GitHub Username: **GeorgeKorovesis**

RecipeMe

Description

This app provides list of best rated recipes based on user selection (e.g. chicken, beef). Also gives the opportunity for bookmarking favorite recipes, sharing with friends and uploading own recipes.

Intended User

This app is intended for everyone in search of recipes.

Features

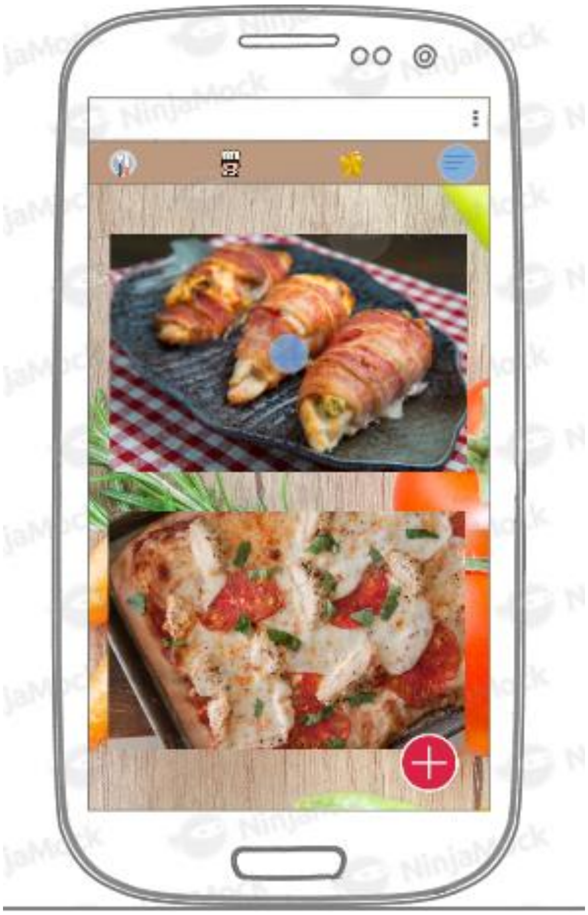
Main features of your app:

- Displays information
- Saves information
- Shares information

Languages used:
This App is written solely in JAVA.
This App refer all the hardcoded strings from the strings.xml file.
This app enables RTL layout switching to support accessibility on RTL supported languages.

