GitHub Username: https://github.com/shenoybrs

OfferBank

Description

Offerbank has the combinations of all the offers provided by customers bank through credit card,debit card and/or through wallets in one place . Customer will usually have 3 -4 credit or debit cards ,they even have wallet accounts at any point of time . The banks randomly open up offers to their customers, but the customer don't get to know this. Offerbank gives a one stop solution to this where customer can see all the offers in one place before making a purchase so he does not miss a cash back or 10X points or discounts though he possesses the banks card.

The Banks will be extremely happy to see this app .

The Merchants who have opened the offers will be happy that there app is discovered and installed as the app flow helps the customer to download and install the app with ease.

Intended User

All the bank customers , all the banks and wallet companies who want to showcase their offers to all their customers . Being mindful of the thought that all the banks and wallets will have their own apps or website but user can't keep checking all the offers , hence this app.

Features

List the main features of your app. For example:

- Select your banks from the list of banks mentioned
- Pull the offers from the server.
- If there is an app providing the app , help user to install it.
- If the app is already installed , the customer is taken to the app and shown the offers.
- If there is no app take the customer to the website.

User Interface Mocks

Flow1

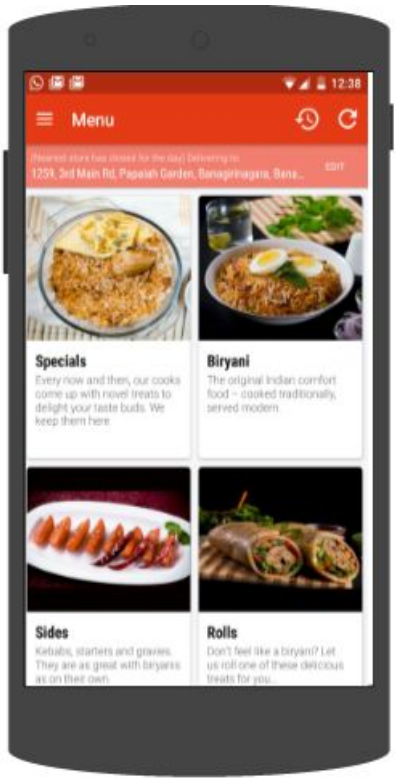
The above flow is as follows ,

1) Select the banks from the list .

2) Fetch the offers
3) On click of the offer , if the app does not exist open google play store to install the app.
4) Use the app to redeem the offer.

PS:- The merchant would love to pay for this app installation .

Flow 2:-



Install the app through the google flow and on back press open button is enabled on click of the open button the app will open up and the transaction can be made

The ammis biryani app is open.

How will your app handle data persistence?

Data will be persisted in the app for all the banks for 24 hours and then will be refreshed.

Describe any corner cases in the UX.

1) User installs the app . App will require data connectivity to pull the offers , if the app does not have good network then whenever network is available the offers will be pulled and stored in the database to show the default bank. The user will be “notified” through a notification.
2) To open the native app google play service is needed and also google play store app is needed.

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

The major libraries used are volley , okhttp .

Next Steps: Required Tasks

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

Task 1: Project Setup

- Server configuration to pull the latest offers.
- List of banks providing the offer atleast one bank need to be selected or else default selected bank offers will be displayed.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build Server to provide the data to the client.
- Build UI to select from the list of banks or wallet providing offers
- Build UI for OfferActivity
- Build UI for DetailActivity.

Task 3: Server Task

Server will use script.google.come for providing data to the client.
Server will fetch the data from the google spread sheet which the administrator will fill on day to day basis.
Server will also provide the list of bank names which would have the offer .

Task 4: BankList Activity

BankList Activity will display the list of banks . User needs to select from this list so he gets the offers regularly. User has to select atleast one bank to move to the offer screen.

a) Build layout for the Activity/fragment
b) Persist the user selection.

Task 5: OfferActivity

Will also depend upon the selected banks from the user list to fetch all the offers from the server. The offers will be fetched only for the user selected banks. The offers will be stored in the database with an contentprovider , A loader will help to display all the offer to the user .

a) Build layout for the Activity/Fragment
b) Build the asynchronous call to fetch the list of offers for the branch
c) Build the contentprovider to store the list of offers.
d) Build the loader to pick the data from the contentprovider to load to the list which will be displayed to the user.

Task 6: OfferDetailActivity

Detail page of the offer is being showed giving the cash back info and some images if needed ,each offer will either be for website or will be for an app . From the offer page the next click will either take user to the website or the native app , if the app is not present then the user is taken to google play store for the user to install it .

a) Build layout for the Activity/fragement
b) Build the asynchronous call to fetch the detail of the offer.
c) Build the contentprovider to store the offers details
d) Build the loader to pick the data from the contentprovider to load to the list which will be displayed to the user.

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

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"