GitHub Username: tarunvsaini

RECIPE DIARY

Description

Recipe Diary is a cooking app that provides a collection of recipes.It provides users an option to upload their own recipes and share them with other users.
Users can save their favorite recipes for offline reading.
In case ingredients required are not available at users home they can add ingredients to their shopping list.
The shopping list is available through a home screen widget.

Intended User

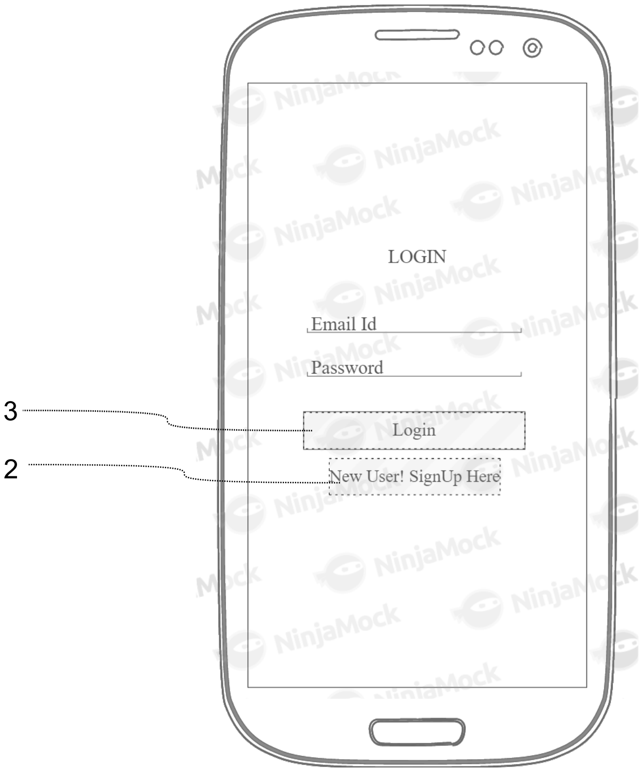
Recipe Diary is for anyone who likes cook and share their creations with others.

Features

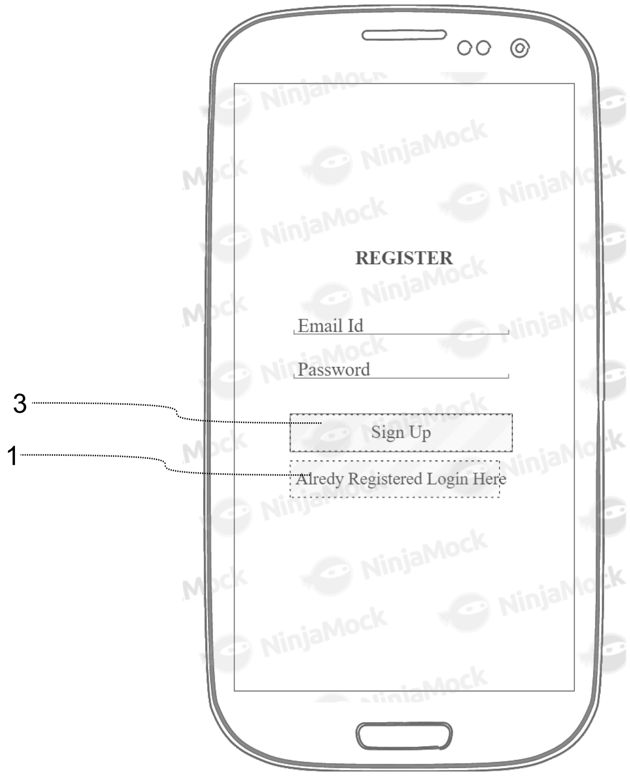
- Access to variety of Recipes.
- Option to upload recipes.
- Save favorite recipes for offline access.
- Create shopping list from ingredient list.
- Access shopping list through home screen widget

