

Aprikot

Recipe Application

Udacity Kotlin Android Developer NanoDegree
Michael May

Description:

Aprikot is a food recipe app which uses data from TheMealDB (<https://www.themealdb.com/>)
This is a free to use API that does **not** require an api key.

When starting the app you will see a categories screen that represents various meal categories. There are three fragments attached to the MainActivity all connected through Navigation Manager and Bottom Navigation bar. Along with the categories screen you can navigate to either the Favorites or Settings Screen.

Categories Screen

- Select a category and you are taken to a new activity where a list of recipes is displayed.
- Scroll through the list, find and tap on your desired recipe. You are then taken to the Preparation Screen.

Preparation Screen

- The Preparation Screen has the ingredients, instructions and image of the meal to help you prepare this meal for yourself.
- The ingredients and instructions are scrollable.
- There is a Heart Icon in the top right section of the screen. Tap the heart icon and the recipe is added to your Favorites Screen. Tap it again and the recipe is removed from the Favorites.
- Pressing the Back Arrow on the upper left takes you back to the Categories Screen
- The Back Button will take you back to the Recipe List Screen.

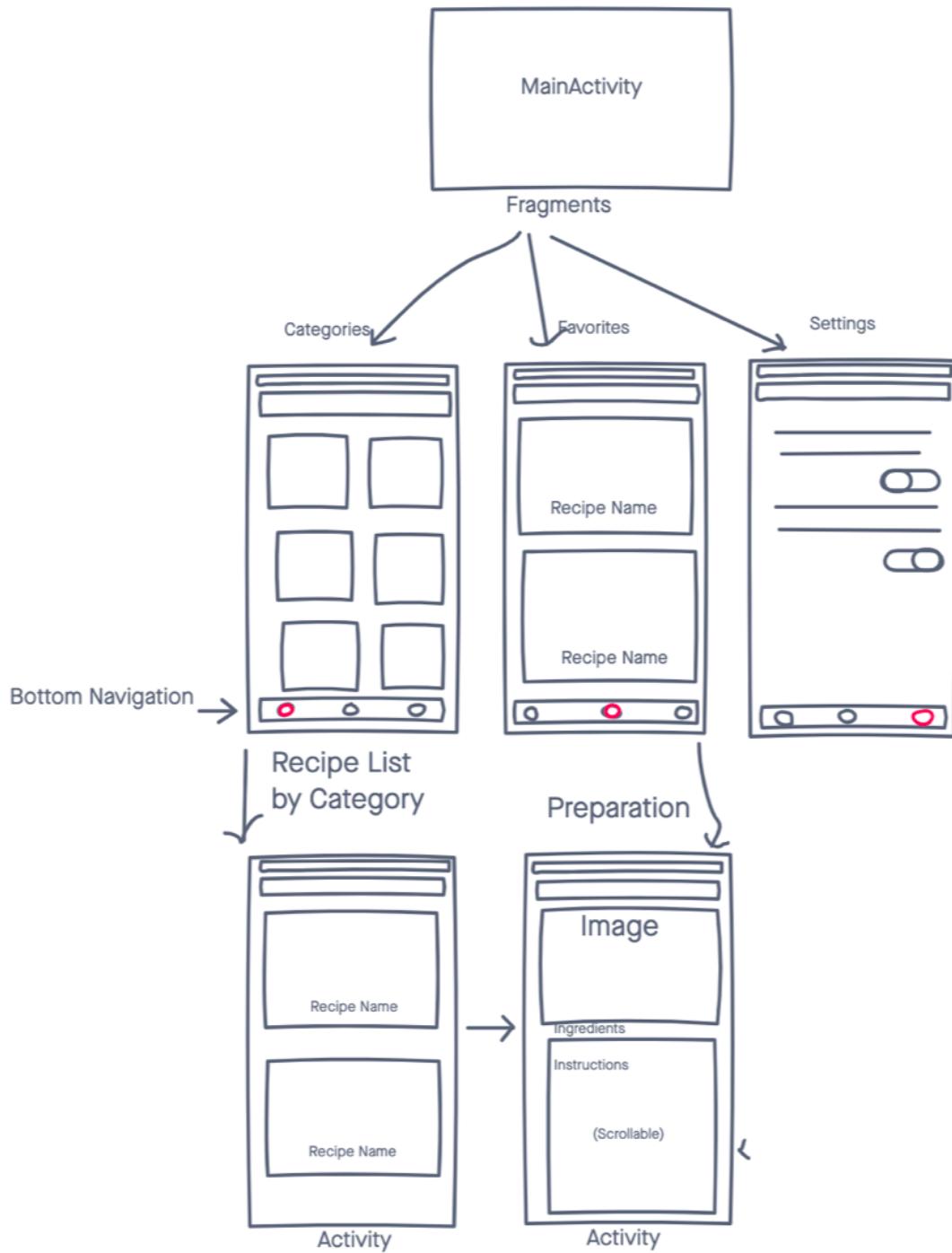
Favorites Screen

- The Favorites Screen shows any recipe that you have marked as favorites in the Preparation Screen. If there are no favorites a dialog will show indicating that.
- Tap on a recipe and you will be taken to the Preparation Screen to view the details of the recipe.

Settings Screen

- At this time the settings screen only has one setting available.
- In the Display Section there is a toggle switch to activate Wake Lock which will keep the Preparation screen awake so the user can keep the recipe details visible without the device sleeping. This is optional and is off by default. If left on, a notification will inform you that this setting is on.

