



APP CODE REPORT

Prepared By
JAVERIA BALOCH
MAHARAM ISMAYILZADA



Table of Content

1. Introduction
 2. Logo selection
 3. Code documentation
-

Introduction

An Simple app which doner shop can use to make online delivery option possible for them. App has many functionalities and project specifications. Some of them are given below:

Functionalities:

An Application for a doner shop. App will include following functionalities:

- Home (including the List of food Items)
- Order History
- Complains
- Cart Page
- Form for data collection to use reservation, takeaway and home delivery Option
- Sign Up
- Sign In
- Sign Out
- Profile Update

Project Specifications:

Project Specifications are given below:

- Project will be designed in figma,
 - Kotlin will be the main coding Language in android studio
 - Firestore realtime database will be used as a database.
-

Logo Selection

There are many discussion involved in this project such as logo design and theme selection for the project, on the first go we prepare many logos and initial black and white themes.

Logos:

Here are few logos we initially prepared for our project:



And after the few debate and discussion we selected the logo given below:

Selected Logo:



Code Documentation

Screens:

Here is the list of screens we have created for the app:

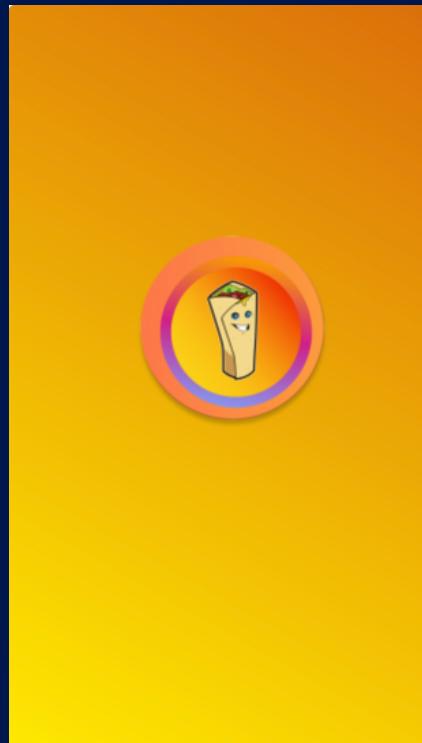
- Splashes Screen
- Home Screen
- Order History Screen
- Complains Screen
- Cart Page Screen
- Form Screen
- Sign Up Screen
- Sign In Screen
- Sign Out Screen
- Profile Update Screen

Assets

There is one menu.json file under and this file contains all the data like, food item name, category, picture url and price,

Splash screen

Whenever you opens the app splash screen will appear for 3 seconds. and then user is navigated to then the user will be navigated to register Intent (which means the register screen).



Register Screen

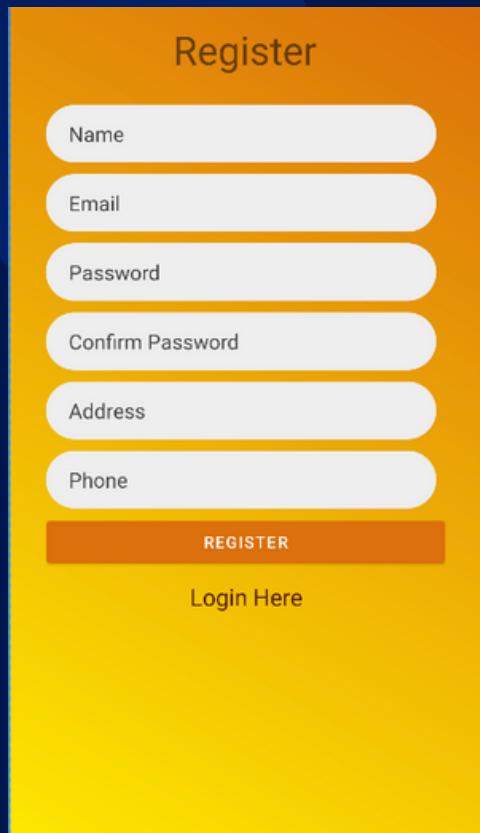
Register screen contains the form for user register, which contains text fields to create a new user. you can see those fields on right.

It will save the user data under "userinfo" in preferences and as we said user for the splash screen will be navigated to this screen so this screen will check that userinfo["name"] and userinfo["id"] exists or not, if it exists user will be navigated to home screen otherwise the need to register in order to use the app.

