MYRECIPE

MyRecipe

Name: Matthew Hibshman
EID: mah8639
Email: matthibshman@gmail.com
October 16, 2022

APP DESCRIPTION

Summary

MyRecipe is a food recipe discovery and management app. Users will discover new dishes to cook, lookup recipes they have saved, and share recipes with other users. Through a “Tinder-like” discovery mechanism, users will see a photo of a prepared dish, filtered by diet or allergies if applicable, and swipe left or right to skip to the next dish or save the dish to their recipes. Users will be able to view the recipe for the dishes they have matched on, mark them as cooked, save their favorites, and share with other users through the application or through other messaging apps on Android. The experience will be delivered through a pleasing, minimal yet interactive UI based on Material Design with relevant animations.

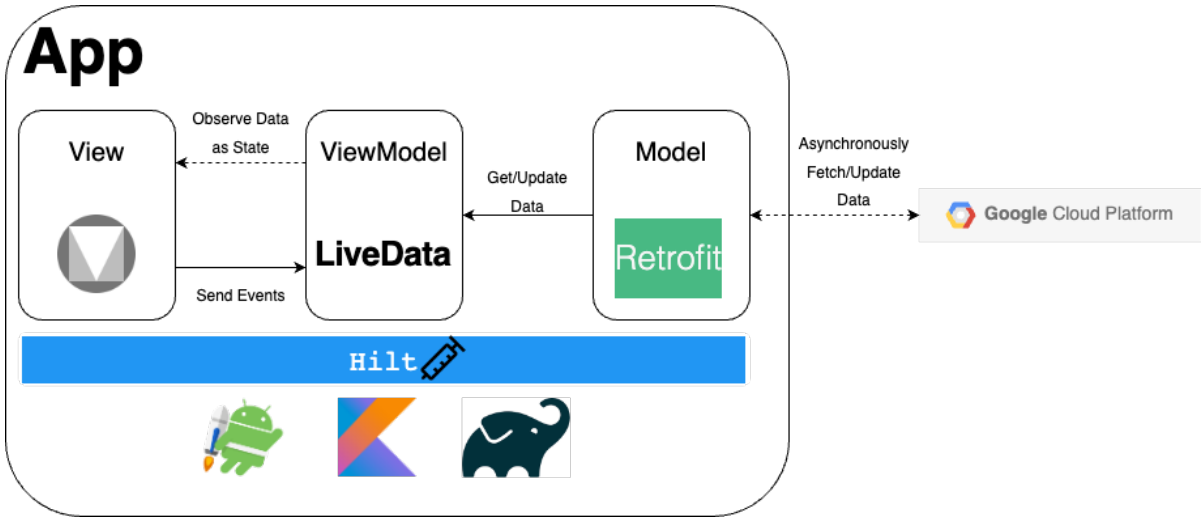
The application will be built with a fairly simple Client-Server architecture for providing dynamic user functionality. The Client will be an Android application backed by a serverless backend on Google Cloud. Further details are given below.

Subsystems

Android Client

The Android Client will be written in Kotlin using standard Android Jetpack libraries and follow the recommended Model-View-ViewModel architecture for Android applications, with a particular focus on correct separation of concerns with the help of Dependency Injection through Hilt and unidirectional data flow through observable state with LiveData.

External libraries will be managed through Gradle. Network requests will utilize HTTPS through the Retrofit library. The user interface design will be guided by Material Design with relevant components and animations.

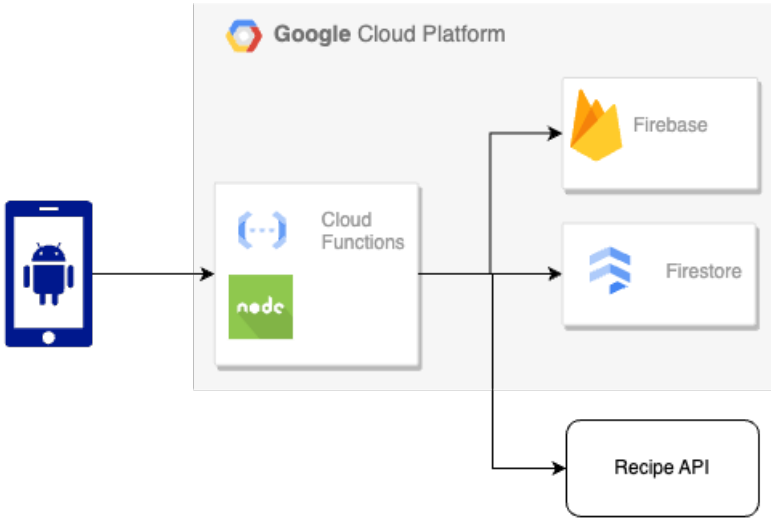


Backend

The data for the application will be delivered from three entities:

- Recipe API: the primary source of recipe data gathered from a public API
- Firebase Authentication: for user sign-up, login, and password management
- Firebase Firestore: a serverless NoSQL database that will manage user-specific data (saved recipes, favorites, shared recipes)

The data will be exposed to the client through a serverless backend deployed via Firebase Cloud Functions written in Node.js.