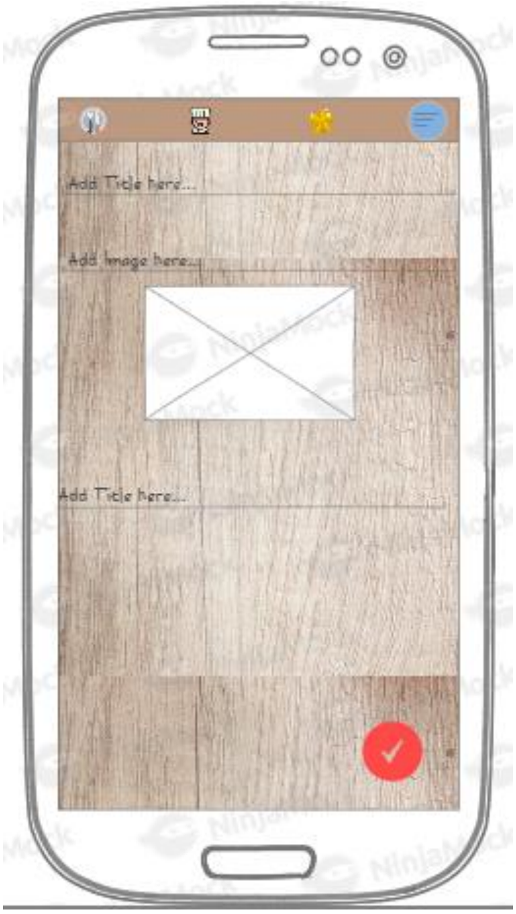
User Interface Mocks

Screen 1



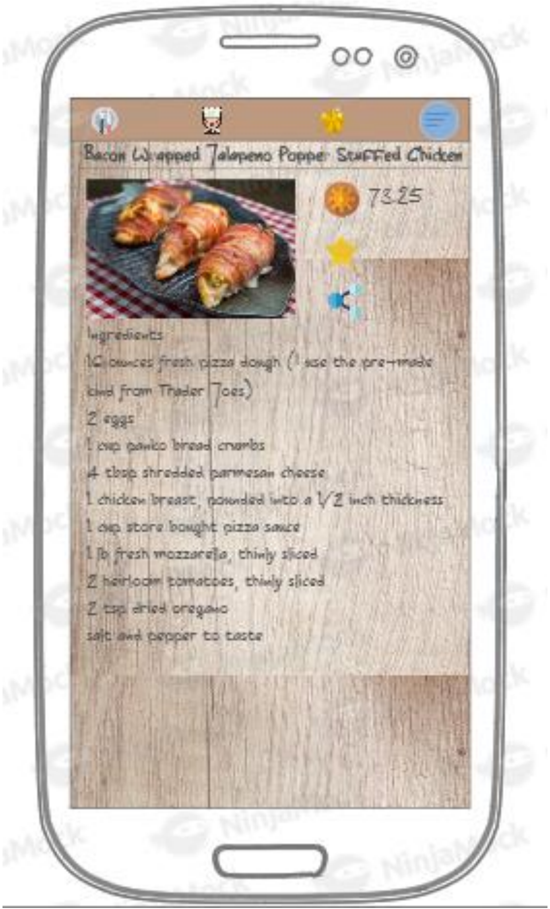
This is the initial screen, where a list of different recipes is presented to user (🍴). This screen is also almost identical to the second tab, with the only difference that in second screen recipes are selected base on time of day and rating (👨‍🍳).

Screen 2



This screen appears, where user adds a custom recipe, by pressing ⊕.There, user has to add a title, load an image and mention the ingredients of the recipe.

Screen 3



Bacon Wrapped Jalapeno Popper Stuffed Chicken

Ingredients

16 ounces fresh pizza dough (I use the pre-made kind from Trader Joes)

2 eggs

1 cup panko bread crumbs

4 tbsp shredded parmesan cheese

1 chicken breast, pounded into a 1/2 inch thickness

1 cup store bought pizza sauce

1 lb fresh mozzarella, thinly sliced

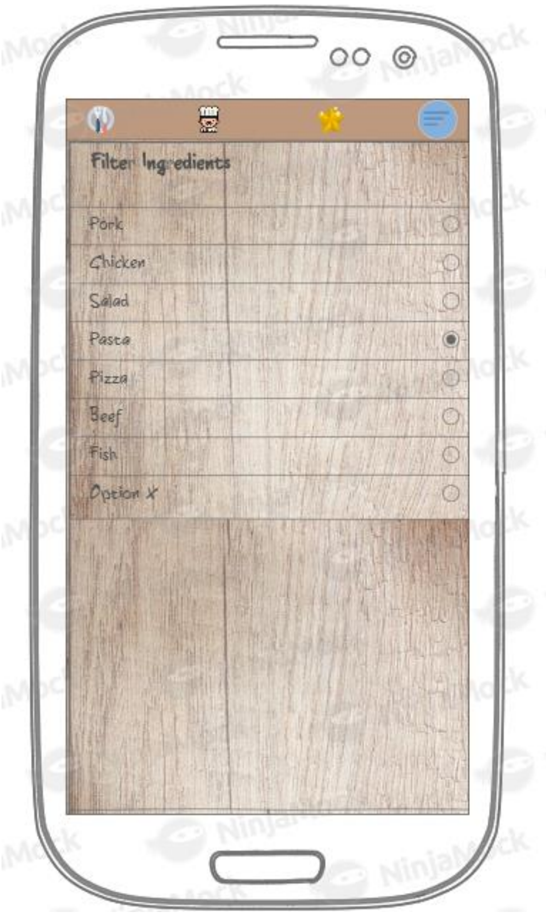
2 heirloom tomatoes, thinly sliced

2 tsp dried oregano

salt and pepper to taste

This screen appears every time a recipe is selected and contains information about recipe, and functions such as bookmark, share

