
GitHub Username: vadivelansr

Coofde

Description

Coofde - Coupons **of**fers **de**als

Coofde allows users to identify latest coupons, best offers and great deals. Coofde helps users to save money every time they shop online by providing up to date information. Our editors check coupon codes, offers and deals from a large number of online shopping sites to ensure validity every day.

Intended User

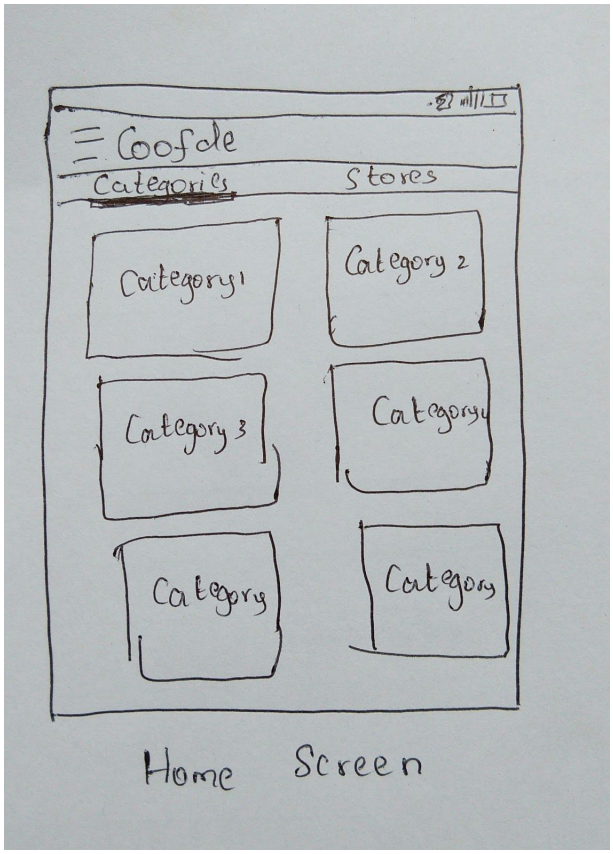
Coofde is developed for users who love to shop online, like to save while shopping and want to save time instead of searching for coupons, offers and deals.

Features

- Allows users to discover coupons, offers and deals in different categories.
    - Each category will have a number of stores displayed
    - Each store will display the coupons, offers and deals under specific category
    - Each coupon or offer or deal will be displayed in a detailed view
    - User will be redirected to the merchant site in a web browser
- Users can browse all the coupons, offers and deals for a specific store.
    - All the stores will be displayed from which the user can select required store
    - Each store will display all the coupons, offers and deals available
    - Each coupon or offer or deal will be displayed in a detailed view
    - User will be redirected to the merchant site in a web browser
- Users can save the coupons, offers and deals which they like for later use.
    - Allows users to view the saved delas even in offline mode.
    - Users can delete saved coupons, offers and deals.

