GitHub Username: TasneemRK

Rosanne

Description

"Rosanne " is online store for people who don't have enough time to shop or they have events and they don't know what to buy for it. So this store will help them to solve all their problems , save their time and efforts and allow everyone to celebrate with his family and his friends a special gift.
All you have to do is choose the gift and we will deliver it to your home.

Intended User

for people who don't have enough time to shop or they have events and they don't know what to buy for it.

Features

- App is written solely in the Java Programming Language

- ∉ App utilizes stable release versions of all libraries, Gradle, and Android Studio.
- ∉ Specify that App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- ∉ it performs short duration, on-demand requests(such as search), app uses an AsyncTask.
- ∉ Room is used and LiveData and ViewModel are used and no unnecessary calls to the database are made.
- ∉ Show many types of gifts for events to let the user choose what he want.
- ∉ Show some information about every product and let user to write review after login
- ∉ Save products you love in favorite
- ∉ Save products you want to buy in shopping cart and choose the quantity you want
- ∉ Select user's location on google map when you want to buy
- ∉ Show all the orders that you order it before and ability to show every order info.

Versions

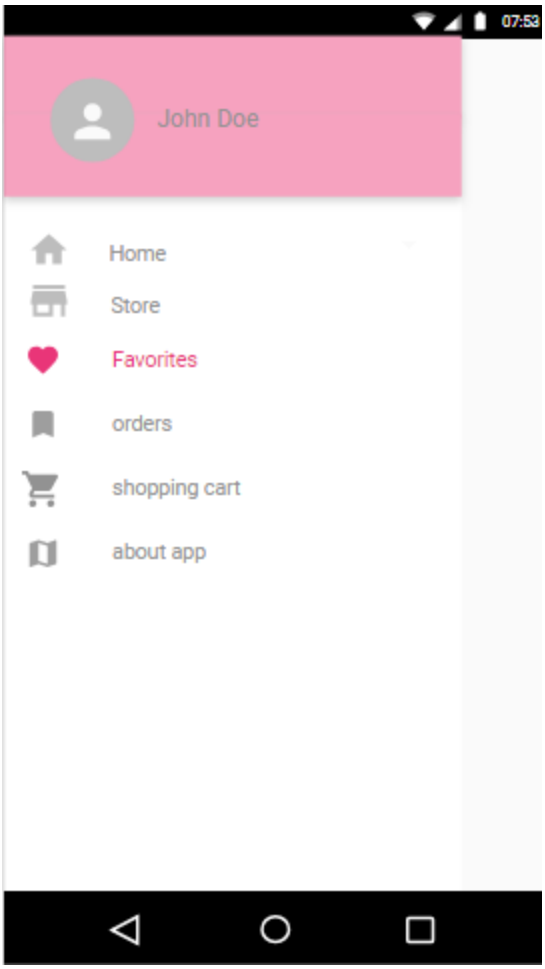
```
implementation fileTree(include: ['*.jar'], dir: 'libs')
implementation 'com.android.support:appcompat-v7:28.0.0'
implementation 'com.android.support.constraint:constraint-layout:1.1.3'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
implementation 'de.hdodenhof:circleimageview:2.2.0'
implementation 'com.android.support:design:28.0.0'
implementation 'com.android.support:recyclerview-v7:28.0.0'
implementation 'com.squareup.picasso:picasso:2.71828'
implementation 'com.android.volley:volley:1.1.1'
implementation 'com.jakewharton:butterknife:9.0.0-rc1'
annotationProcessor 'com.jakewharton:butterknife-compiler:9.0.0-rc1'
implementation 'com.google.code.gson:gson:2.8.5'

def room_version = "1.1.1"

implementation "android.arch.persistence.room:runtime:$room_version"
annotationProcessor "android.arch.persistence.room:compiler:$room_version"

implementation 'android.arch.lifecycle:livedata:1.1.1'

// ViewModel and LiveData
implementation "android.arch.lifecycle:extensions:$room_version"
// alternatively - just ViewModel
  implementation "android.arch.lifecycle:viewmodel:$room_version"
```

