GitHub Username: tariqywsf

BakingApp

Description

If you are about to make your own cake and eat it, try this app to learn step by step how to make your favorite cake or any baking of your choosing to learn to make it in an easy and friendly way , by video and instructions for each step after you prepare the needed ingredients.

Intended User

- This app is for cooking enthusiasts , housewives , or anyone who likes to try cooking for the first time, or for a hobby.
- This is an app that will allow a user to select a recipe and see video-guided steps for how to complete it.

Features

- Save favorite Recipe for offline use .
- Widget to keep ingredients on homescreen .
- Play Videos of steps with instructions of each step .

User Interface Mocks

Screen 1: Select a Recipe - Portrait (Phone)
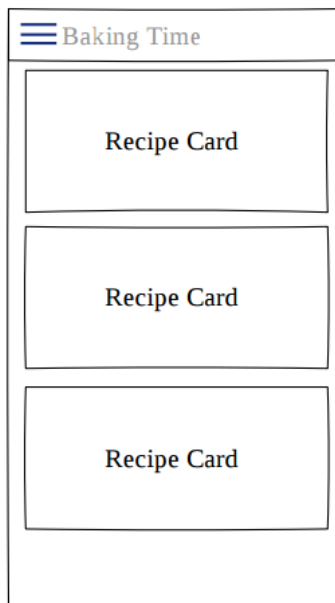
Screen 2: Select Recipe Detail View - Portrait (Phone)

☰ Recipe Name
Recipe Ingredients
Recipe Step Description
Recipe Step Description
Recipe Step Description
Recipe Step Description
Recipe Step Description
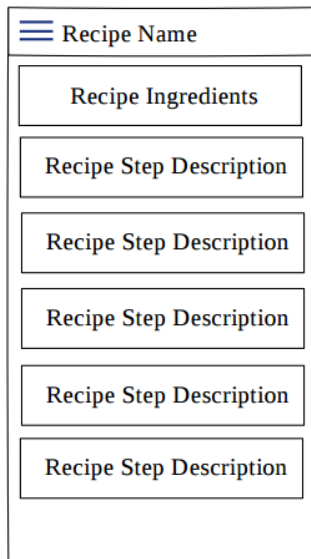
Screen 3: Select Recipe Step Detail View - Portrait (Phone)

☰ Recipe Name
Media Player
Recipe step instruction
Navigation to next or previous step?
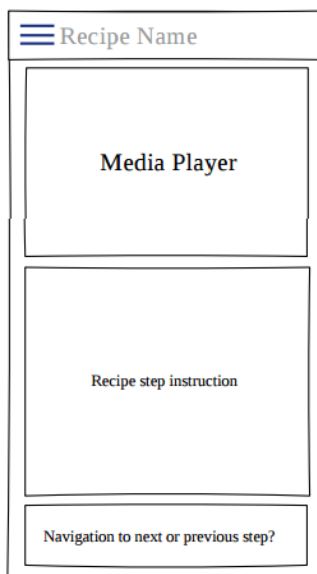
Screen 4: Select Recipe Step Detail View - Landscape (Phone)

Video takes up full screen
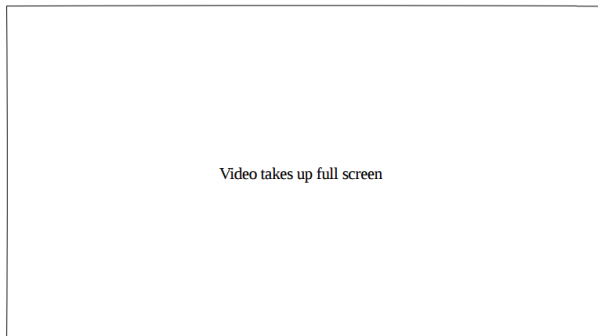
Screen 5: Select a Recipe (Tablet)

☰ Baking App

Recipe Card	Recipe Card	Recipe Card
Recipe Card	Recipe Card	Recipe Card
Recipe Card	Recipe Card	Recipe Card

Screen 6: Select and View Recipe Step Detail - (Tablet)

