

GitHub Username: Ushnesha

Jewelevation

Description

This app is an online jewellery shopping app that allows its user to browse a plenty of jewellery items. Now-a-days, online shopping has become a trend and thus an app that helps customers to buy jewellery online aids to the trend. This app will help its customer to choose and adds the jewellery on the wishlist from where they can buy it later.

Intended User

Anyone who is interested in buying jewellery online either for self purpose or intended to gift someone, this app will help them to buy their choiced jewellery while staying at their home.

Features

Following are the main features of the app:

- 1.Login/Signup facility
- 2.Fetch images of the specific jewelleries from server
- 3.Choiced jewellery can be added to the wishlist
- 4.The user can sign out whenever they want
- 5.The user can change its profile setting as needed
- 6.It will include good material design
- 7.Customer can search for the jewellery whenever they want

This app uses java as its backend programming language.

Versions:

Android Studio: 3.1.3

Gradle: 4.4

Dependencies:(later other dependencies may also be added)

implementation 'com.android.support:appcompat-v7:27.1.1'

implementation 'com.android.support.constraint:constraint-layout:1.1.2'

testImplementation 'junit:junit:4.12'

implementation 'com.squareup.okhttp3:okhttp:3.1.2'

implementation 'com.google.firebase:firebase-auth:16.0.3'

implementation 'com.google.firebase:firebase-database:16.0.1'

```
implementation 'com.google.android.gms:play-services-auth:16.0.0'
implementation 'com.google.firebase:firebase-core:16.0.1'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
implementation 'com.android.support:design:27.1.1'
implementation 'com.android.support:gridlayout-v7:27.1.1'
implementation 'com.facebook.fresco:fresco:0.7.0'
implementation 'com.android.support:cardview-v7:27.1.1'
implementation 'com.squareup.picasso:picasso:2.5.2'
```

User Interface mocks

Screen 1: Its contains the signup Activity



A mockup of a mobile application screen for a signup activity. At the top, there is a square placeholder with a diagonal 'X' inside. Below this, there are three text input fields labeled 'Username', 'Email', and 'Password'. Under the 'Password' field is a 'REGISTER' button. Below the button is a link that says 'Forgot Password?'. At the bottom, there is a link that says 'Already registered? Login here' which is highlighted with a blue rectangular border.

Screen 2: LogIn Activity

Email

Password

LogIn

Forgot Password?

Not registered? Sign up here!