User Interface Mocks

Screen 1

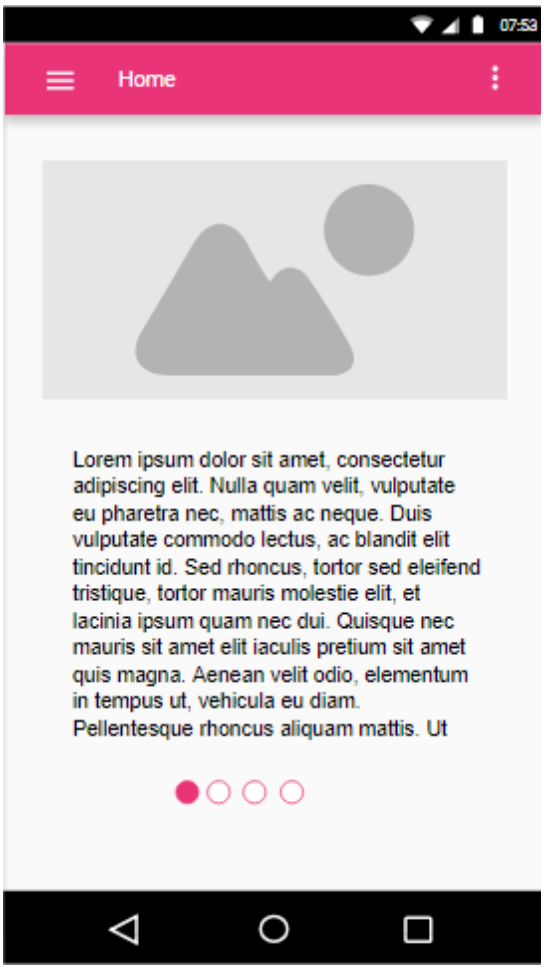
Splash screen : welcome screen.

Screen 2



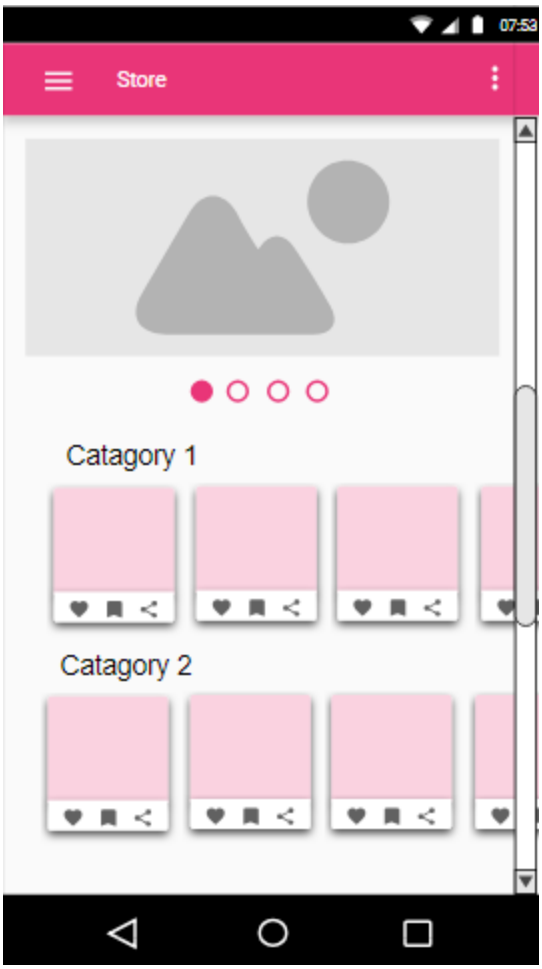
Nav drawer : shows the side menu which contains all the screens in the app.

