GitHub Username: geetha10

KitchenManager

Description

KitchenManager is a food recipes app. This free recipe app provides a variety of delicious and healthy dishes from different regions of the world and it provides the feature to save all of your favorite recipes. Instead of using cooking books you can use this app to explore tasty dishes.

Apart from providing recipes it also provides:

1. Pantry Manager, which keeps track of the expiry date of items in the pantry and notifies the user

2. Shopping List, where users can note down the list of things to buy. Users can add collaborators for shopping lists.
3. Expense calculator

Intended User

Families

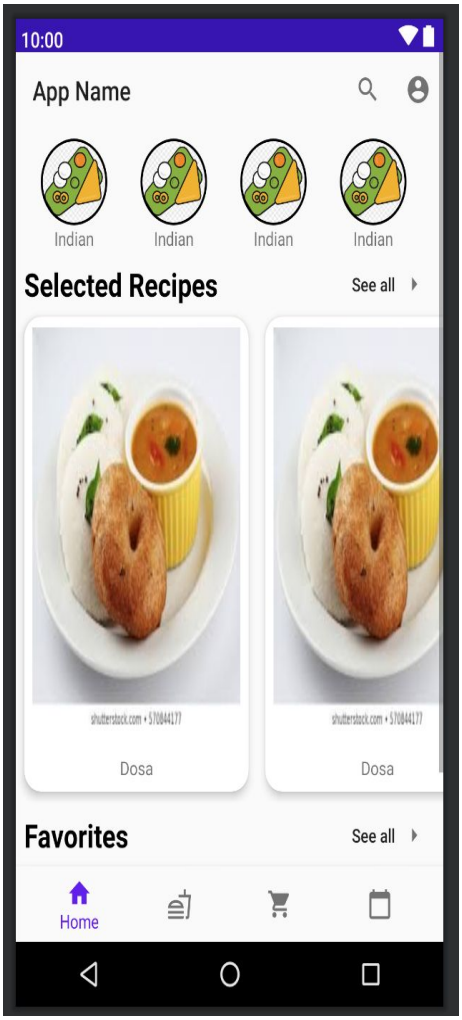
Features

The main features of app are:

- Provides Recipes
- Saves Favorite Recipes
- Shopping List
- Pantry Manager
- Expense Calculator

User Interface Mocks

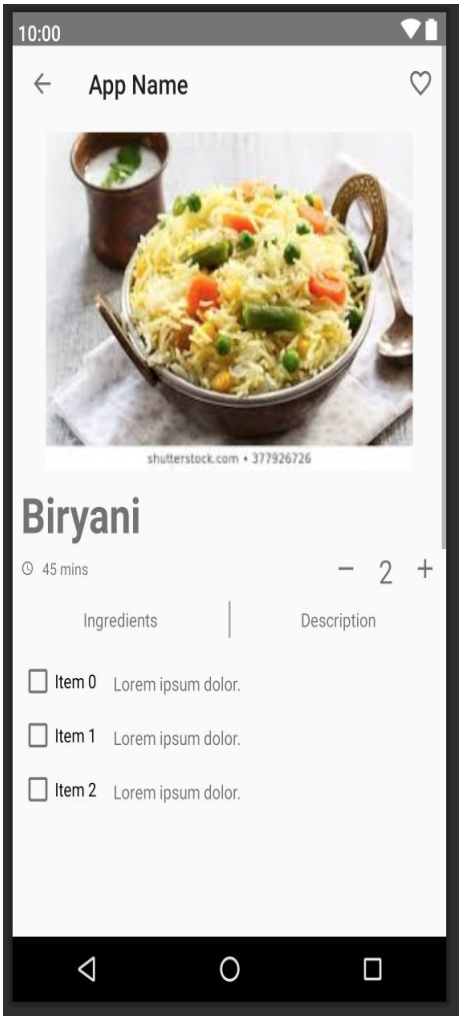
Screen 1



This is the welcome screen of the app.

- It contains different cuisines displayed on top of the screen.
- User preferred recipes are displayed in the select recipes section.
- User favorites are displayed in the favorites section