Screen 4

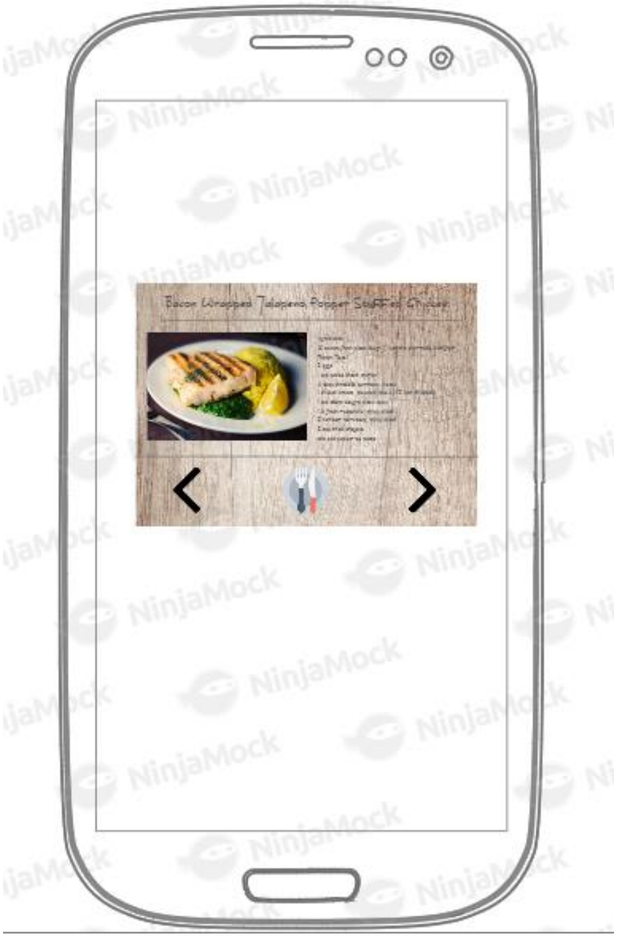


Filter Ingredients

- Pork ○
- Chicken ○
- Salad ○
- Pasta ●
- Pizza ○
- Beef ○
- Fish ○
- Option X ○

This image appears every time Option menu item is selected (marked in blue). It contains options to filter the recipes shown in initial screen (🍴)

Screen 5



This image is the widget of the app

Key Considerations

How will your app handle data persistence?

Room will be used for handling the local Database

SharedPreferences will be used for storing User Preferences.

Describe any edge or corner cases in the UX.

- Connection Lost: List will not be populated, Text with “no results found” will appear in background. Also a dialog will prompt user to enable internet connection (“Enable Internet Connection”). A scheduler will be scheduled to check every 1 minute (or earlier) for connection, in order to fetch data and populate list
- API issues: Same as with “Connection Lost”, but with different dialog message (snackbar) to usage (“Could not fetch data from endpoint”)

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

App utilizes stable release versions of all libraries, Gradle, and Android Studio.

SW Name	SW Version	Reasoning
Gradle	4.4	-
Android Gradle plugin	3.1	-
Android Studio	3.1.4	-
Picasso	2.71828	Image loading
Retrofit	2.4.0	API calls
Room	1.0.0	Local DB
Butterknife	8.8.1	Android resources
WorkManager	-	Scheduling works

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

- Crashlytics will be used for crash reporting.
- Firebase Cloud Messaging will be used in case notification for update is needed for users.
- Firebase Realtime Database will be used to store custom recipes from users.
- WorkManager will be used for pulling/sending data to/from API/remote db.
- Google Mobile Ads will be used for showing advertisements to users, since this app is free

Next Steps: Required Tasks

Task 1: Project Setup

- Update necessary gradle files (e.g. Android plugin, libraries, build tools).
- Gather necessary icons and images.

- Collect API calls needed and other stuff (e.g. keys, tokens).

Task 2: Implement Basic Activity and Fragments

There is one main Activity, with an action bar with 4 items, and 4 different Fragments

Task 3: Create necessary services to handle API requests

At least one service has to be implemented to take care of requests towards either Firebase Realtime DB, which is used for storing own recipes, or food2fork.

Task 4: Create Database

Database should contain info about existing recipes (from food2fork) and newly added recipes (from Firebase Realtime DB). Database should be created using Room.

Task 5: Create Rest API calls

At first, it is needed to check which API calls are needed for fetching all recipes, and recipes based on user preference (from food2fork). Additionally, calls must be also prepared for sending custom recipes to Firebase Realtime db.

Task 6: Create schedulers for updates

Schedulers have to be created to trigger updating the list of recipes. WorkManager should be used for that task.

Task 7: Handle Error Cases

Handle error cases (e.g. no connection at the beginning, connection lost in the meantime, wrong use of app by User)