➋PULL&BEAR (Inditex)

It’s an online shopping app which has features of purchasing online,

discovering the latest new arrivals and trends, managing user accounts, etc.

------------My responsibility---------------

1. I was responsible for designing and developing Product Listing module

where products are listed in Grids.

I used Retrofit to fetch data from server

and RecyclerView with GridLayoutManager to show products in Grids.

2. I also worked on showing Pull&Bear stores on Google Maps.

I used custom markers to show stores nearby to the users.

Also I used Fused Location Provider to get users current location.

3. I also integrated Firebase FCM service to send notifications

about new offers, discounts, delivery status etc to registered users.

\*\*\*\*\*\*Additionally\*\*\*\*\*\*\*\*\*\*\*

1. For this app, we followed MVVM architecture.

Initially few modules were developed using Java

and later on few modules developed using Kotlin.

We also used various other libraries such as RxJava, Dagger 2, Retrofit,

DataBinding, Firebase Services, JUnit and Mockito for Unit Testing.

2. We followed Agile methodology

and used Jira for project management and bug reporting.

For code versioning we used Git and Github.

---------------Challenge-----------------------

Integration of FCM was completely new to me.

After following some tutorials and videos about FCM integration,

I was able to successfully integrate FCM to receive push notifications

for offers, order delivery updates, etc.

\*\*\*\*\*Firebase implementation\*\*\*\*

To integrate FCM, first we have to create our project on FCM console

and integrate that project with our Android studio

using Android Studio Assistant for Firebase.

1. Whenever our app runs for first time on users device,

FCM SDK integrated in the app registers itself with FCM server.

2. In response, I received FCM token from the server.

3. I have sent this FCM token received to our backend server.

Backend server maintained FCM token of each user.

4. Backend developers written their logic

to send push message target device.

For this, backend server sent request to FCM server

to deliver push message.

In this request, backend server sent fcm token of user

to whom message to be sent, data, API key.

5. In the app, I also created a service

which extends from FirebaseMessagingService.

In this service, we have overridden onMessageRecieved()

where message is delivered.

In this method, based on the message type, I displayed notification.

➌Dwella (bromin7, Inc)

Dwella app offers a completely new way

of searching and listing No Fee apartments in NYC.

Using this app, users can also upload their properties

which they want to rent.

App also provides feature of virtual tour

where user can take virtual tour of property

to know more about property.

App provides comprehensive analytics about their property

on the dashboard.

App also provides a feature of chat which can be used

by property owners and users interested in that property to discuss.

------------My responsibility---------------

1. For this app, I was responsible for working

on the Authentication module which enables users to sign-in, sign-up,

forget password and reset password.

I used Firebase authorization service for this implementation.

2. I also Worked on the Dashboard module

to show property visit analysis to property owners

using line graphs and information about users who visited the property

by using ACharEngine for charts.

Fetched property analysis data from server using Retrofit and RxJava.

3. I was also responsible for writing unit test cases for various test cases

using JUnit and Mockito

\*\*\*\*\*\*Additionally\*\*\*\*\*\*\*\*\*\*\*

1. We used MVVM architecture for this app and used Kotlin language.

We used RxJava for asynchronous programming,

Retrofit for network calls, Room DB, Firebase Authorization,

Firebase Cloud Messaging, JUnit, Mockito, AChartEngine library

for line graphs.

2. We followed Agile methodology and used Jira

for project management and bug reporting.

For code versioning we used Git and Github.

---------------Challenge-----------------------

As it was my first attempt to integrate Firebase Authorization service

to login using email id or to login with Google account was challenging.

I followed Firebase Authorization documentation to understand

how various authorization works.

I watched some videos on Youtube, also read some articles to learn

how to integrate that.

After couple of days of learning curve,

I was able to implement it successfully.

➍Roomer Travel (Roomer Travel Ltd.)

Users can book their favorite hotels using Roomer Travel app.

Users can search hotels, filter search results,

add hotels to their favorite list, and make payments online.

App also suggest top destinations for users based on their recent tours.

------------My responsibility---------------

1. I was responsible for Implementing the search hotel module

using Volley for API call and used RecyclerView

to show list of search result items.

2. I also worked on the Hotel Details module

by fetching data from the server using Volley

and showed detailed information of the hotel

including hotel images, booking charges, rating, hotel amenities,

showing hotel location on Google Map etc.

3. I integrated Google Map to show nearby hotels

with customized markers to show hotel room pricing

and used Google Fused api to get users current location.

\*\*\*\*\*\*Additionally\*\*\*\*\*\*\*\*\*\*\*

1. I used Java language and followed MVP architecture.

I used Volley for network call, Firebase crashlytics,

Facebook sdk for login with Facebook, Google Maps, Material design.

2. I followed Agile methodology with Jira tool

for project management and bug tracking.

I used git and github for source code versioning.

---------------Challenge-----------------------

Since I was using Google Maps for very first time,

I struggled little bit to setup it for first time.

But main problem was Google Map was not being displayed in Release built.

After an hour of search, I came to know that I have to add SHA-1 key

generated from our Release jks file to Google Developer Console

under Google Maps section.

After adding this SHA-1 key, Google Map was loaded successfully.