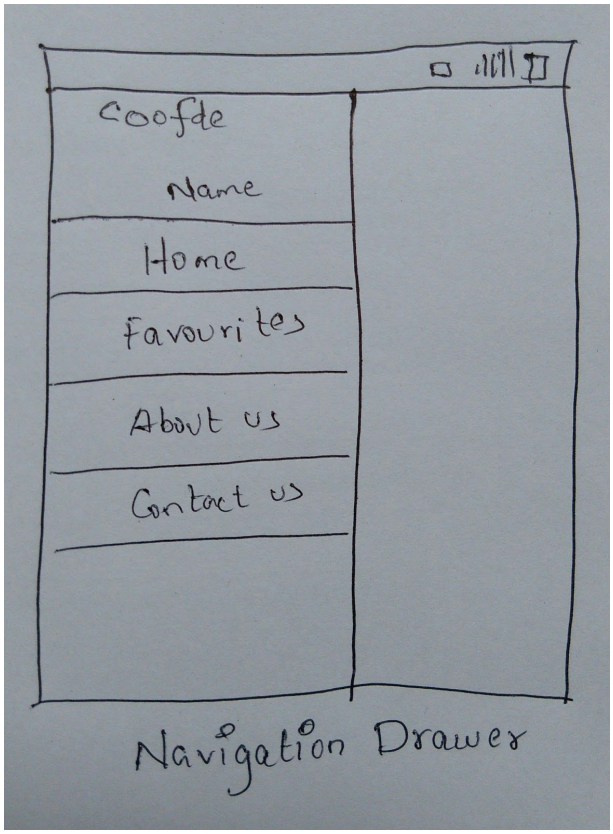
User Interface Mocks

Screen 1



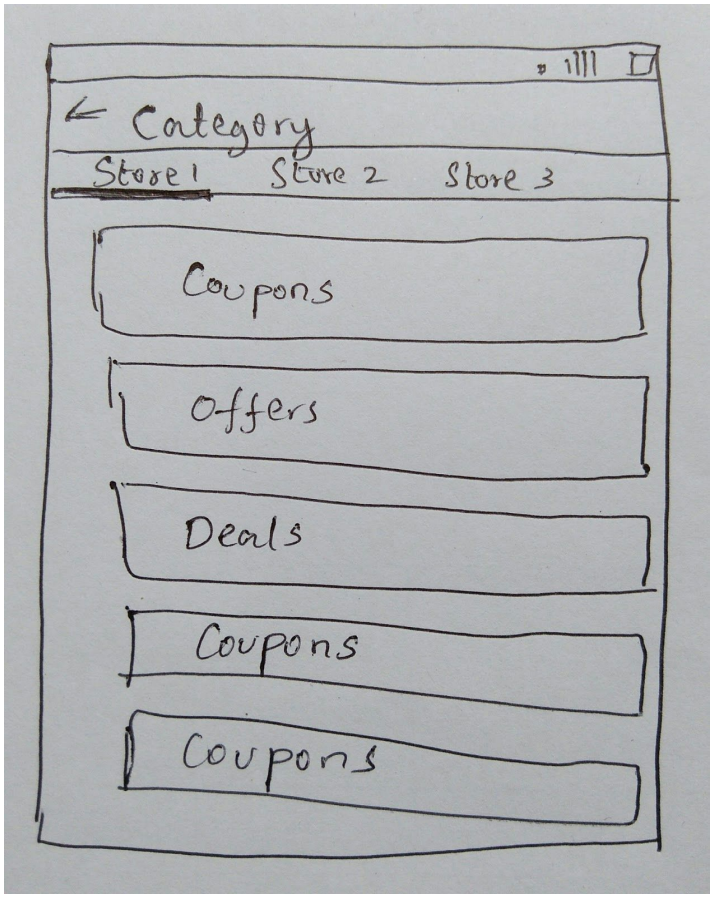
Home Screen - This screen displays coupons, offers and deals in Categories and Stores tab. Categories tab displays the categories in grids and it allows the users to select offers based on specific category.

Screen 2



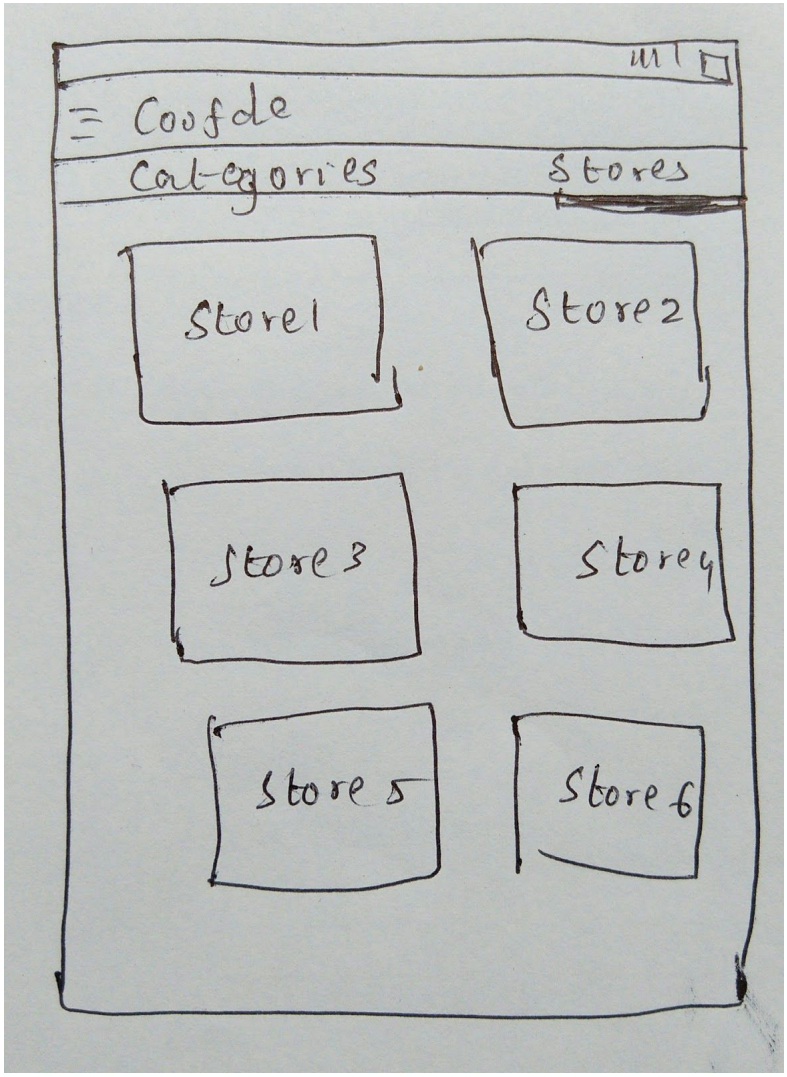
This menu will appear on top of the Home Screen. Navigation drawer allows the users to select Favourites screen to view the saved offers. Also, it allows to navigate back to the Home Screen. It includes general menus like About Us and Contact Us page.