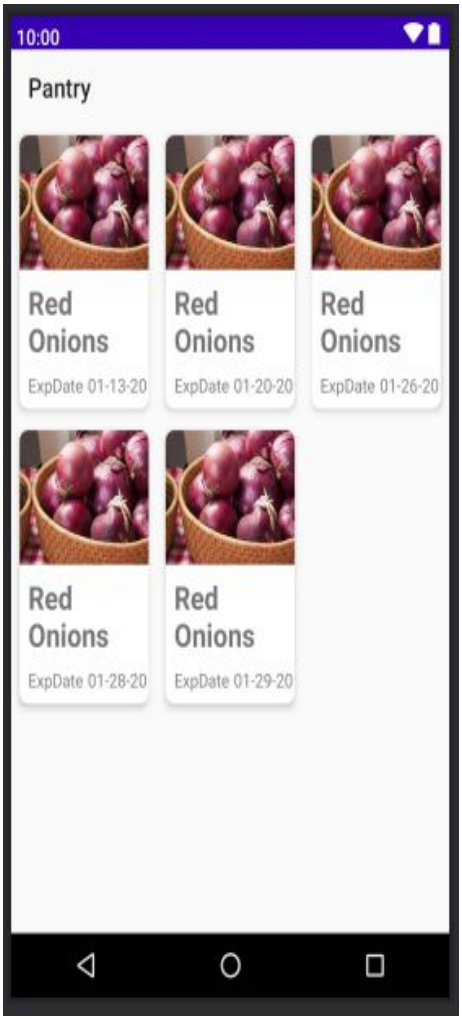
Screen 2



User will navigate to this screen on clicking the recipe. This recipe detail screen will display the

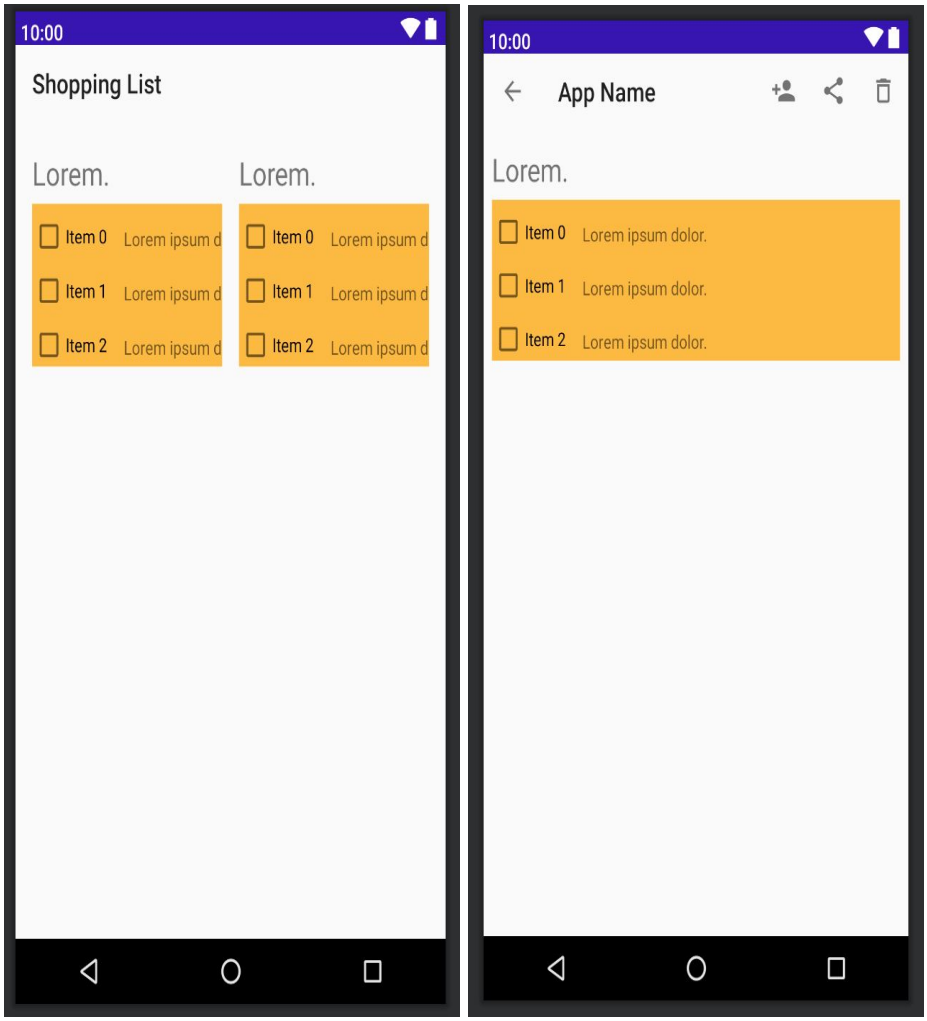
- Image of the recipe
- Preparation time
- Portion/how many persons this recipe serves
- Ingredients required for the recipe
- Description Steps

Screen 3



User will navigate to this screen by clicking the pantry button in the Bottom Navigation bar of the home screen. This screen shows the items which are available in the pantry and its Expiry Date.