Screen 3



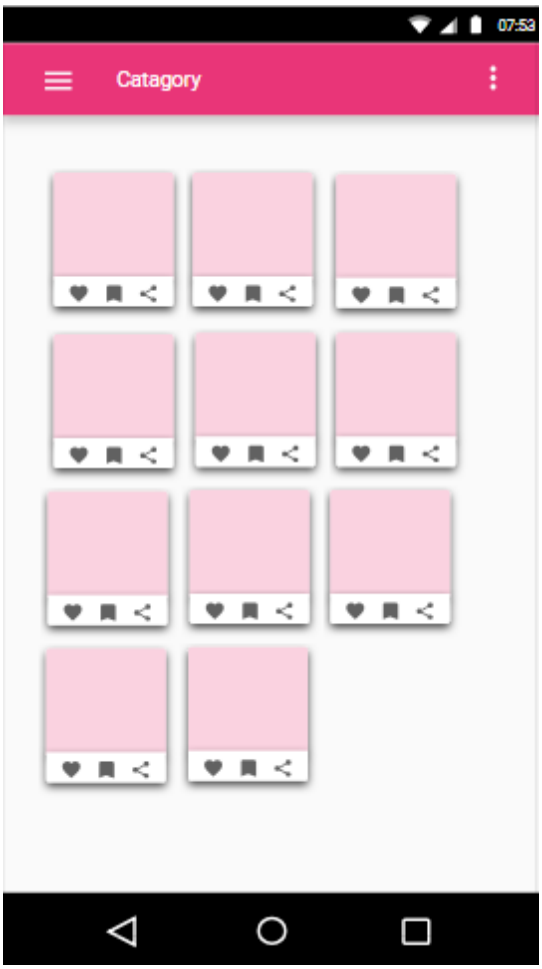
Home screen : shows some advertisements about the latest products in the app.

Screen 4



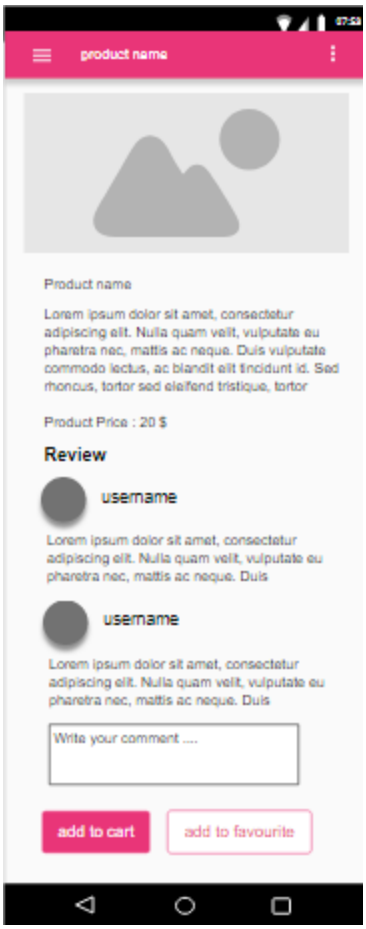
Store screen : shows categories of gifts and flowers and every catagory have the last products follow it.