Screen 3



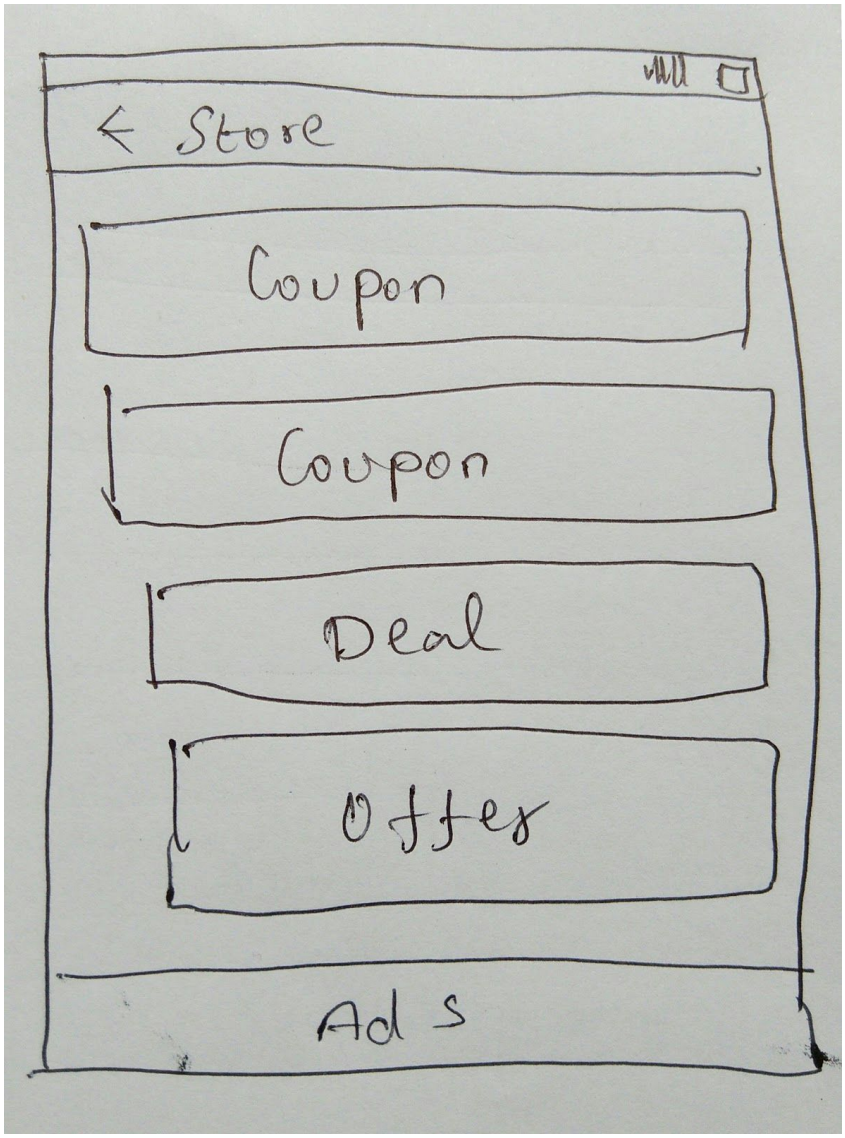
This screen displays the stores as tabs for a specific category. Each store tab displays the list coupons, offers and deals for that specific store from which the users can select.

