

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1: Select a Recipe - Portrait \(Phone\)](#)

[Screen 2: Select Recipe Detail View - Portrait \(Phone\)](#)

[Screen 3: Select Recipe Step Detail View - Portrait \(Phone\)](#)

[Screen 4: Select Recipe Step Detail View - Landscape \(Phone\)](#)

[Screen 5: Select a Recipe \(Tablet\)](#)

[Screen 6: Select and View Recipe Step Detail - \(Tablet\)](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Handling Variations](#)

[For Landscape variant:](#)

[For Tablet Variant:](#)

[For the widget:](#)

[Task 4: Enhancing the Code](#)

[Task 5: Further Enhancements and handling Edge Cases](#)

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

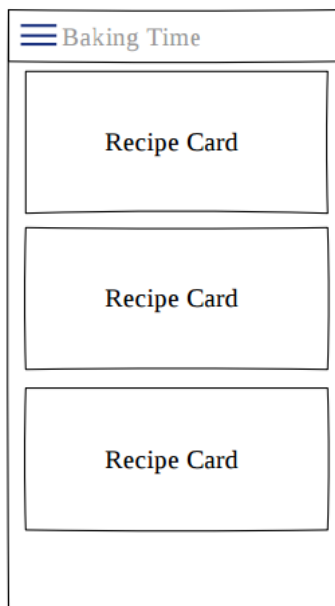
This is an app that will allow a user to select a recipe and see video-guided steps for how to complete it.

Features

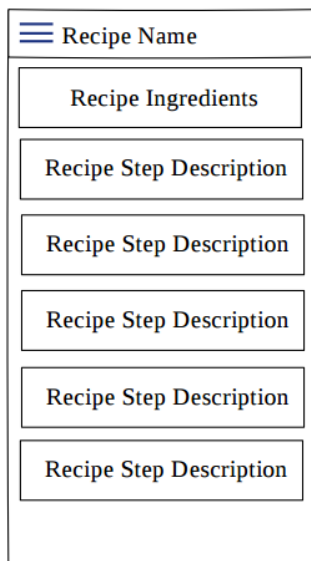
- Save favorite Recipe for offline use .
- Widget to keep ingredients on homescreen .
- Play Videos of steps with instructions of each step .

User Interface Mocks

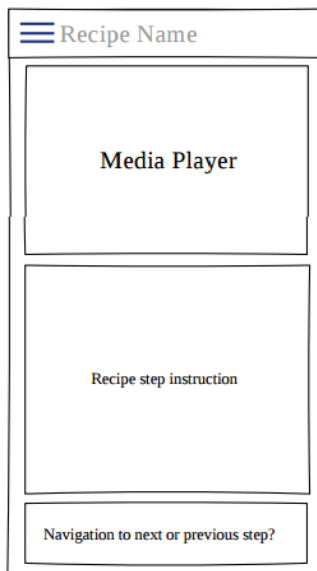
Screen 1: Select a Recipe - Portrait (Phone)



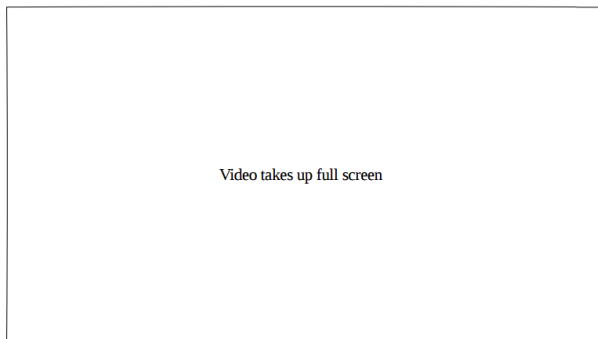
Screen 2: Select Recipe Detail View - Portrait (Phone)



Screen 3: Select Recipe Step Detail View - Portrait (Phone)



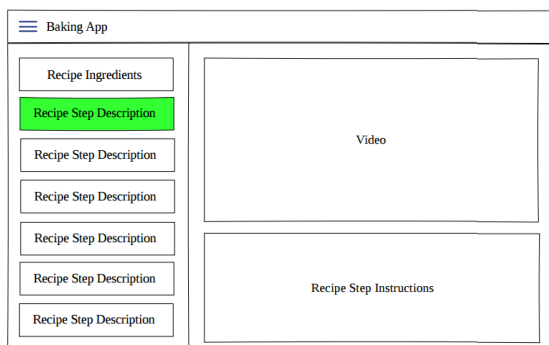
Screen 4: Select Recipe Step Detail View - Landscape (Phone)



Screen 5: Select a Recipe (Tablet)



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



Key Considerations

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 Crashlytics

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

Firebase Realtime Database for Data persistence.
Firebase Analytics
Firebase Crashlytics

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