User Interface Mocks

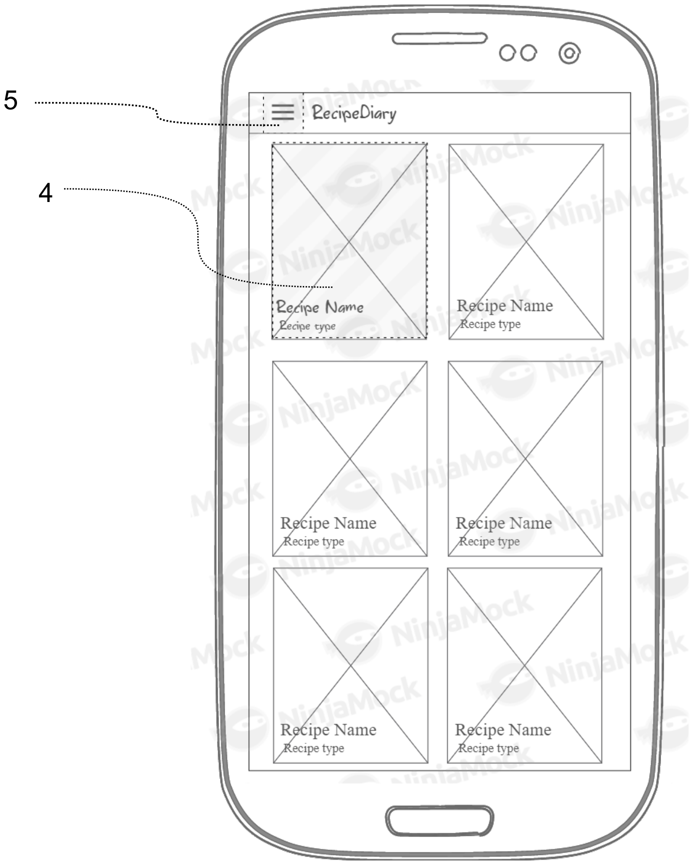
https://www.ninjamock.com/s/GBF56



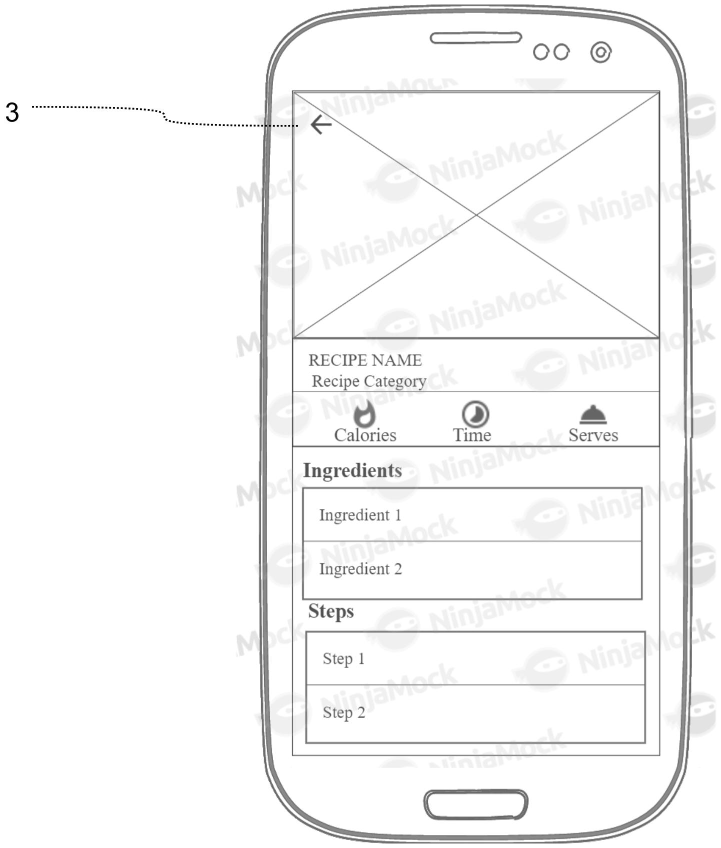
1 - Login Activity



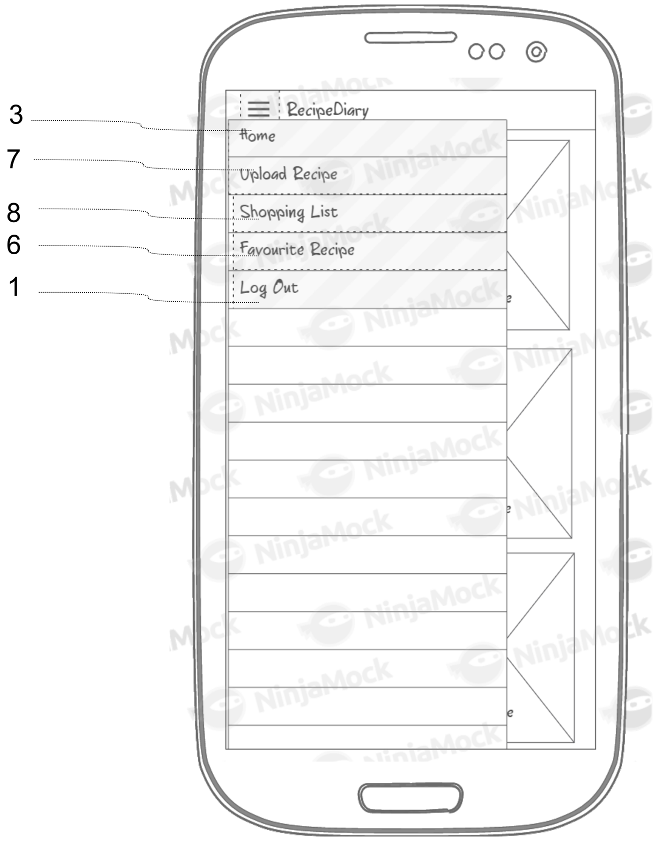
2 - SignIn Activity



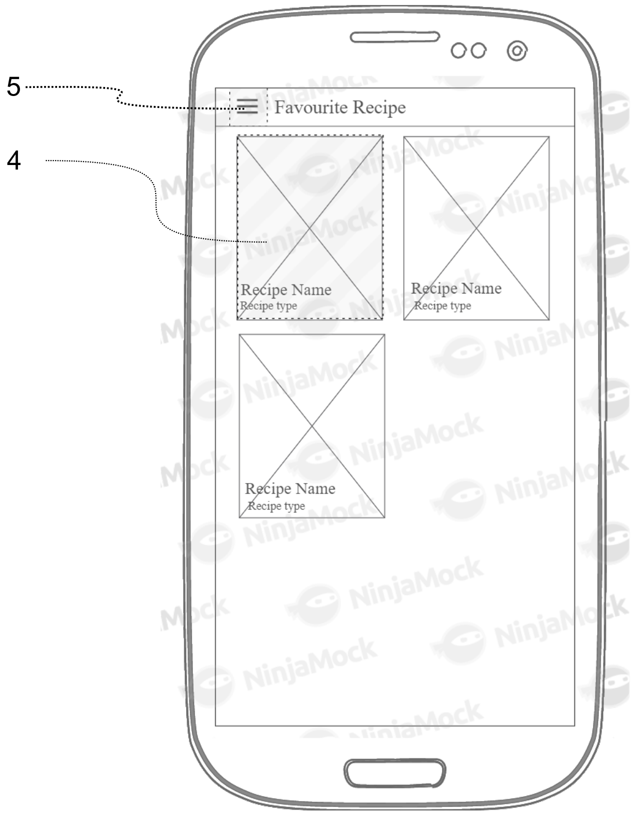
3 - Main Activity



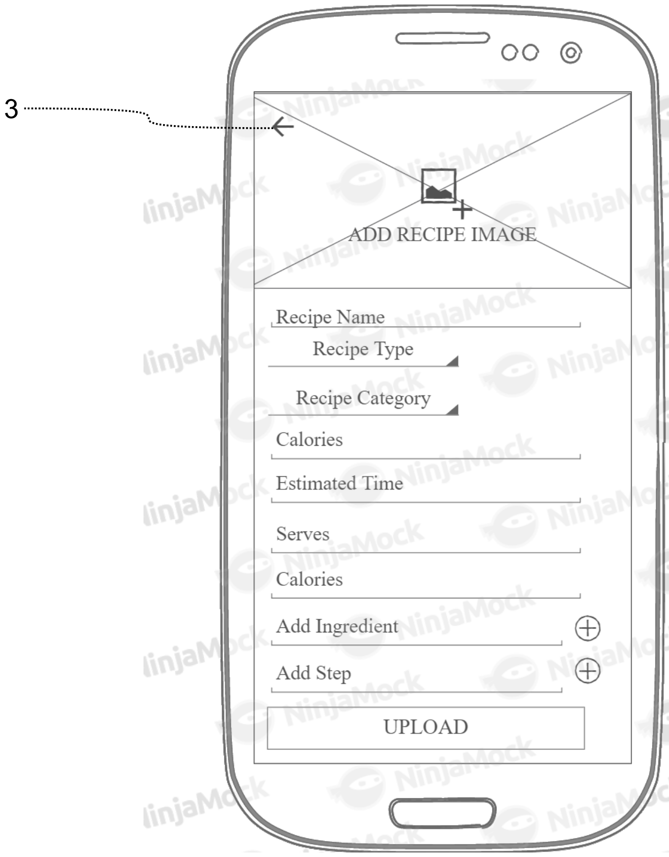
4 - Recipe Detail Activity



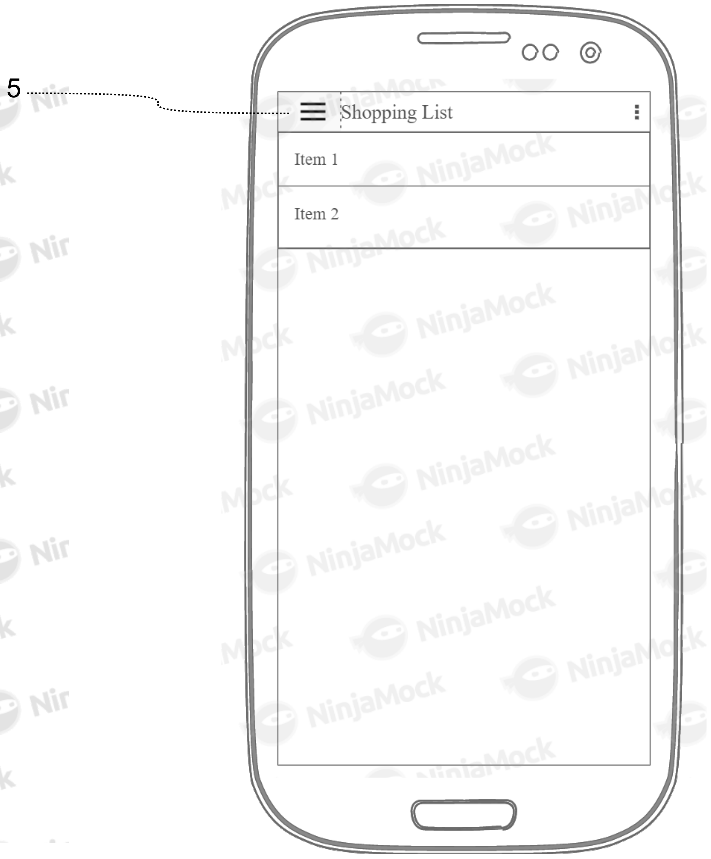
5 - Navigation Drawrer



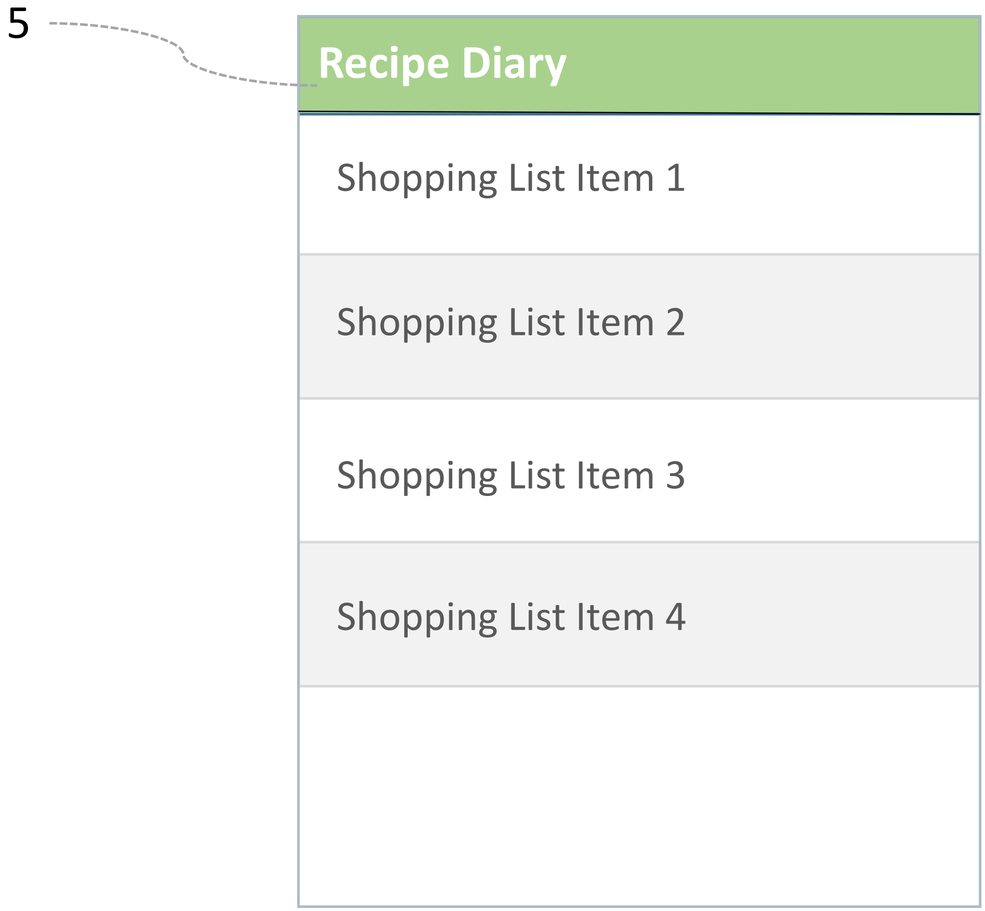
6 - Favourite Recipes Activity



7 - Upload Recipe Activity



8 - Shopping List Activity



Recipe Diary

Shopping List Item 1

Shopping List Item 2

Shopping List Item 3

Shopping List Item 4

9-Home Screen Widget

Key Considerations

How will your app handle data persistence?

I will implement content providers for saving favorite recipes and creating shopping list.

Describe any edge or corner cases in the UX.