WireFrames



Project Requirements

UI/UX

- Build a navigable interface consisting of multiple screens of functionality and data.
 - Aprikot has 5 Screens
 - Navigation Controller Used for Fragment Navigation via Bottom Navbar
 - Navigation to Recipe list and Preparation Screen are handled through intents
 - Aprikot uses DataClasses, Bundles and Parcelize to pass data.
- Construct interfaces that adhere to Android standards and display appropriately on screens of different size and resolution.
 - This application displays on different size screens and orientations.
 - Constraint layouts are used in every screen.
 - Resources are stored appropriately. Strings, Drawables etc.
 - Data collections are loaded through MVVM architecture from a local database.
- Animate UI components to better utilize screen real estate and create engaging content.
 - Motion layout is used in the Preparations Screen to make the content more engaging in landscape and portrait orientations.

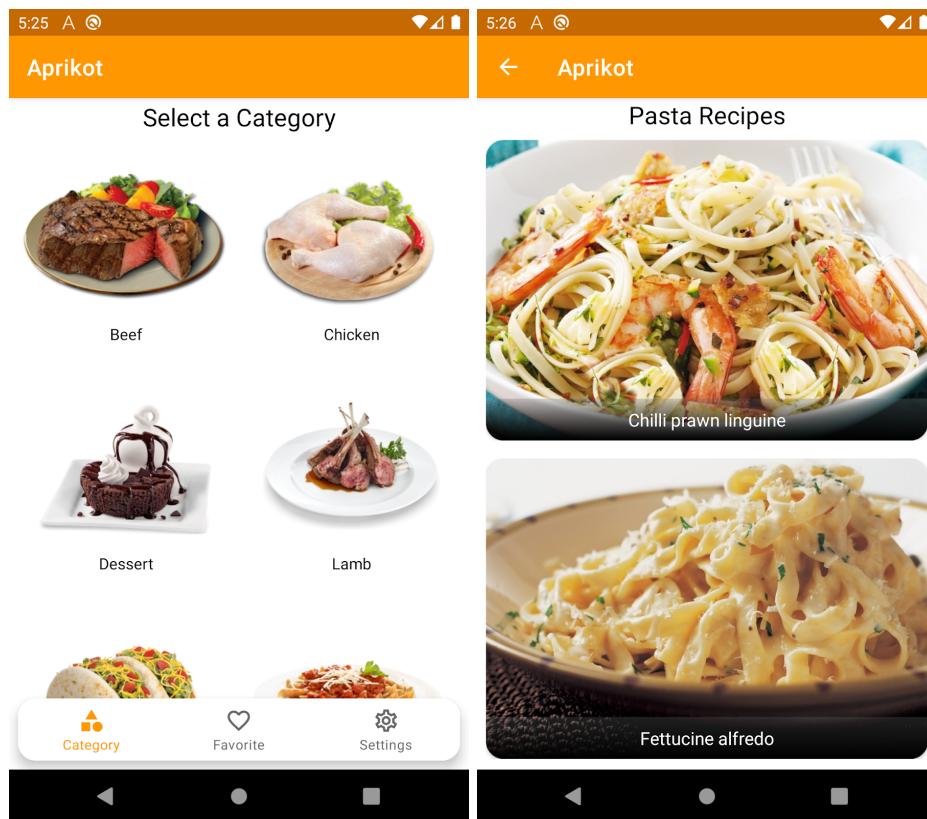
Local and Network Data

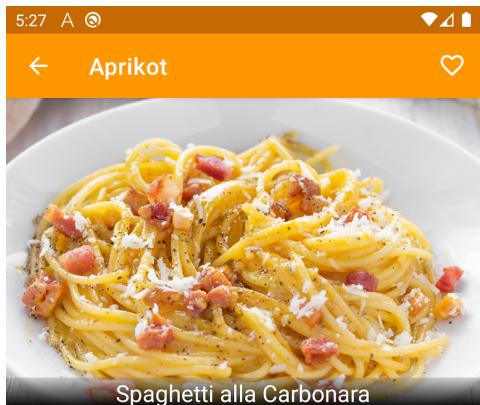
- Connect to and consume data from a remote data source such as a RESTful API.
 - Aprikot uses Retrofit to make API requests from TheMealDB.com then use Gson converter and a network utility to insert the data into a local database.
 - Coroutines/ Dispatchers are used for threading.
- Load network resources, such as Bitmap Images, dynamically and on-demand.
 - Picasso Library was used to load images from the web.
- Store data locally on the device for use between application sessions and/or offline use.
 - All data downloaded from the api is stored in the Room Database locally.
 - Downloaded data is checked. If it is in the database then an api request is not made.
 - Data is stored appropriately using threading to maintain performance and offline usage.

Android system and hardware integration

- Architect application functionality using MVVM.
 - Aprikot uses the MVVM design pattern
 - Uses models, fragments, activities and viewmodels
- Implement logic to handle and respond to hardware and system events that impact the Android Lifecycle.
 - Handles lifecycle events in respect to orientation changes and notifications.
- Utilize system hardware to provide the user with advanced functionality and features.
 - Permissions handled in the manifest. No run-time permissions were needed.
 - Permissions include: Internet, WakeLock, Access Network State
 - The app uses notifications as suggested by the Rubic to meet the hardware requirement.

Screen Shots





Ingredients

320g Spaghetti
6 Egg Yolks
As required Salt
150g Bacon
50g Pecorino
As required Black Pepper

Instructions

STEP 1

Put a large saucepan of water on to boil.

STEP 2

Finally chop the 100g pancetta, having first