Screen 4



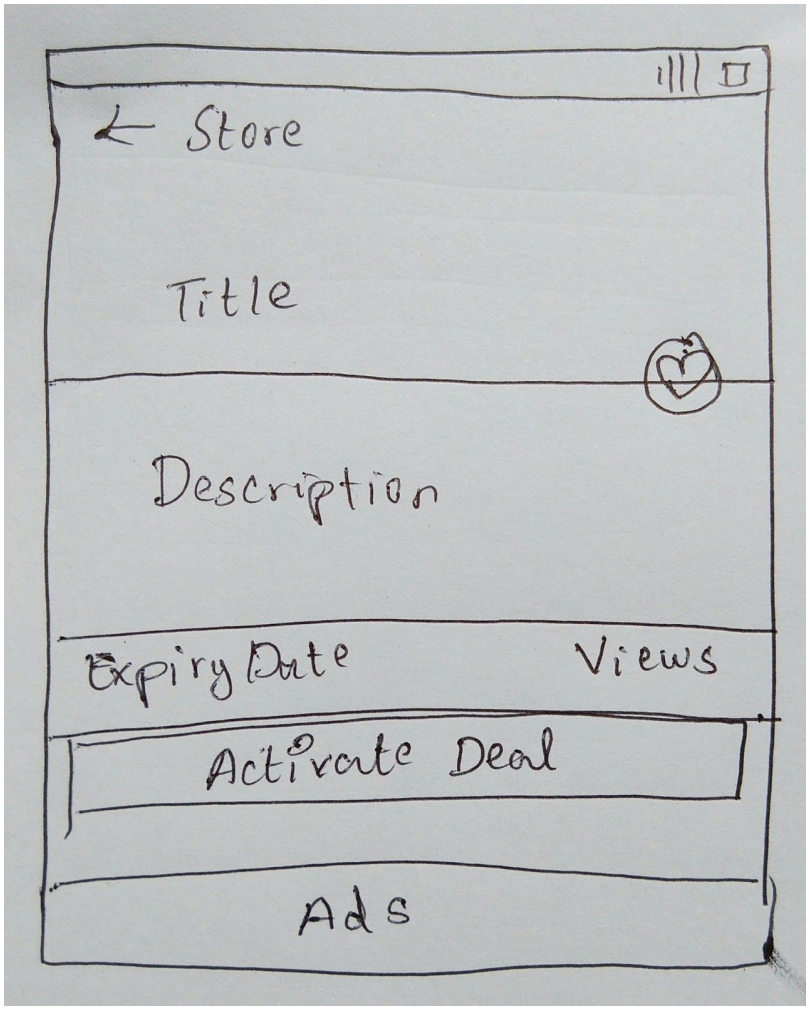
This screen displays the all available stores in a grid view. Users can select any store to view the list of available coupons, offers and deals.

