CAPSTONE_STAGE1 - FOOD VOOD APP PROPOSAL

GitHub Username: madhurgupta10

FoodVood App

Description

Tired of standing in the lines or waiting for waiters to hit your table for order?
Well don't punish your empty stomach or scratch your hair anymore.

FoodVood is here, this app let you order your favourite food from your favourite restaurants by click of a button!

And wait the best part is you don't have to sign up for an account, just order your food and collect it from the counter showing your order id and pay the amount to restaurant.

Hungry? Get the FoodVood app now :P

Intended User

Travelers, Foodies, or anyone who is hungry.

Features

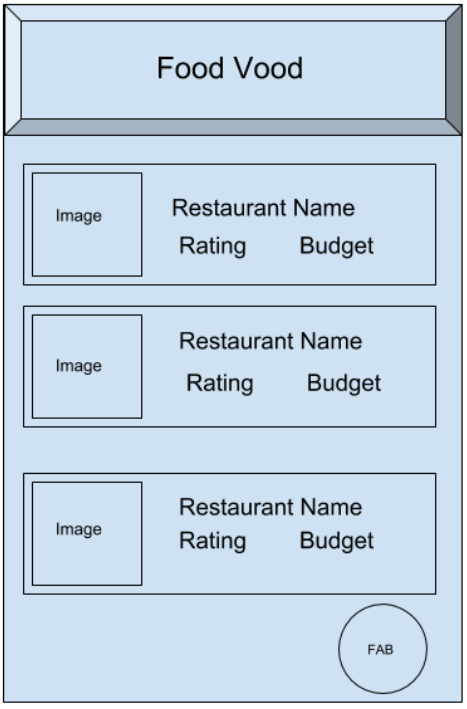
List the main features of your app. For example:

- Explore Nearby Restaurants and there menu.
- Calculate your total amount and see offers.
- Order food by click of a button

User Interface Mocks

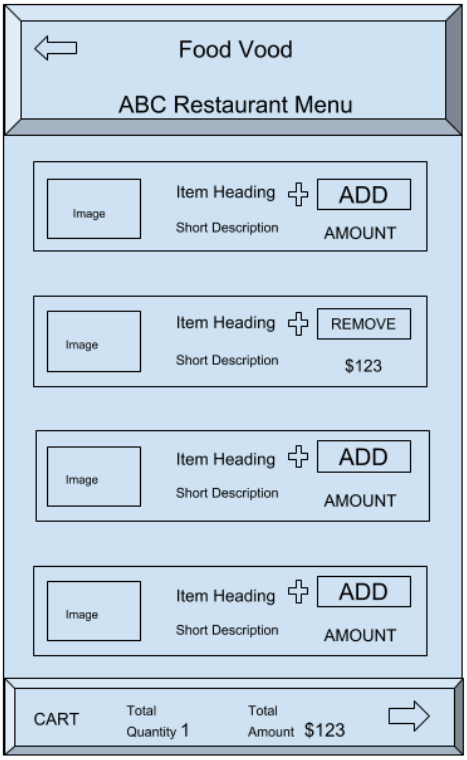
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



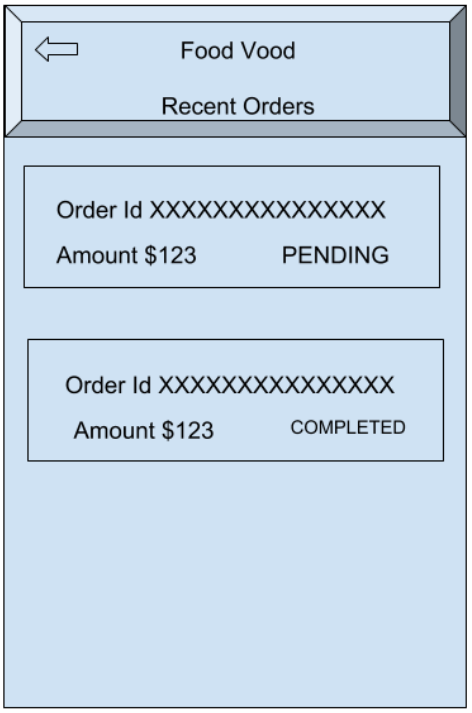
This is the Launcher Activity. In this activity, user will see available restaurants, ranked on the basis of their rating. User can select any restaurant they want.
User can click on FAB and it will take them to orders activity, where they can see there most recents orders.

Screen 2



On Clicking upon a restaurant, the user will be directed to this activity. This activity let user add/remove his favourite items in the cart and he/she will see total items and amount.

Screen 3

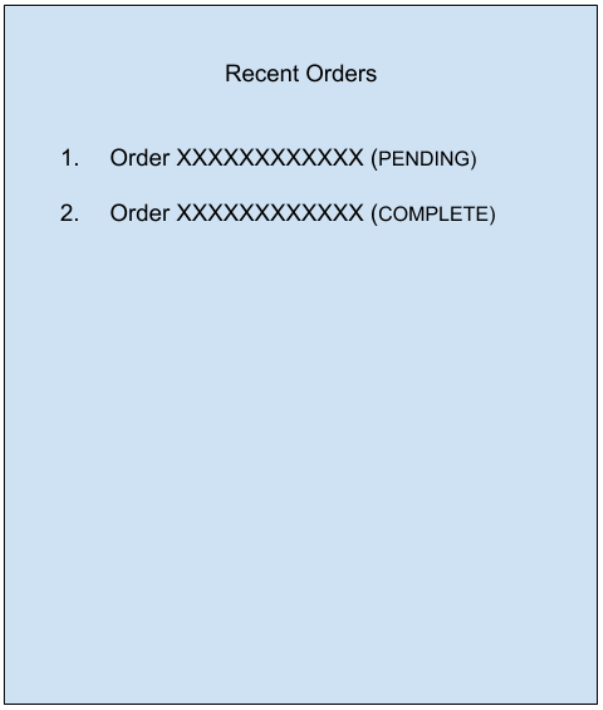


On Clicking upon a checkout arrow, the user will be directed to this activity. This activity let user see the recent orders, there order id, amount and status. User can click on any order and see more details about it.

Screen 4

On This screen user can see more details about a particular order

Screen 5 - Widget



Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

The app will user Firebase for all kinds of data handling. The Information will be fetched from firebase and after user place, order the order is sent to

Describe any edge or corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

We will be using Picasso for loading images, ION or OKHTTP, incase we need to fetch data from our custom cloud api. PaperDB for saving data to widget.

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

We will use Firebase RealTime Database for our most of the tasks, from fetching restaurant information to saving order details.

We will also use Google Maps Services to know the nearby restaurants within a 1km Circle.

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

First Create a Basic Activity from Android Studio Wizard.

Then add the required dependencies to build.gradle file.

The dependency can be found on Maven Repository Website, copy and paste the url from there to build.gradle file in app folder.

Task 2: Implement UI for Each Activity and Fragment

Next Build the UI for each Activity.

Don’t forget to move hardcoded strings, colors, etc. to their respective xml files.

Task 3: Permissions

Add permissions to Manifest files.

Task 4: Fetch Location Address

Fetch Location Coordinates and display a Toast message with location address using google maps service.

Task 5: Java and Recycler View

Next Complete the other UI components. Make corresponding Java Class for serving those activities.

Make Recycler View Adapters for each of the activity.

Task 6: Firebase

Fetch Data from firebase, display them via AsyncTask Loader for each corresponding activity.

Task 7: Widget and Testing

Next Complete the widget, test and debug the app.

Add as many tasks as you need to complete your app.
