

OffersGalore

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Screen 7](#)

[Screen 8](#)

[Screen 9](#)

[Screen 10](#)

[Screen 11](#)

[Screen 12](#)

[Screen 13](#)

[Screen 14](#)

[Key Considerations](#)

[How will my app handle data persistence?](#)

[Corner cases in the UX.](#)

[Libraries I'll be using and my reasoning for including them.](#)

[How will I implement Google Play Services](#)

Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: Design Local and Remote Database

Task 4: Implement Login Flow

Task 5: Sync Local DB with remote DB

Task 6: Adding Customers and Offers

Task 7: Handle Error Cases

Task 8: Send sms for created offers

Task 9: Create App Widget

Task 11: Configure installRelease in build.gradle

Task 12: Create wrapper for realm using content provider.

OffersGalore

Description

This App is for the owners of small stores, which will allow them to save their customer details and send offers to the added customers through SMS.

Some of the use cases for this -

- a) the app will help them get rid of their old stock at a discounted rate,
- b) It will help them in competing with other stores
- c) It will help them keep the customers informed about the latest products based on their interests as per the availability
- and
- d) In due course will help build customer loyalty.

Intended User

This App is mainly for small shop or small grocery store owners, allowing them to create promotional offers through SMS.

Features

List the main features of your app. For example:

Login with mobile number.

OTP verification (sms is sent from merchants SIM who will be charged by the network provider)

Allows Sellers to create or add customers details with mobile number(mandatory), interests ,name and email.

Allow Sellers to edit created customers.

Allows Sellers to create offers with the saved list of users or from the phonebook, with a scheduled date and offer text.

Allow sellers to edit scheduled offers that has not been sent already

Filter offers based on scheduled and sent

Sends SMS to all the selected customers in the offer when the scheduled date is reached(sms is sent from merchants SIM who will be charged by the network provider, this could be replaced with a bulk sms api in future).

Incase of returning user, fetch customer and offer data from firebase on launch of landing page.

User Interface Mocks

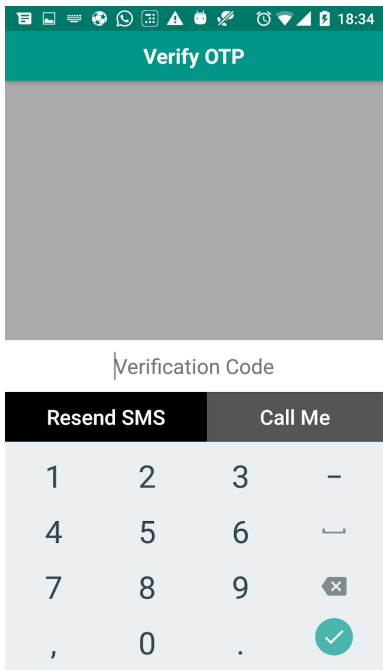
Screen 1

The mockup shows a mobile app interface for phone number verification. At the top is a teal header with the text "Verify Phone Number". Below the header is a white input field with a placeholder "Enter Phone Number" and a "+91" country code selector. Under the input field are two links: "PRIVACY POLICY" and "TERMS AND CONDITIONS". Below these links is a grey button labeled "ACCEPT & CONTINUE". At the bottom is a numeric keypad with digits 1-9, *, #, 0, and a green checkmark button. Each digit has its corresponding letters displayed next to it.

ACCEPT & CONTINUE			
1	2 ABC	3 DEF	-
4 GHI	5 JKL	6 MNO	↩
7 PRQS	8 TUV	9 WXYZ	✕
* #	0 +	.	✓

Login with Phone Number verification, 10 digit phone mobile number .