There are few validation checks while registering, like:

- All fields are required or no field should be empty while clicking the Register button.
- Password should be eight characters long.
- Password and confirm password field's input text should match.
- email is not already registered with all as our app will check this condition from firestore and if it exists app will ask the user to login

Firebase auth.createUserWithEmailAndPassword function will be used to create a new user and if user is create, user info will be store in users firestore collection and also stored in user preferences under userInfo. once all this is done a toast message "Register Successfully" will appear on the screen and user will be navigated to home screen.

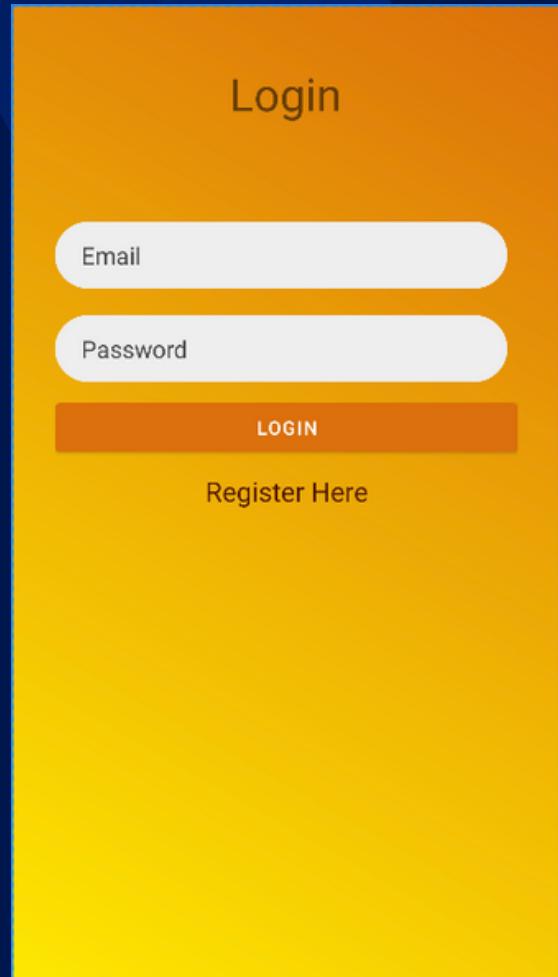


Login

In Login Screen user is required to put email and password to login. On the click of Login button, firstly app will check that password and email fields are having text inside them and then invoke firebase loginWithEmailandPassword will be called if the login went successful, user id will be stored in a user constant and firestore collection users will be searched and data where id is equal to user constant will be retrieved from collection.

Once information is retrieved from firestore then user information such as name, email, id, address phone will be stored in preferences named "userInfo", and finally user will be navigated to MainActivity and from there to Home Screen.

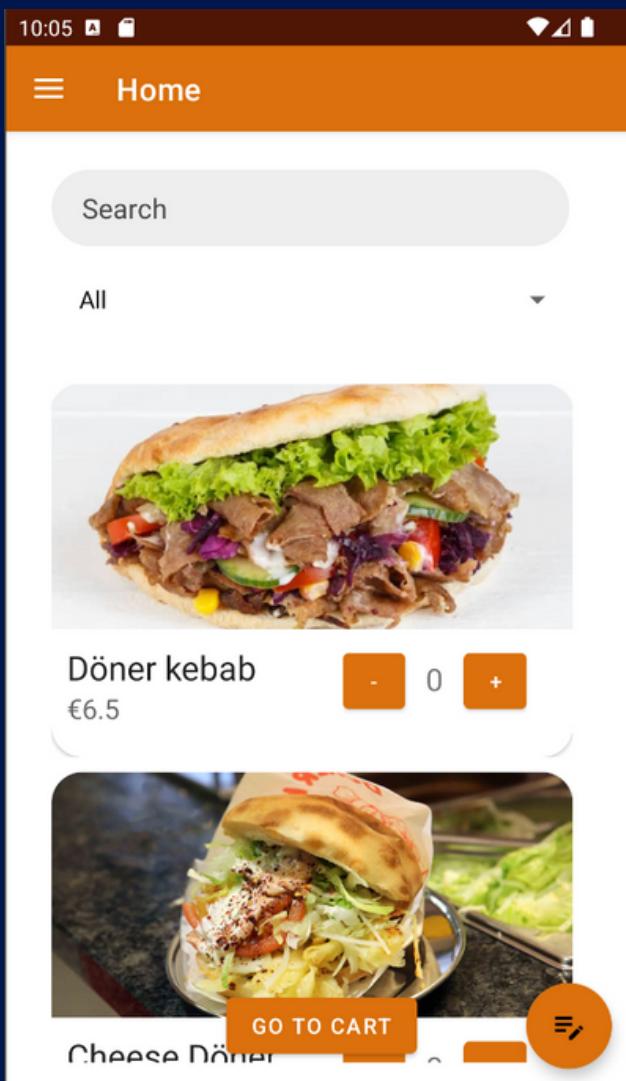
In Login attempt failed, a toast will be displayed on screen informing the user that login is failed due a problem.