Screen 5



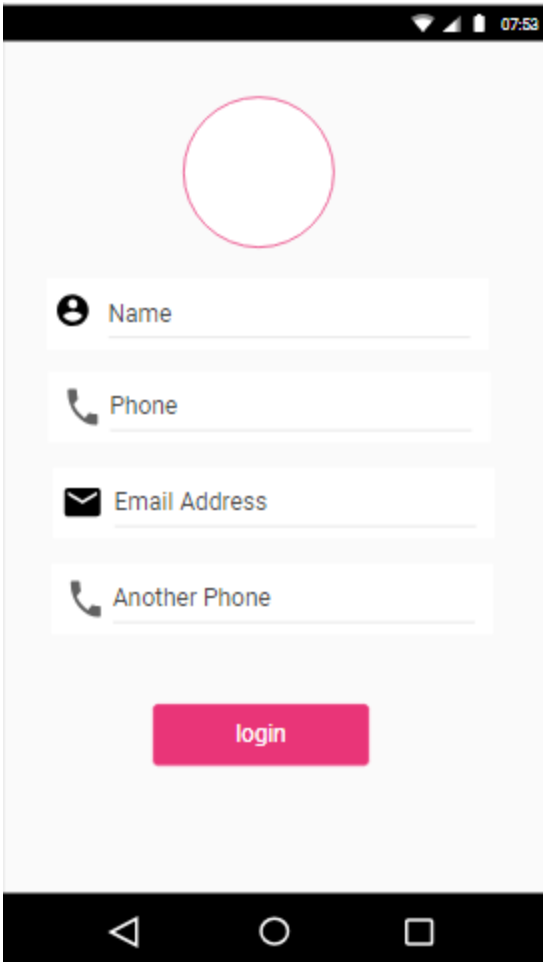
Category products : shows the category's products

Screen 6



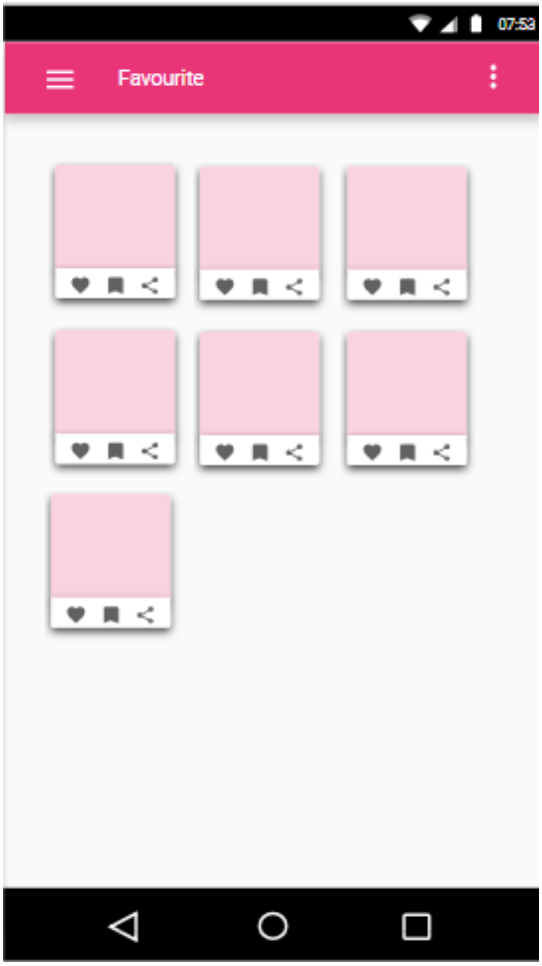
Product info : shows the information about the product and reviews and it allow to user to write his review and two button for adding to card and favorite .

Screen 7



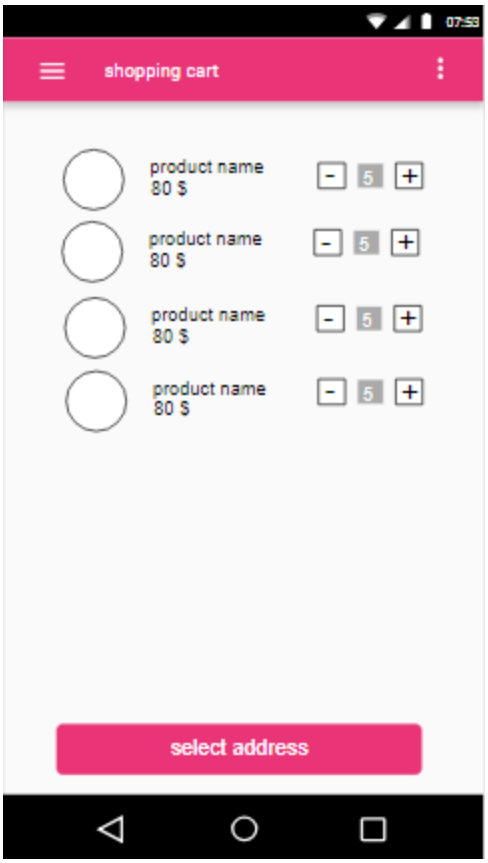
Login : let user write his username , password , phone and photo.

