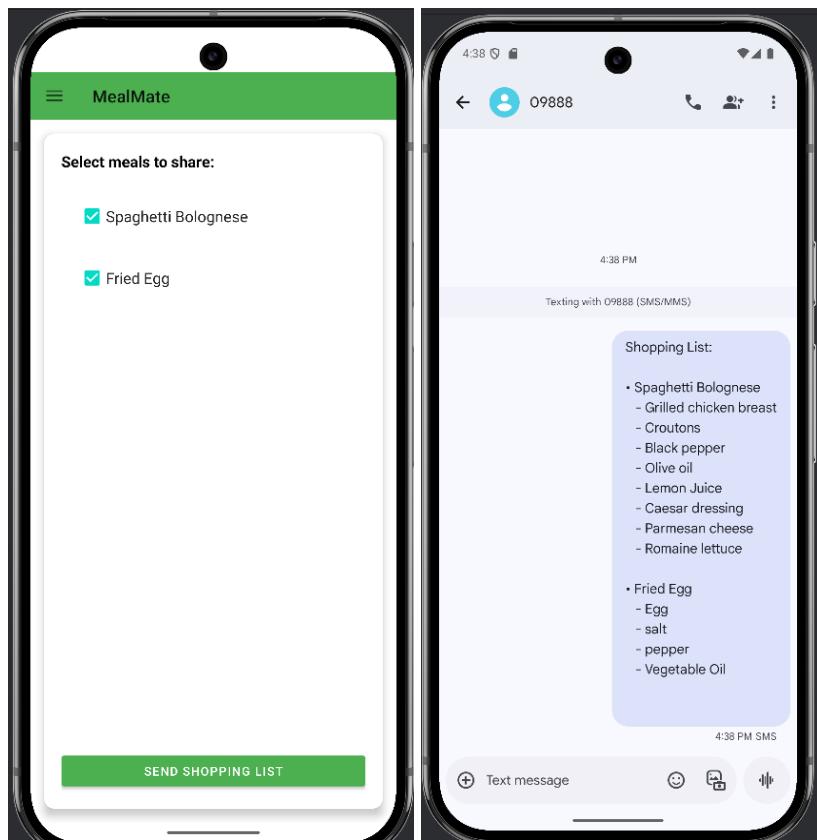
Test case 13



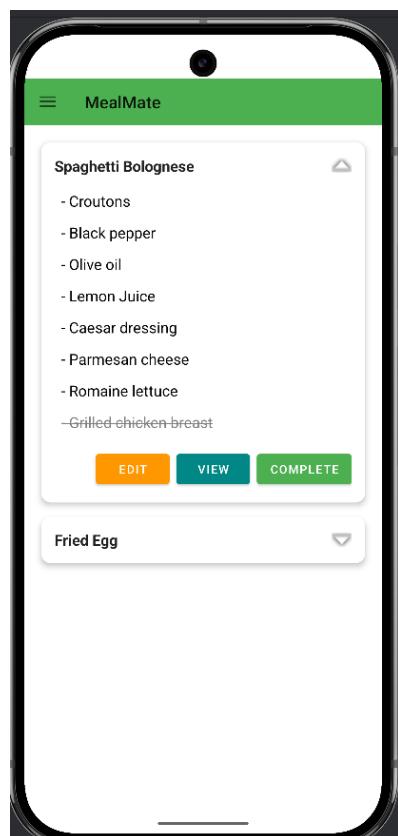
Test case 14



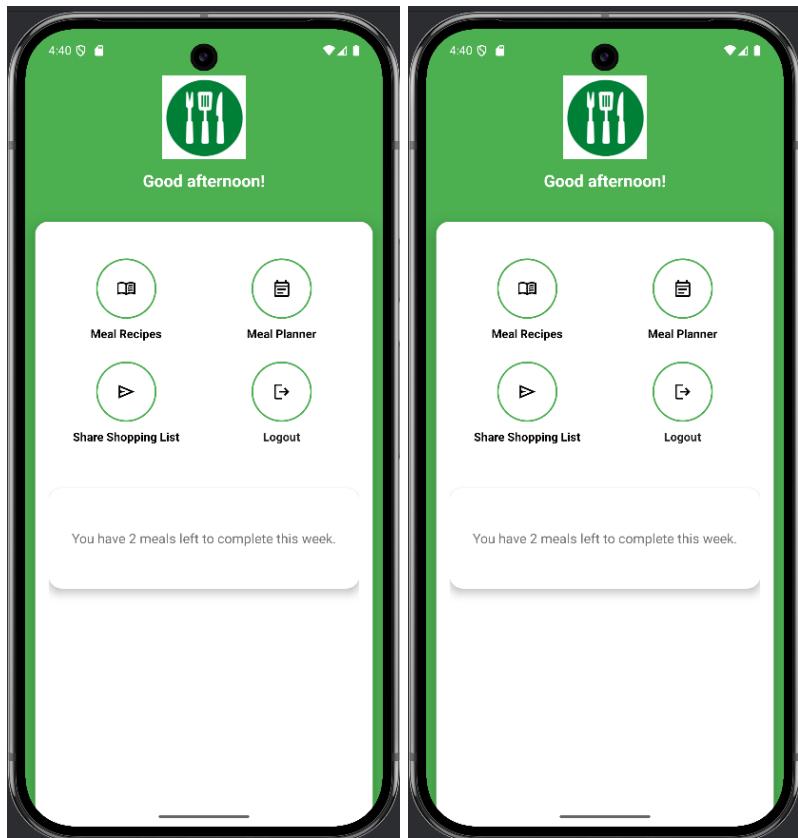
Test case 15



Test case 16



Test case 17



Test case 18

Non-functional Test

Performance Test

No.	Test Description	Expected Result	Performed as Expected?	Issue & Status
1	Load main activity	Load within 2 seconds	Yes	N/A
2	Open Meal Planner Screen with 10+ meals	Meals and ingredients are shown within 3 seconds	Yes	N/A
3	Save recipe to Firestore	Completes within 2 seconds	Yes	N/A