Home Screen

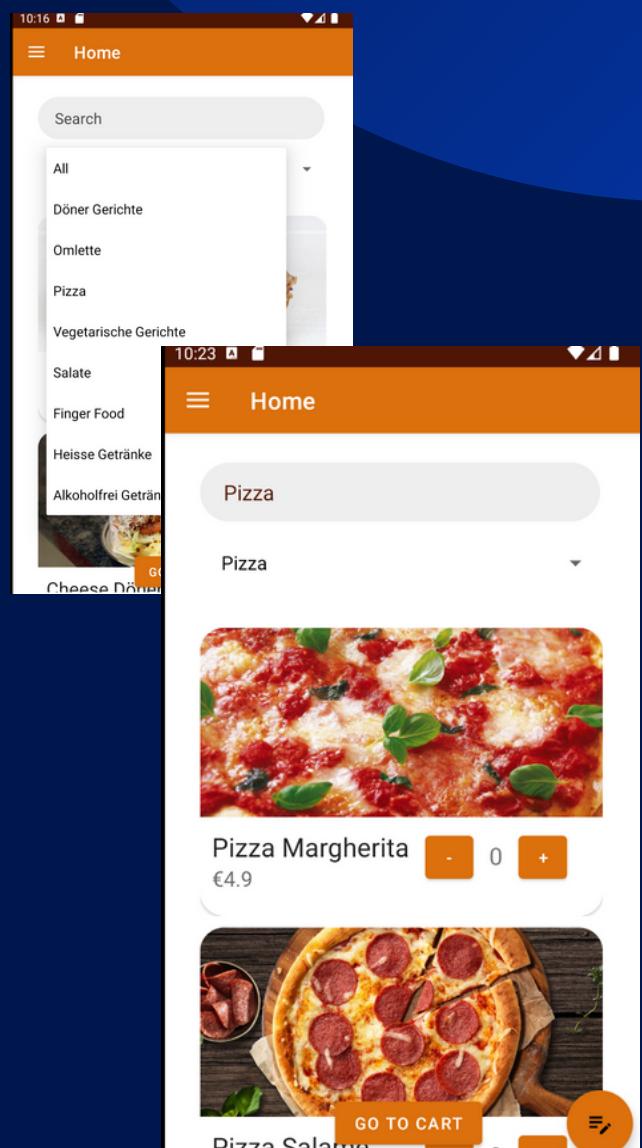
Actually home screen is the primary fragment as it is under drawerMenu. Home screen contains a EditText, Spinner, Recycler view, go to cart and floating complain button.

EditText can be used to search for any food and it will update the recyclerView food item cards on the changing of text. There is category named String Type array show the categories in Spinner.



