CPSC 464 Project Report
EasyRecipes

Table Of Contents

Description

EasyRecipes is a mobile application project developed in Kotlin by team Retrofit, utilizing the Android Studios IDE. This project is designed specifically for CPSC 411a-03/04 spring 2024 course at California State University, Fullerton. This app aims to provide a user-friendly storage platform for browsing and saving culinary recipes. *EasyRecipes* features a clean intuitive user interface that allows for easy navigation.

Github Repository

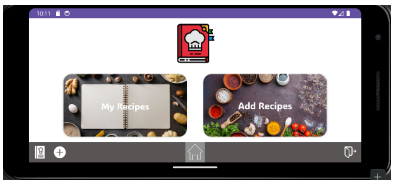
https://github.com/FMZOrganization/final-project-team-retrofit

Tech Stack

Android Studios IDE
Kotlin (Target Version 1.8)
Retrofit 2.0
Android SDK 25+ (main version 34)

CPSC 464 Project Report

Requirements

Requirement	Description	Examples
Callbacks	The user should be able to click at least one View that causes a callback function to be called, which produces an apparent result back to the user	The mainActivity.kt showcases majority of the callbacks
Logging	Important events should be logged to the Logcat area with varying levels and unique tags	Logging showcased in mainActivity viewable through the Logcat with any of the button presses.
Layouts	Your app must use the ConstraintLayout at least once with multiple internal Views. Elsewhere in your app, you must use another Layout container type, such as LinearLayout (horizontal or vertical), or TableLayout	Refer to our xml files
Resources	There should be at least two image resources utilized in some way by the application, in an interactive way	 Interactable buttons
Resources (extra credit)	Earn extra credit by having your app embed a playable video or audio file. You may implement this using resources OR asset file streams.	A success sound plays when the user clicks the “save” button
Resource qualifiers	The app should make use of resource qualifiers to dynamically choose alternative resources based on	The app supports translations for German Spanish Japanese(Kanji & Hiragana)

CPSC 464 Project Report

	device configuration. There should be at least three dynamically chosen resources. The first dynamic resource should be the strings file that allows for multiple translations of the app; The app should support at least 3 translations. The second dynamic resource should be a layout file that shows the user a different layout based on orientation (or alternatively the screen size). The third (and beyond, if desired) resources may be anything of your choosing	The app also supports rotations
Persistence	The app should allow the user to generate data in some way. The data should survive device rotations by utilizing a ViewModel. The app should also have the ability to save files. At least one file should be saved to the device via user interaction, survive across app exit, and loaded+viewable on the next launch. The app should save at least two user preferences by utilizing the DataStore package (or alternatively, the FireStore package). Preferences should persist across app exit. App allows user to enter data that can be saved to a local SQLite database and fetched later. User is able to save/fetch multiple entries for the same	Check through rotating phone/tablet device

CPSC 464 Project Report

	type of data (i.e., multiple rows in a database table). This requirement may be tweaked slightly the week before presentations, if needed	
User Interface	The app should contain at least two fragments. Each fragment should be used in at at least two places in the app. At least one layout should contain multiple fragments. The use interface must contain at least one List with RecyclerView. The user interface should be user-friendly, functional, optimal and polished on devices of various sizes and in both orientation (portrait/landscape).	The recycler view mutable list is displayed on the RecipeListfragment
RESTful Interactivity	The app should communicate with a RESTful API server using Retrofit. The app should pull some sort of data from the remote API using an HTTP GET request and render it somewhere in the local app. The app should also allow the local user to provide some sort of data that will be sent to the remote API using an HTTP POST request. When the user exits and relaunches the app, data previously sent to the remote API should be fetched from the remote API and	HTTP requests and endpoint validation can be viewed through the Logcat

	shown to the user. The previously sent data should not be saved using local persistence, but fetched from the live server instead, to prove the app is “cloudified” in some way. You may choose to use an existing API or create one yourself, as long as your app is able to both create data on the remote server and fetch those records.	
SQL Database	The database should have endpoints, save and fetch data	

Compatibility

EasyRecipes has been tested with the listed devices below and their API versions:
Pixel 3a API 34
Pixel Tablet API 30

CPSC 464 Project Report

UI/UX

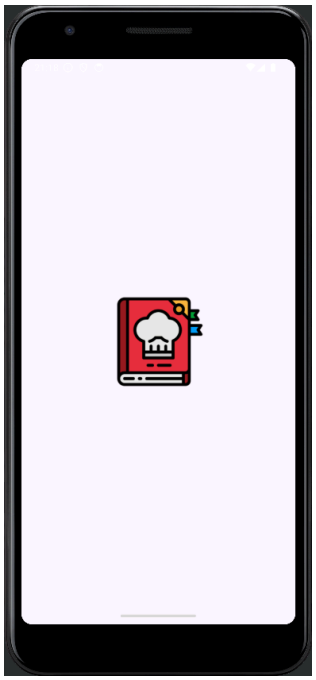


Figure 5.1 Loading Screen (Pixel 3a)

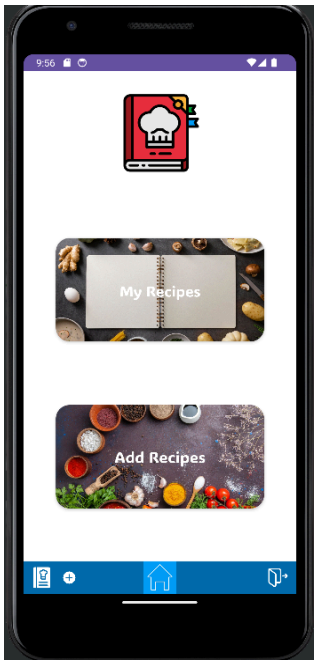


Figure 5.2 Landing Page Home Screen (Pixel 3a)

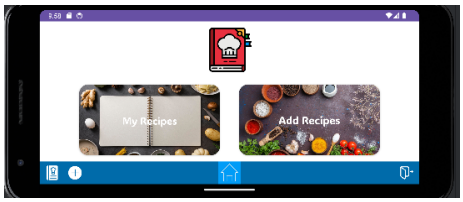


Figure 5.3 Landscape View (Pixel 3a)

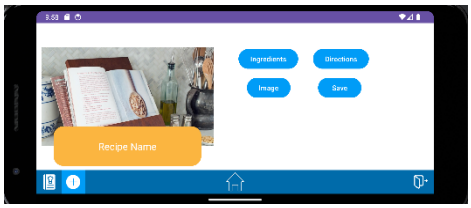


Figure 5.4 AddRecipe Screen (Pixel 3a)

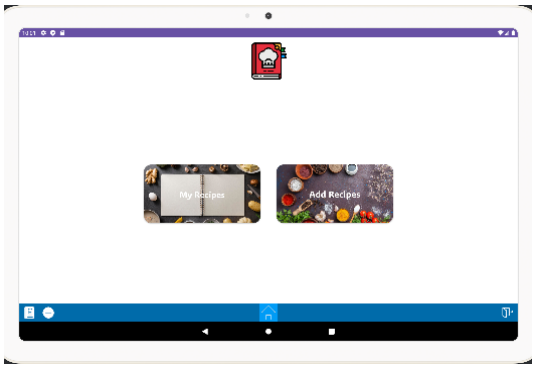


Figure 5.5 Tablet Landing Page (Pixel Tablet)