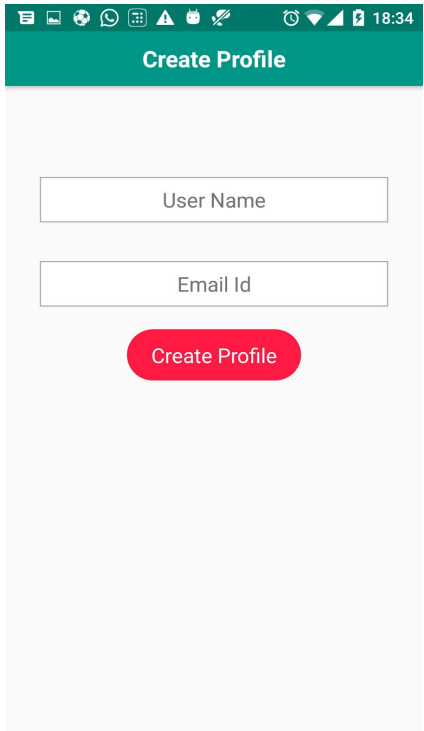
Screen 2



User will receive otp through sms and will have to get it verified here.

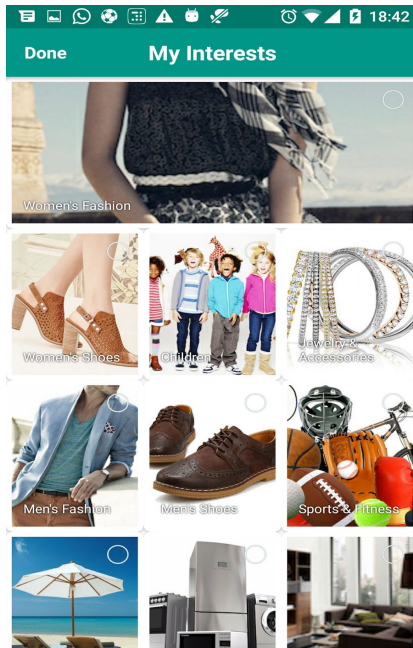
3

Screen 3



Create Profile with username and email id

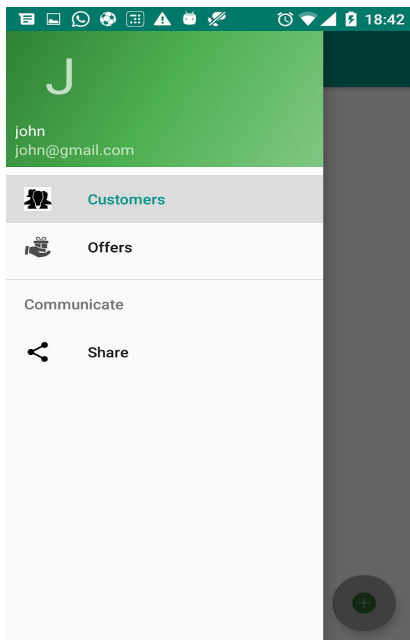
Screen 4



Select interests of merchant just for records.

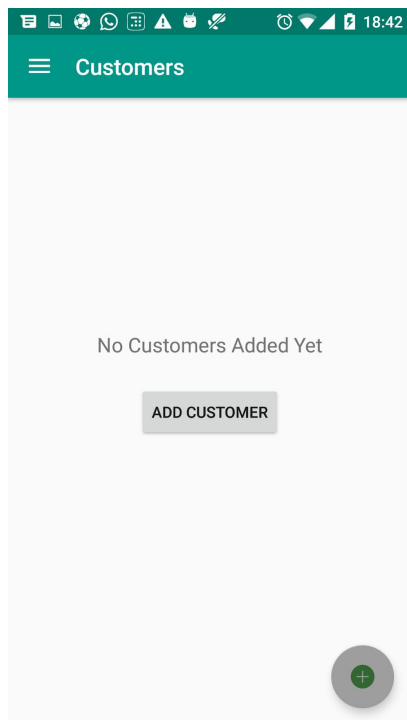
4

Screen 5



Landing page with navigation drawer displaying 2 main options Customers and Offers.

Screen 6



Page that displays created customers in case not created displays a button with option to add customer

Screen 7

Screen 7 is the 'Add Customer' form. It features a green header bar with a back arrow and the title 'Add Customer'. The form contains three input fields for 'Enter Name', 'Enter Mobile Number', and 'Enter Email'. Below these fields is a 'Select Interests' button. At the bottom of the form is a large grey button labeled 'Add Customer'. A mobile keyboard is shown at the bottom of the screen.

On click of the + button or the “Add Customer” button in the previous screen, it launches this page to add customer.

On Click of “Select Interest” button launches Screen 4 to register customer interest.