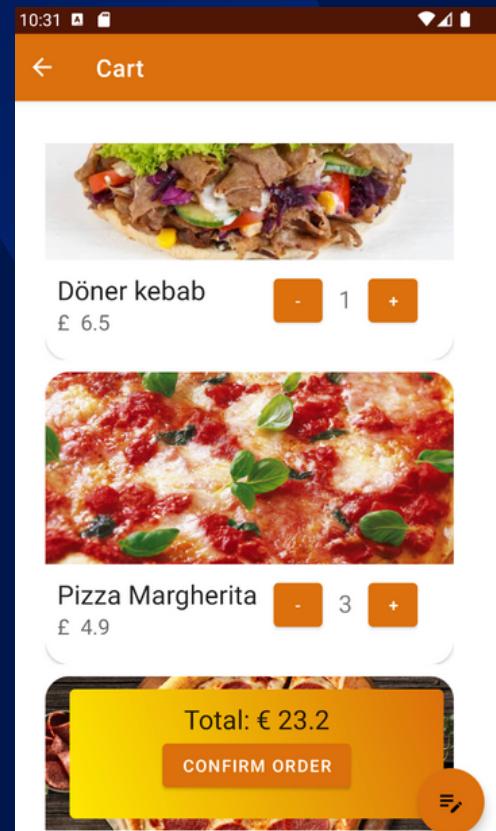
As you can see there are two buttons inside the card, one is plus and other is minus but use to store quantity of food items required. Inside Food Adapter preferences are called and these quantities along with item name is stored in preferences under cart object. and once user clicked cart button user, will go to cart screen and see all the items with quantities and total bill.

For displaying the food items, there is a menu.json file under assets and itemClass is a data class, having a function named getMenuItems to convert json in data class, and that data class is called in homepage and passed to Food adapter as a arrayListOf type list.



Cart Page

Cart page is almost similar to home page, it is having food adapter and food item cards under recycler items, it also have total and confirm order part to move to checkout screen.



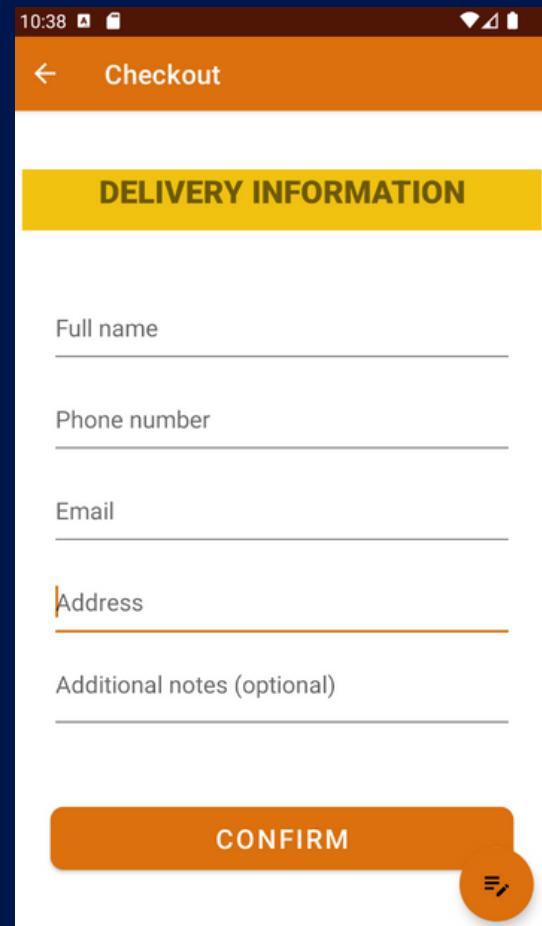
Checkout Screen:

Checkout Screen is initially filled by getting Information from the userInfo under preferences.

After getting the essentials information from the user or from preferences, there is also a option to provide any additional note or instructions that user wants to provide.

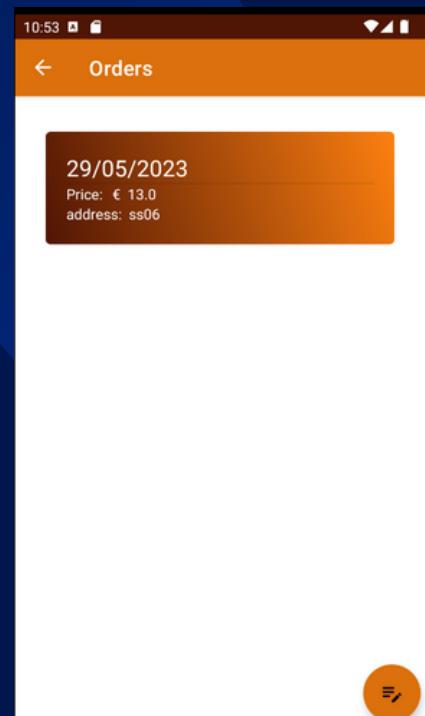
Once user clicks confirm the information will be converted in a hashmap and hashmap also contains the cart items info from user preferences and current date of order. and that hashmap will be passed to firestore orders collection.