Challenges

This is my first time using the Google Cloud suite of products. I do have experience in cloud platforms (AWS) but I'm sure there will be some growing pains getting everything hooked up on the new platform.

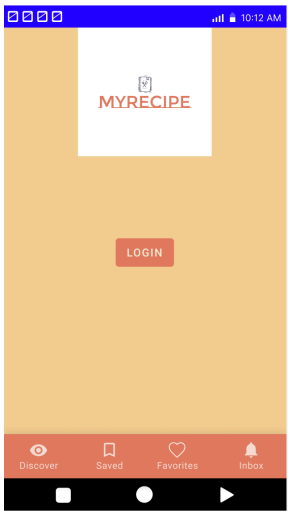
I want to enable some offline functionality, and will be saving the user's favorite recipes to the local device, a pattern I haven't worked with before.

The discovery feature of the application will utilize Swiping gestures and UI animations, which will require Android libraries I haven't used before.

USER INTERFACE

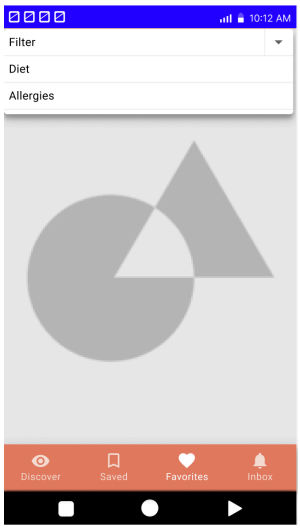
Login

Shown when the user tries to access online-features (Discover, Saved, Inbox) and is not currently logged in.



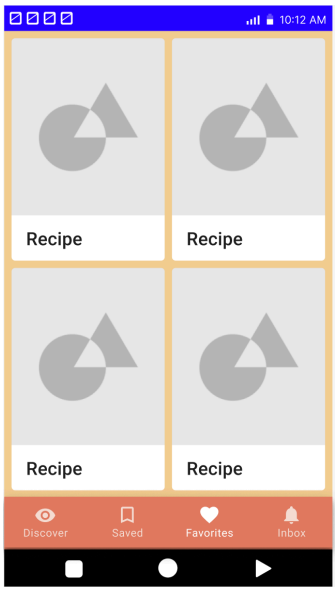
Discover

Displays images of dishes for the user to swipe through, filtering by Diet and Allergy if needed.



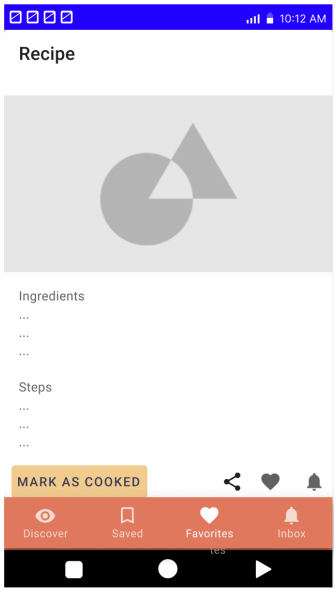
Saved

Displays images of dishes the user has saved, with option to sort by date added or whether the recipe has been cooked already.



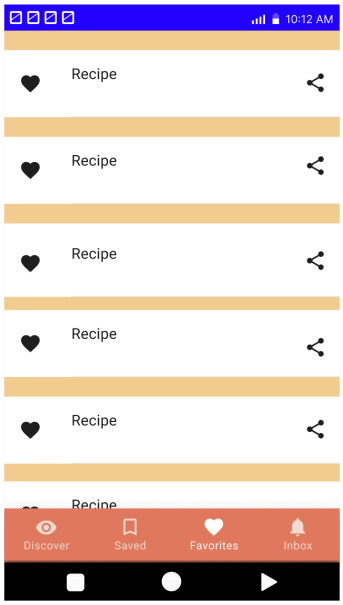
View Recipe

Displays full recipe (ingredients and steps) for a dish, activated when a user clicks on a Saved or Favorited recipe. User can mark recipe as cooked or as a favorite, or share with other uses in-app or out-of-app.