In Favorite Recipe Activity and Shopping List Activity if no data would be present then there will be some UI element to indicate that.
If due to some reason network is unavailable a message will be displayed.
While uploading if user rotates screen uploading process will not be affected.

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

1.Retrofit for network operations.

2.Glide for loading and caching of images.

3.Dmytrodanylyk circular progress button library for representing loading while login and

signup.4.Firebase Auth for Login and Registration.

5.Firebase Storage for storing Recipe Images.

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

1.Firebase Storage for storing Recipe Images.

2..Firebase Auth for Login and Registration.

3.Admob to display adds.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

1. Create a new Android Studio Project RecipeDiary.
2. Assign the minimum API Level Required.
3. Once the basic project setup is complete add the important libraries in build.gradle file like RecyclerView,Retrofit,Design Support Library,CardView etc.

Task 2: Implement UI for Each Activity and Fragment

1.Design the layout of MainActivity.Add RecyclerView and create a card layout for it. Implement NavigationDrawer for navigation between different Activities.

2.For Recipe DetailActivity implement coordinator layout for displaying recipe

Information.3.For UploadRecipe Activity use coordinator layout,edit texts and spinners for

getting recipe Information.

4.For Favourite Recipe use recyclerView for displaying recipes.

5.Use recyclerView or ListView for creating shopping list.

6.Design Login and SignUp Activity Layouts.

Task 3: Implement network operations

1. UPLOAD RECIPE:
   For uploading recipes I will use retrofit to PUT recipe to RecipeApi which i have created for this App.
   For storing recipe images I will use firebase storage.Images can be taken through camera or from gallery.
   For list of steps and ingredients editText will get the user input and textView will inflate to show user input.
2. GET RECIPE:
   For displaying uploaded recipes I will use retrofit to GET recipe from RecipeApi.
   Recipes will be displayed in recyclerView in MainActivity.

Task 4: Data persistence

1. Implement Content Providers for handling shopping List and Favorite Recipes.

2. Use CursorLoader for Favorite Recipes.

3. Use AsyncTask with Headless Fragment to handle upload operation during device rotation.

Task 5: Other Tasks

1. Create a widget for displaying Shopping List.
2. Create a MultiPane layout for tablets.
3. App theme extends AppCompat.
4. Use AppBar and ToolBars.
5. App keeps all strings in a strings.xml.
6. Enable RTL layout switching on all layouts.
7.If data does not exist or is in the wrong format, the app logs this fact and does not crash.