Once order data is successfully stored in firestore cart is cleaned and a Toast message "Ordered Successfully" is shown to user and user is navigated back to home fragment (or as mentioned Home Screen).



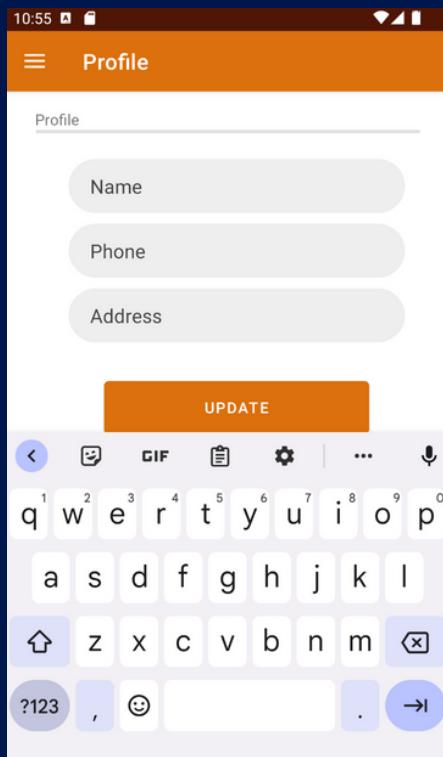
Orders

Orders Fragment contains order history of user, user can see the list of orders he made previously. Recycler view is used here to show the order cards.



Profile Update:

As our email and password are essential sign in credentials so here we use name and Phone Address update in Update screen. Initially these fields are pre-filled by using preferences "userInfo" about you can update them in this screen and click the update button to store the changes.



Once you click the update button, this fields along with user id is converted in a mutable map and update the firestore collection "users" for that id document. if update operation went successfully as Toast message "Updates are saved successfully" will be displayed to user.

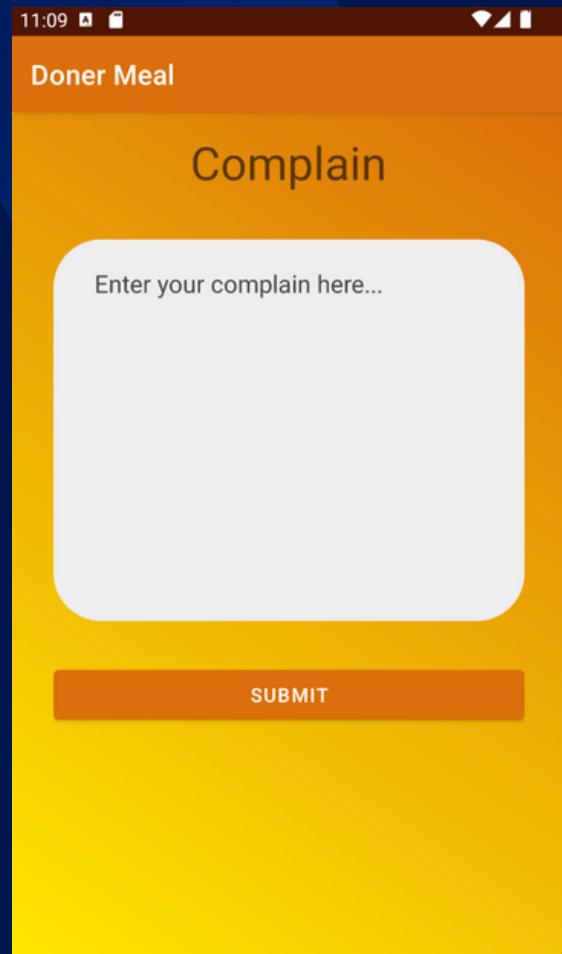
Logout:

Logout is final fragment. It only contains a button and once you click that button preferences are cleared first and then user signout firebase function is called.

Complains

Complains screen is a separate intent for registering the user complains. As you can see user only need to write complain text in the input app collects the other information like user id, and email from user preferences and once user clicks the submit button complain is added in firestore's "complians" collection.

If adding complain went successful, a toast message "Your complain is registered!" will be shown on screen and user will be navigated to home screen.



Gradients

here is the list of gradients maybe useful for you, these are under drawable folder.



luanchscreen.xml



orderlayout.xml



primary_btn.xml



shape.xml