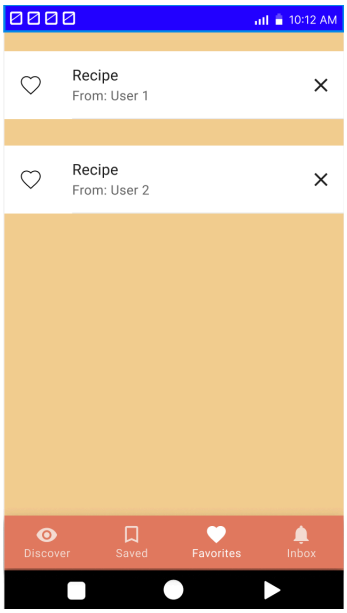
Favorites

Displays user's favorite recipes. User can remove recipes from favorites or share with others.



Inbox

Recipes sent from other users. User can open the recipe, favorite, or remove from the inbox.



SCHEDULE

Week	Focus	Deliverables
10/24 - 10/30	Setup Backend	User management and authentication setup in Firebase Firestore instance setup for User data Google Cloud Functions setup to expose backend
10/31 - 11/6	UI Layout Recipe Discover Functionality	Basic Layouts created with Material components, PoC for animations Client connected with Backend Discover functionality implemented with Swiping and Animations, filtering by Diet or Allergies
11/7 - 11/13	Recipe View Functionality Recipe Management Functionality	Fetch and view all matched recipes, sorted by date added or cooked status View individual Recipes Mark recipes as cooked Add recipes to Favorites View Favorited recipes
11/13 - 11/20	Offline mode Recipe inbox and sharing in-app Functionality Recipe sharing out-of-app Functionality	Save favorite recipes in Local data, able to load even without internet connection Send recipes to other users in-app Share recipe out-of-app
11/21 - 11/27	E2E Testing Usability Testing	Enhance test suite to work through complex user scenarios Polish Layout and Animations for enhanced usability
11/28-12/2	Rollover from previous weeks Project report	Complete any pending items from above Complete final Project report

APIS

Client

Android GestureDetector: https://developer.android.com/develop/ui/views/touch-and-input/gestures/detector#detect

Android Animations: https://developer.android.com/develop/ui/views/animations/overview

Android Local Persistence: https://developer.android.com/training/data-storage/app-specific

Android Share: https://developer.android.com/training/sharing/send

Firebase: https://firebase.google.com/docs/android/setup

Dependency Injection with Hilt: https://developer.android.com/training/dependency-injection/hilt-android

Material Design: https://material.io/components?platform=android

Backend

Recipe API: https://spoonacular.com/food-api

Firebase Cloud Functions: https://firebase.google.com/docs/functions/callable?authuser=0#call_the_function

Firebase Authentication: https://firebase.google.com/docs/auth/android/firebaseui

Firebase Firestore: https://firebase.google.com/docs/firestore/query-data/get-data

FUNCTIONAL SPECIFICATION

Login

A new User can sign-up with a user name and password

An existing User can login with their user name and password

This screen is displayed when the user is not logged in and tries to access online features: Discovery, Saved Recipes, Inbox

NavBar

Consists of 4 tabs: Discovery, Saved, Favorites, and Inbox

Corresponding Screen is displayed when tab is clicked

Discovery

Displays a single image of a dish

Fetched dishes are filtered according to Diet or Allergies as specified by the user

Swiping right on image causes Recipe to be Saved and image of next dish to appear

Swiping left causes image of next dish to appear

Saved Recipes

Saved Recipes are displayed with their Images and Names

User can sort Saved Recipes by date added or whether they have already cooked the recipe

User can click on a Recipe which opens the View Recipe Screen

View Recipe

Displays an image of the dish

Displays ingredients and steps needed for Recipe

User can mark the Recipe as cooked

User can mark the Recipe as a Favorite

User can Share the Recipe with other app users or through standard Android messaging out-of-app

Favorite Recipes

Recipes displayed as list with image and name

User can remove a Recipe from Favorites

User can Share the Recipe with other app users or through standard Android messaging out-of-app

User can click on a Recipe which opens the View Recipe Screen

This screen is displayed even if the user does not have internet connectivity

Inbox

Displays Recipes shared with user by other users

User can mark Recipe as a Favorite

User can remove a Recipe from the Inbox

User can click on a Recipe which opens the View Recipe Screen

INSPIRATION

My primary inspiration was my own passion for cooking new things and trying new foods. I spend a good amount of time meal planning and hunting down new recipes, which I manage through bookmarks in a web browser or links in a notes app. Creating a centralized place to organize my cooking activities seemed like a good use case.

In terms of UI, design is not one of my strong suits, but I have had good experience creating fairly attractive (for a backend developer) and snappy web applications using Material Design Components so it also seemed like an easy choice to follow in this project.

ACKNOWLEDGEMENTS

Logo: app.logo.com

Diagrams: app.diagrams.net

UI Mockups: www.figma.com

Color Scheme: coolors.co/