☰ Baking App

Recipe Ingredients

Recipe Step Description

Recipe Step Description

Recipe Step Description

Recipe Step Description

Recipe Step Description

Recipe Step Description

Video

Recipe Step Instructions
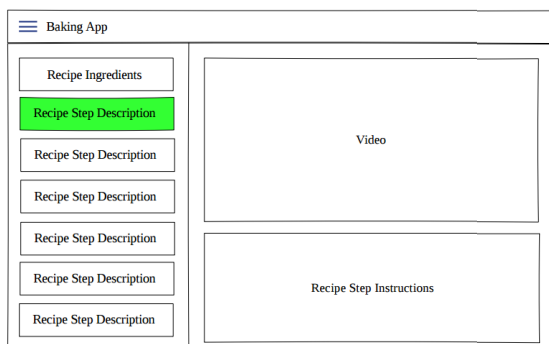
Screen 7: The App widget

Recipe Name

Ingredient Quantity	Measurement Unit	Ingredient
Ingredient Quantity	Measurement Unit	Ingredient
Ingredient Quantity	Measurement Unit	Ingredient
Ingredient Quantity	Measurement Unit	Ingredient

Key Considerations

- App is written solely in the Java Programming Language
- App utilizes stable release versions of all libraries, Gradle, and Android Studio.
- App integrates a third-party library.
- App validates all input from servers and users. If data does not exist or is in the wrong format, the app logs this fact and does not crash.
- App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.
- App keeps all strings in a "strings.xml" file and enables RTL layout switching on all layouts.
- App provides a widget to provide relevant information to the user on the home screen.

How will your app handle data persistence?

Data will be saved by Firebase Realtime Database.

Describe any edge or corner cases in the UX.

On rotating the phone to landscape the video(in details Fragment) resume from the same time before rotating.

Describe any libraries you'll be using and share your reasoning for including them.

Timber Library for Logging for easy logging and to remove the logs easily in production.
Dagger Library for Dependency Injection , to remove boilerplate code and for better testing.
Glide Library , for better scaling Images and placeholders.
Retrofit Library together with RxJava , for better networking handling.
Exoplayer Library for Video player.
Firebase Realtime Database Library for Data persistence.
Firebase Analytics
Firebase Crashylitics

Describe how you will implement Google Play Services or other external services.

Firebase Realtime Database for Data persistence.
Firebase Analytics
Firebase Crashylitics

6

Next Steps: Required Tasks

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:
- Configure libraries found here
- Configure screens and widgets used
- Configure color palettes and fonts following the Material Guidelines

Task 2: Implement UI for Each Activity and Fragment

- Build UI for activity_recipes_list.xml
- Build UI for list_recipes.xml
- Build Ui for item_recipe.xml
- Build Ui for item_ingredient.xml
- Build Ui for activity_recipe_detail.xml
- Build Ui for fragment_recipe_detail.xml
- Build Ui for fragment_step_detail.xml
- Build Ui for item_step.xml
- Build Ui for widget_configure.xml
- Build Ui for widget_recipes.xml
- Build Ui for widget_recipes_item.xml

Task 3: Handling Variations

For Landscape variant:
- Build Ui for fragment_step_detail.xml

For Tablet Variant:
- Build Ui for activity_recipe_detail.xml
- Build Ui for fragment_step_detail.xml
- Build Ui for List_recipes.xml

For the widget:
- Build Ui for recipes_widget_info.xml

Task 4: Enhancing the Code

- Implement Dagger
- Implement widget classes
- Implement ViewModels
- Implement MVVM Architecture with Repository and package classes properly
- Implement Exoplayer to display videos.

Task 5: Further Enhancements and handling Edge Cases

- Application uses Master Detail Flow to display recipe steps and navigation between them.
- Make sure Application properly initializes and releases video assets when appropriate.
- Make sure Application should properly retrieve media assets from the provided network links. It should properly handle network requests.
- Application makes use of Espresso to test aspects of the UI.
- App uses RecyclerView and can handle recipe steps that include videos or images.
- App Behaves responsibly when changing orientation
- App handles portrait and landscape during showing videos and other screens on both phone(handset) and Tablet gracefully