Screen 8



Favorite : shows all the products which the user love it and add it to favorite.

Screen 9

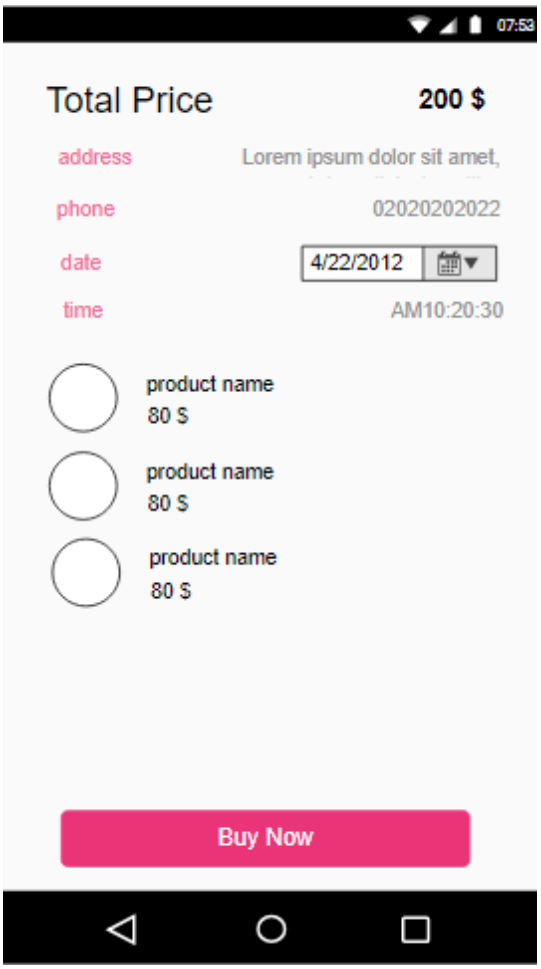


Shopping cart : shows the products which user want to buy it soon or later and he can increase and decrease the quantity as he like before confirm buying.

Screen 10

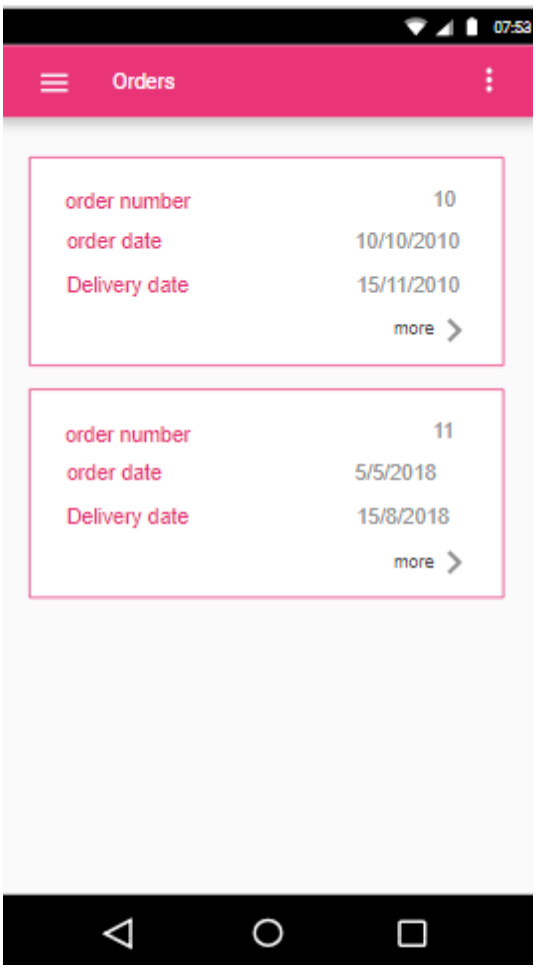
Select location : let the user select his location on google map.