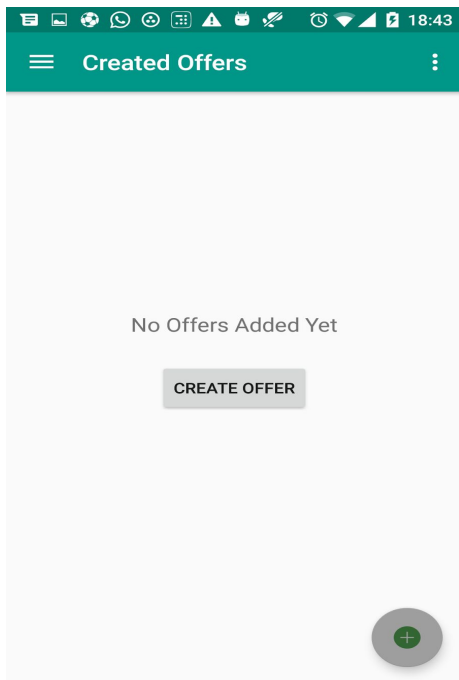
Screen 8

Screen 8 is the 'Customers' list screen. It has a green header bar with a hamburger menu icon and the title 'Customers'. The screen displays a list of three customer cards. Each card shows a blue square with a white letter (T, T, S), the customer's name, email address, and phone number. At the bottom center of the screen is a circular button with a green plus sign.

Initial	Name	Email	Phone Number
T	tom	tom@gmail.com	9880311011
T	tim	tim@gmail.com	9845738620
S	santy	santy@gmail.com	9844587768

On creation of customer comes back to screen 6 where it displays the list of Customers created.

Screen 9



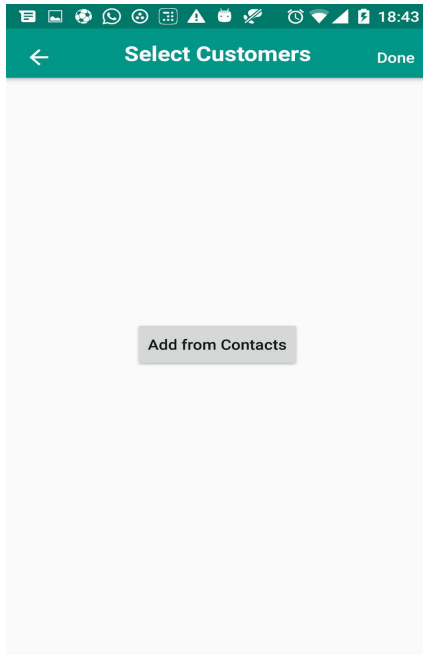
On click of the offers tab in navigation view on screen 5 it displays this screen, which lists the Created offers and incase no offers created displays a “create offer” button to create offer.

Screen 10

A screenshot of a mobile application screen titled "Create Offer". The screen has a teal header bar with a back arrow on the left and the text "Create Offer" in the center. Below the header are three input fields: "Enter Date", "Enter Offer text", and "Select Customers". The "Select Customers" field has a blue button with the text "Select Customers". At the bottom of the screen is a gray button labeled "Create Offer".

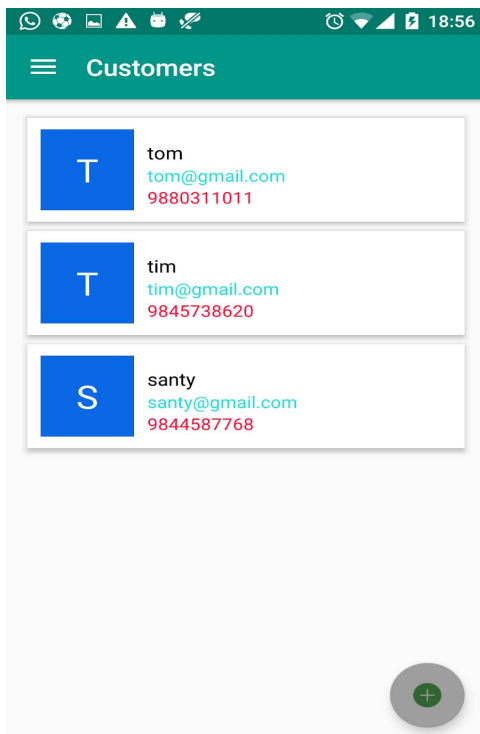
On click of create offer from previous screen it displays this screen where you can enter the date, as to when the offer needs to be sent , along with the offer text and the customers.