Test case 1: The home screen was opened repeatedly after login. Load times were consistently under 2 seconds.

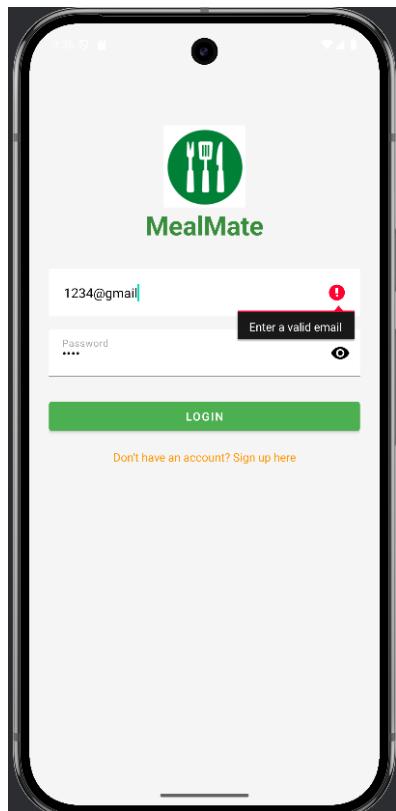
Test case 2: The Meal Planner screen was tested with over 10 tracked meals, each with multiple ingredients. The full list displayed in under 3 seconds after opening the activity, with expand/collapse animations working smoothly.

Test case 3: After tapping the save button, the recipe was uploaded to Firestore within 1 seconds and appeared in the recipe list almost instantly.