Screen 11



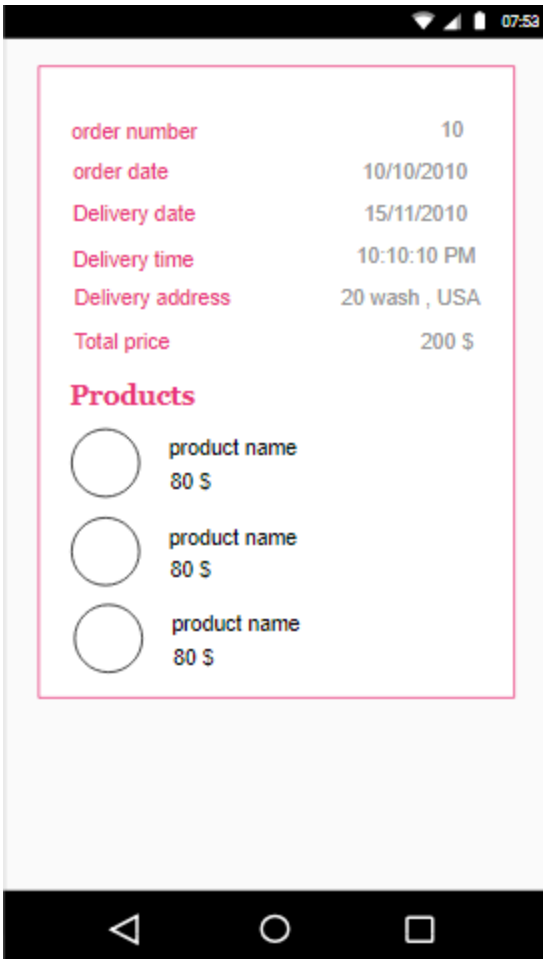
Confirm order : the user fill some data about the delivery to complete it.

Screen 12



Orders : shows all the orders that the user order it before.

Screen 12



Order info : shows all information about order such as time , date , address , ... etc.

Screen 12

Widget : it will show the list of products in the favorite screen.

Key Considerations

How will your app handle data persistence?

Firebase Realtime database and room database.

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

- Picasso for images
- Volley for internet
- Design library
- Room for database
- Firebase libraries
- espresso
- And maybe I will use another libraries through programming the app.

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

- Firebase
- Google Map (places and location)

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

- ∉ Configure libraries
- ∉ Create project at firebase console

Task 2: Implement UI for Each Activity and Fragment

- ∉ Build UI for splash screen
- ∉ Build UI for home activity and add nav drawer
- ∉ Build UI for store activity and its fragments : category products and product info .
- ∉ Build UI for login activity .
- ∉ Build UI for favorite activity .
- ∉ Build UI for shopping cart activity and its fragments : select location and confirm order
- ∉ Build UI for orders activity and order info

Task 3: Analysis building Firebase database

- ∉ Analysis the database for all app
- ∉ Start building some keys and values in firebase
- ∉ Write some data in firebase (this step because there is no control panel so I will fill data in firebase directly)

Task 4: Implementing Google Map Services

- ∉ Create project
- ∉ Get the key and put in app
- ∉ Activate places and location service

Task 5: programming

- ∉ Start writing the code for every activity and connect it with firebase if required

Task 6 : Testing

- ∉ Write or register test for some activates
- ∉ Test all app (run app on real device and with espresso)

Task 7 : Add widget on the home screen for the app

- ∉ Create require layouts for widget
- ∉ Write the suitable code to implement the widget in app (provider , service ,..etc)