Screen 11



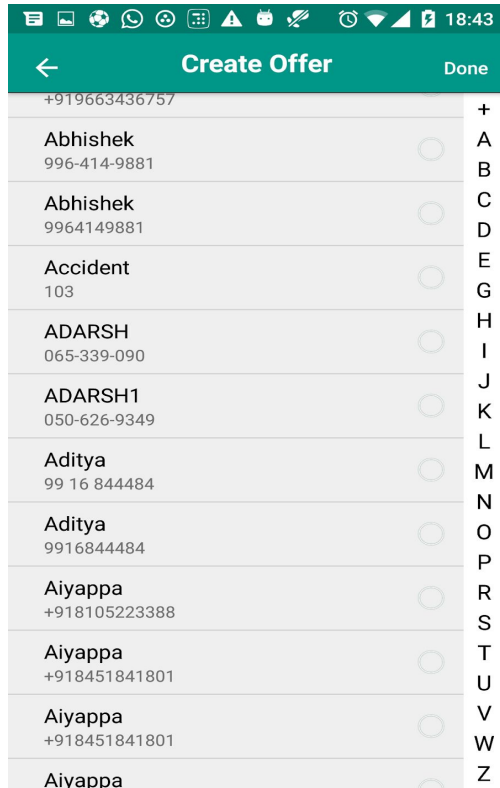
On click of “Select Customer” button from previous screen it launches the list of customers added if not added then it displays an add from Contacts button to add from phonebook as shown in this screen.

Screen 12



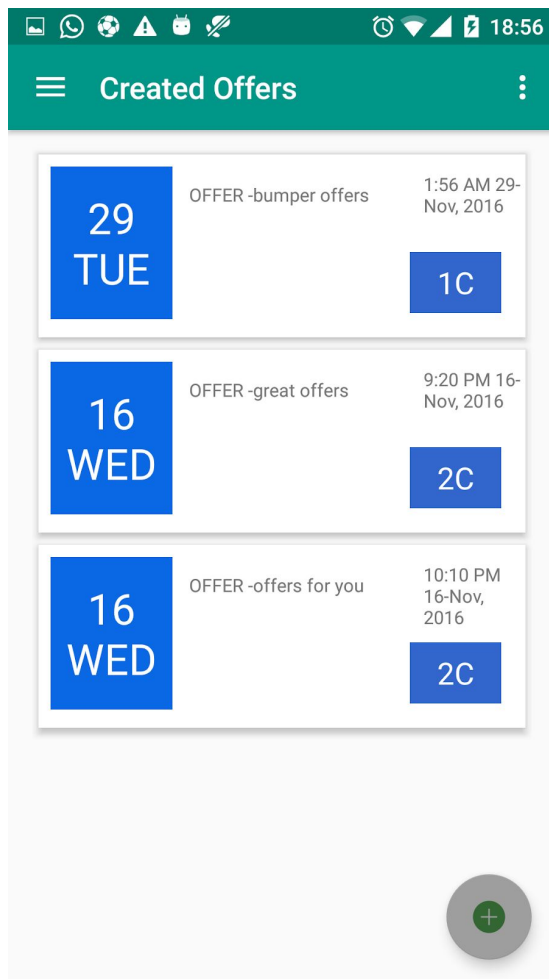
On click of “Select Customer” on screen 10, it navigates to this screen where it displays the list of customers to choose from for the offer.

Screen 13



On click of “Add from Contacts” on screen 11 it launches this screen with an option to select customers from phonebook.

Screen 14



On click of create offer button in screen 10, it creates offer and navigates to this screen where list of offers created are displayed.

A pending intent is created to send the SMS at the scheduled date to the numbers of all the selected customers for the created offer.

Key Considerations

How will the app handle data persistence?

The app uses firebase for remote persistence and realm database for local persistence.

Corner cases in the UX.