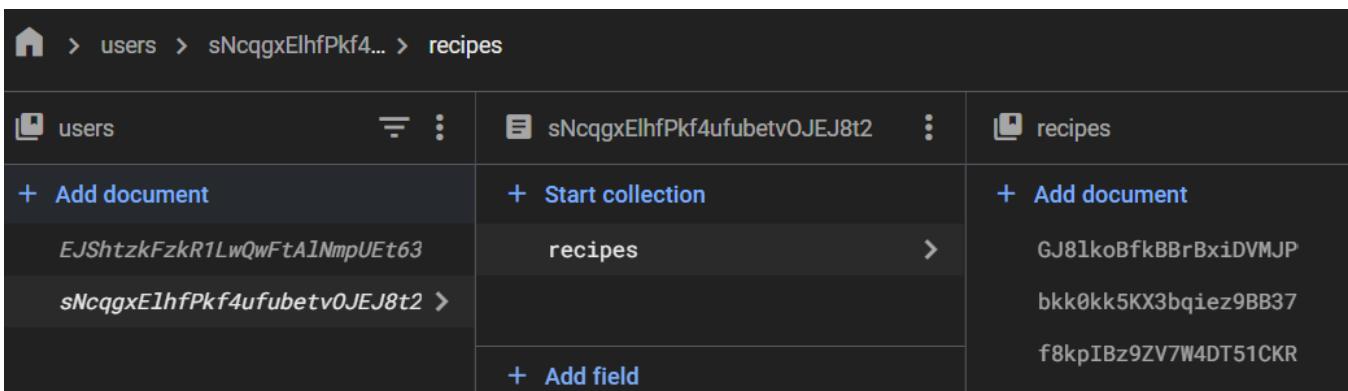
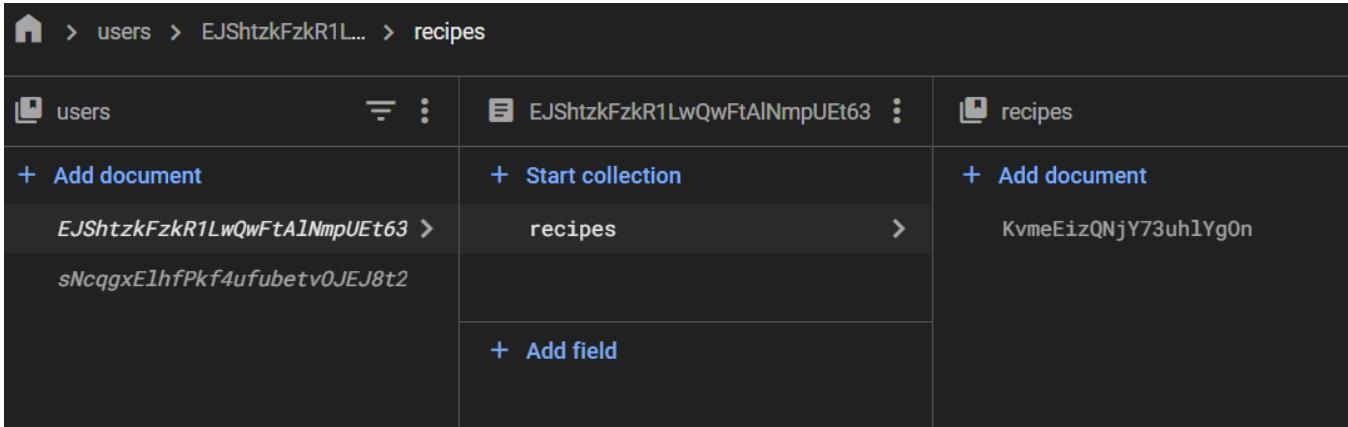
Security Test

No.	Test Description	Expected Result	Performed as Expected?	Issue & Status
1	Try accessing app without login	Redirected to login screen	Yes	N/A
2	Enter invalid email format	Error message shown, can't proceed	Yes	N/A
3	Use Firebase rules to restrict access	Users can only access their respective recipes	Yes	N/A

Test case 1: The app was launched after clearing session data. Attempts to access any screen without being logged in redirected the user back to the Login screen.



Test case 2: An email field was filled with an invalid format like "1234@gmail.com" and submitted. The input triggered built-in validation, showing an error message and preventing log in or sign up.



Test case 3: Firebase rules were configured so that users could only access their own documents under `/users/{userId}/recipes`.

Compatibility Test

No.	Test Description	Expected Result	Performed as Expected?	Issue & Status
1	Test on emulator and real Android device	Layout adjusts properly, no crashes	Yes	N/A
2	Enable android's system dark mode	Layout adapts, white background remains visible	Yes	Adjustments were made to force light background and black texts

Test case 1: The app was tested on both a Pixel emulator and a Samsung Galaxy A24. All screens and interactions (including gestures and navigation drawer) functioned as expected. UI layout adapted correctly.

Test case 2: When Android's dark mode was activated, the app maintained a forced light background for readability. Text elements retained black colors.