Screen 5



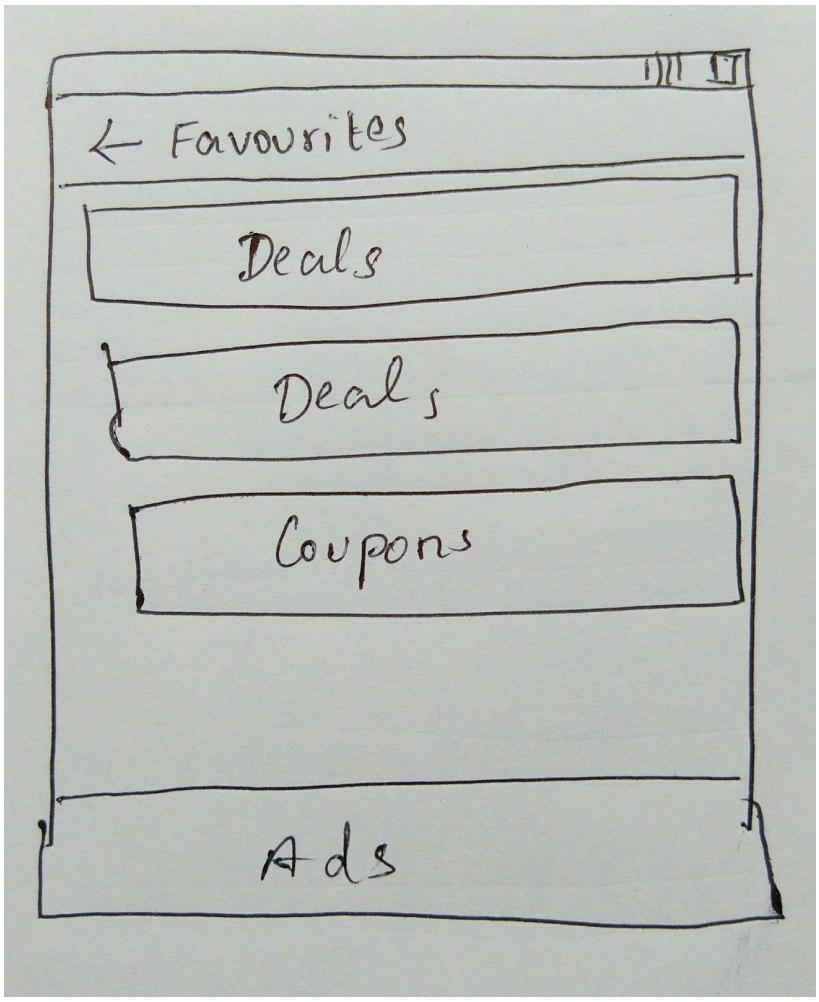
When users select a store from Stores tab in Home Screen this screen will be displayed. This screen lists all the available coupons, offers and deals from which the user can select the intended one. The screen also displays AdMob ad in the bottom.

Screen 6



This screen is the details screen which displays the title, description, expiry date, number of views about a coupon or offer or deal. The screen has a FAB which allows the user to save the offer for later or offline use. It also has a button which will display the coupons or redirect the user to the merchant site through a web browser. The screen also displays AdMob ad in the bottom.

Screen 7



The screen displays the user favourites in a list. The user can select and view the offers in offline.
The screen also displays AdMob ad in the bottom.

Key Considerations

How will your app handle data persistence?

All the coupons, offers and deals will be stored in Firebase and it acts as the backend for the app. Firebase offline mode is enabled to provide offline functionality for the app. Also, Content providers are used to save the deals locally and to view whenever necessary.

Describe any corner cases in the UX.

If an image is unable to be loaded the app should not error out, but rather use a placeholder image that lets the user know there was a problem.

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

Picasso is used to handle the loading and caching of images.
Butterknife is used to bind views and listeners.

Next Steps: Required Tasks

Task 1: Project Setup

Major steps to setup the project:

- Update Android Studio to latest stable version
- Update Android SDK
- Configure Git repository
- Configure gradle dependencies
- Configure Firebase

Task 2: Implement UI for Each Activity and Fragment

Subtasks for the project:

- Setup basic UI structure, Navigation Drawer
- Build top part of each activity/fragment (AppBar, CoordinatorLayout)
- Build each separate activity/fragment and navigation/transition to and from those activities/fragments
- Build Master/Detail for Tablet Support
- Build UI elements
- Integrate Firebase with respective activity or fragment

Task 3: Implement Data models and Content Provider

Build up the data models and implement data persistence.
Subtasks:

- Create data model classes
- SQLite database setup and CRUD
- Loader/adapters and UI

Task 4: Configure Firebase

Setup the new Firebase app in firebase.com website.
Subtasks:

- Create an account in Firebase
- Create Firebases in dashboard and populate handpicked coupons, offers and deals
- Setup FIrebase on Android
- Implement activities or fragments with Firebase

Task 5: Google Play Services

Subtasks:

- Setup initial Google play communications
- Setup AdMob ads in activities
- Setup Google analytics wherever necessary

Task 6: Others

Subtasks:

- Widget
- Accessibility
- RTL

Task 7: Testing

Subtasks:

- Rotation
- Phone vs Tablet
- Performance