The app has a navigation drawer to navigate to 2 of the major features provided like offer and customer.

It also has detailed view for tablets required for editing and saving created customer details and created offer (scheduled offers only) details.

Libraries I'll be using and my reasoning for including them.

I Will be using realm library to persist data as it makes implementation easier while using firebase since both are nosql databases, also integration is faster and conversion of json to object is already available as firebase libs.

Also Picasso for images and Butterknife for databinding.

How I will implement Google Play Services.

I'll be using firebase which is a part of google play services to store data remotely, which will be useful if the user uninstalls the app and reinstalls it with the same number, in which case it shall fetch the data(customer and offer related) already created remotely from firebase.

Required Tasks

Task 1: Project Setup

Configure firebase with android project following instructions on the link - <https://firebase.google.com/docs/android/setup>

Add libs Butter Knife, realm and Picasso.

Task 2: Implement UI for Each Activity and Fragment

Build UI for splash screen ShopOnSplashActivity

Build UI for login screens ShopOnMsisdnActivity, SmsOtpVerify, ShopOnProfileCreation,

Build UI for interests screen ShopOnCategoryActivity.

Build UI for landing page ShopOnActivity with NavigationView(displaying two tabs offers and customers).

Build UI for customer screens CustomerFragment, CustomerActivity, CustomerDetailFragment, CustomerDetailActivity.

Build UI for offer screens OfferFragment, OfferActivity, OfferDetailFragment, OfferDetailActivity.

Handle UI for tablets by adding detail screen(CustomerDetailFragment, OfferDetailFragment) and created list(CustomerFragment, OfferFragment) in one screen.

Task 3: Design Remote and Local Database

On firebase, design 3 different dictionaries in the database, one for merchant, 2nd for merchants customers and 3rd for merchants offers.

Map each of the dictionaries in the firebase to objects. Have to create 2 pojos for all 3 dictionaries because firebase cannot use pojo that extends RealmObject.

Create utils functions for creating retrieving and updating data in firebase.

Create util functions for converting firebase pojo objects to realm pojo objects and vice versa.

Task 4: Implement Login Flow

On Number verification screen generate and send sms through subscribed SIM

Store number and OTP in shared pref.

Store login

state(msisdn_state,otp_state,profile_state,category_state,login_complete) in pref on each of the login screen so that incase user closes app before completion he can revisit the state where he had left off.

On otp verification screen check if number already created in firebase, if created then proceed to landing page(ShopOnActivity) else proceed to create profile.

On create profile screen process profile name and email id and proceed to interest screen,

Select interest and save all the details to firebase.

Task 5: Adding Customers and Offers

On Created Customers screen list all the created customers.

FAB button to launch Add Customer, in Add Customer screen create customer with name, email, interests and number.

On Created Offer screen list all the created offers.

FAB button to launch Create Offer, in Create Offer Screen create offers with selected numbers of created customers, offer text and schedule date.

Edit options for created offers that has not been sent yet.

Edit option for created customers

Filter option based on scheduled and sent Offers in Created Offer Screen.

Task 6: Sync LocalDB with RemoteDB

Sync Local Offer table with remote offer dictionary

Sync Local Customer table with remote customer dictionary

Initiate sync on launch of landing page.

Task 7: Handle Error Cases

Check Internet connectivity before proceeding with OTP verification.

Field validations for creating profile and in edit offer for and edit customer, create offer and create customer page.

Add permissions to access features programmatically for marsh mellow onwards to selectively grant permission through user input to use phone features like sms, phone state and to access contacts.

Task 8: Send SMS for created Offers

On creation of offers create a pending intent to send sms and update the status locally and remotely.

Send SMS through merchants registered SIM(wil be charged by network provider , atleast for this version).

Task 9: App Widget to display Scheduled or sent offers for the current day

Create App widget.

Display the list of scheduled or sent offers for the current day.

Task 10: Configure InstallRelease command in build.gradle

Generate .jks file and store credentials and file in app

Confiure path to jks file and key value in gradle.properties file.

Configure gradle installRelease build

Task 11: Create wrapper for realm using content provider

Create content provider.

Override content provider methods to perform realm specific operations.

Use content provider to query, update, insert, data and use loader in list view.