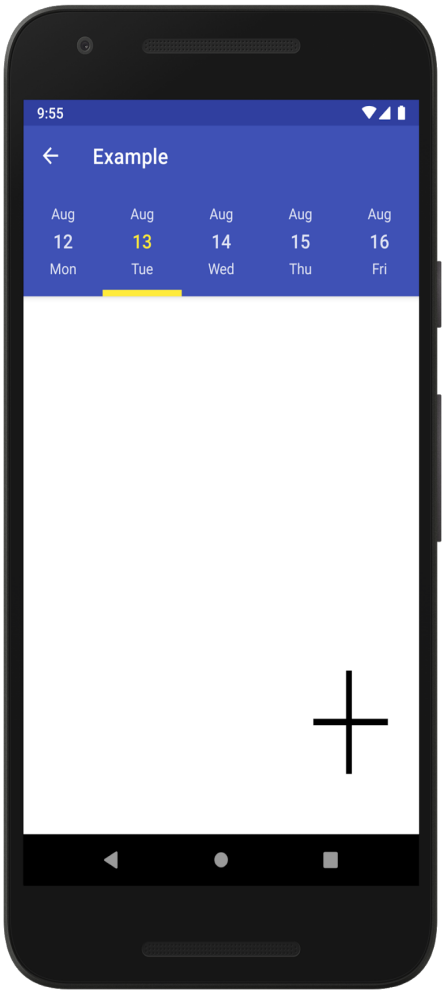
Screen 4



User will navigate to this screen by clicking the shopping cart button in the Bottom Navigation bar of the home screen.This screen displays the list of all saved shopping lists. On clicking the list user will get the following options.

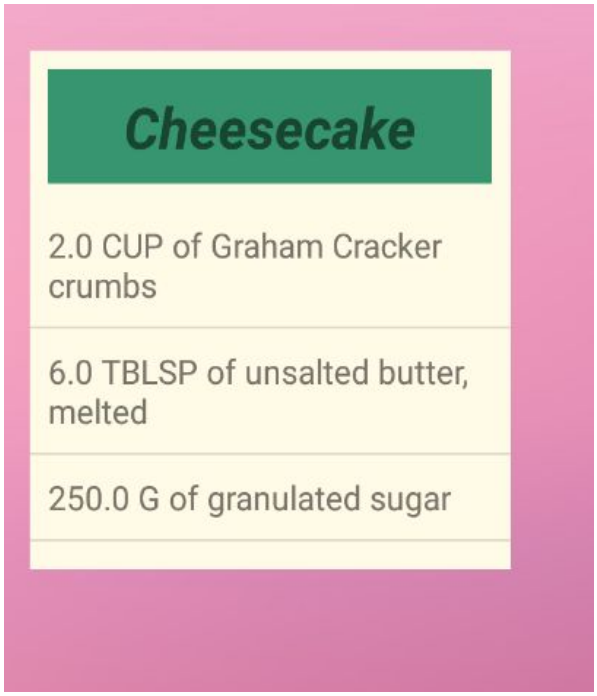
- Add/update the items
- Share shopping list
- Add collaborator
- Delete the list

Screen 5



Users will navigate to this screen by clicking the Calendar button in the Bottom Navigation bar of the home screen. Users can log the meals/expense by clicking the add button.

Screen 6



App will have a widget which shows the ingredients of the selected recipe.

Key Considerations

- **App will use solely Java for development purposes.**
- **App should use stable versions of all libraries.**

How will your app handle data persistence?

Describe how your app with handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)

- Shared Preference for saving user preferences.
- Room Database.

Describe any edge or corner cases in the UX.

I am handling situations where a user loses network connectivity by making my app ‘offline first’. All the content is cached locally for accessing in case of no network. All the network dependent operations will be queued and automatically resumed or retried on network availability.

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

- Glide(V4.11.0) to handle the loading and caching of images.
- Firebase Auth(V6.2.0) for user authentication.
- hdodenhof/CircleImageView(V3.1.0) for showing circular imageview.
- kizitonwose/CalendarView(V0.3.5) for calendar.
- Retrofit(V2.1.0) for network calls.
- WorkManager for scheduling jobs.

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

FireBase Authentication for user authentication.
FireBase Database for collaborating shopping list.

Next Steps: Required Tasks

Task 1: Project Setup

You may want to list the subtasks. For example:

- Configure libraries
- Create layout files for fragments

Task 2:Integrating recipes API

- Get the API key
- Define API interface

Task 3:Define Database for app

Room Database

- Create entities
- Define DAO classes

Task 4: Implement UI for Each Activity and Fragment

- Build UI for MainActivity

- Build UI for Home Fragment
- Build UI for RecipeDetail Fragment
- Build UI for Pantry Fragment
- Build UI for Calendar Fragment
- Build UI for Shopping Cart Fragment

Task 5: Implementing the logic

- Make the network call and save data in database
- Implement Home fragment
- Implement RecipeDetail fragment
- Implement Pantry fragment
- Implement Calendar fragment
- Implement ShoppingCart fragment
- Implement User Authentication
- Implement OnBoarding

More Technical Specifications

- Communication between app and backend will be handled on a background thread, and making sure UI is fluidly clean.
- All resources will be handled as dictated by Android design guidelines. For example, all strings will be handled in strings.xml and nowhere inside the layouts or code.
- Coming to the design and architecture, all layouts will be using Fragments inside them for ease with Tablet screens as well. Communication between Fragments on same Activity will be handled by the Activity enclosing the Fragments.
- App updates recipes from API(Spoonacular) in database(Room) at regular intervals using a JobDispacter.