Screen 3: Navigation drawer Activity

< Jewelelection

Label
email@example.com

Rings

Earrings

Pendant

Bangles & Bracelets

Nosepin

Necklace

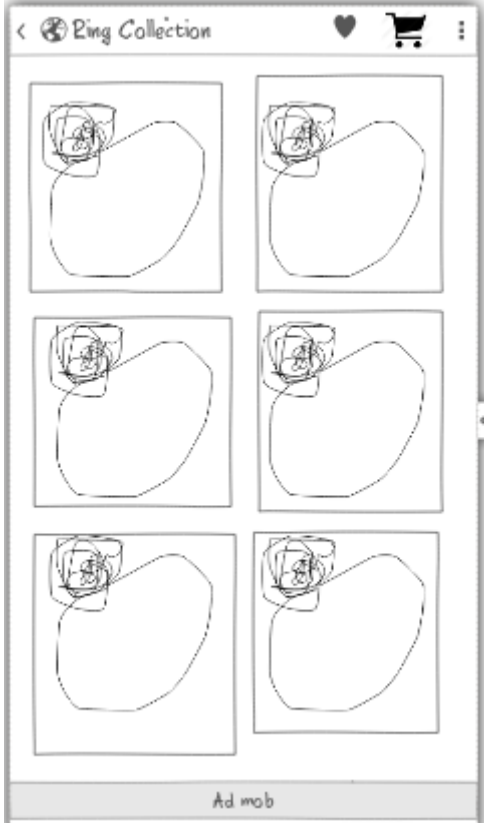
Chain

Hangalsutra

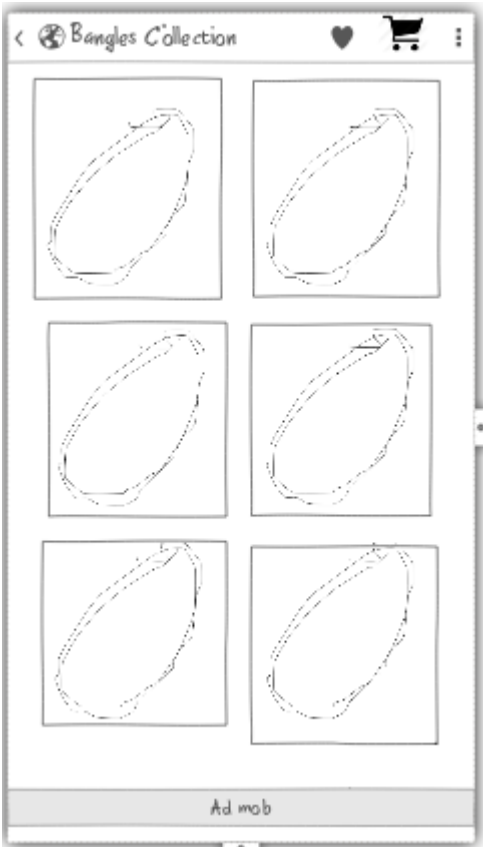
Screen 4: HomeActivity



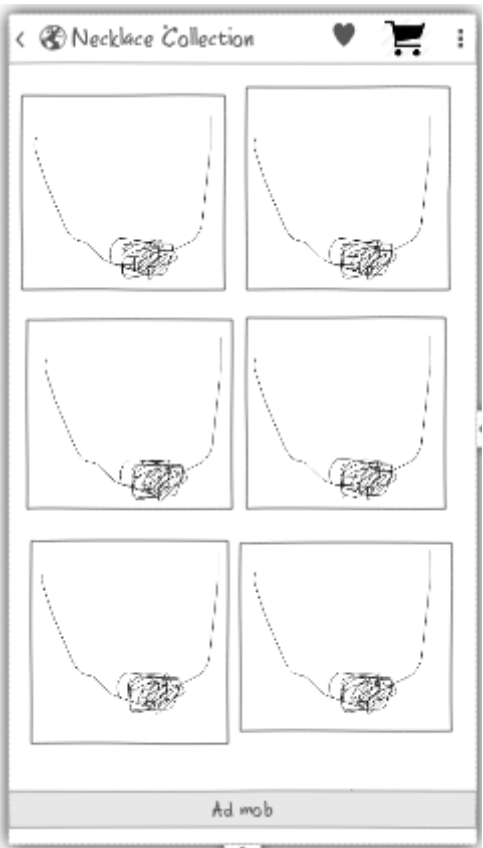
Screen 5: Ring Collection Activity



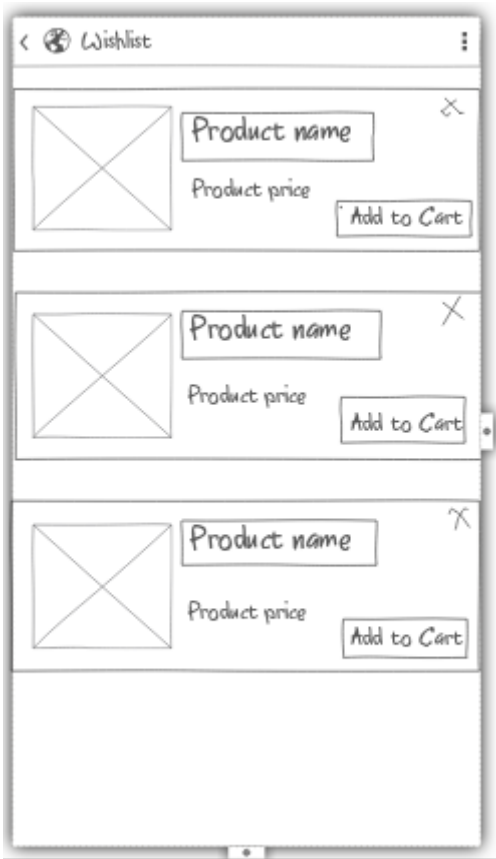
Screen 6: Bangle Collection Activity



Screen 7: Necklace Activity



Screen 8: Wishlist Activity



Screen 9: Profile Activity

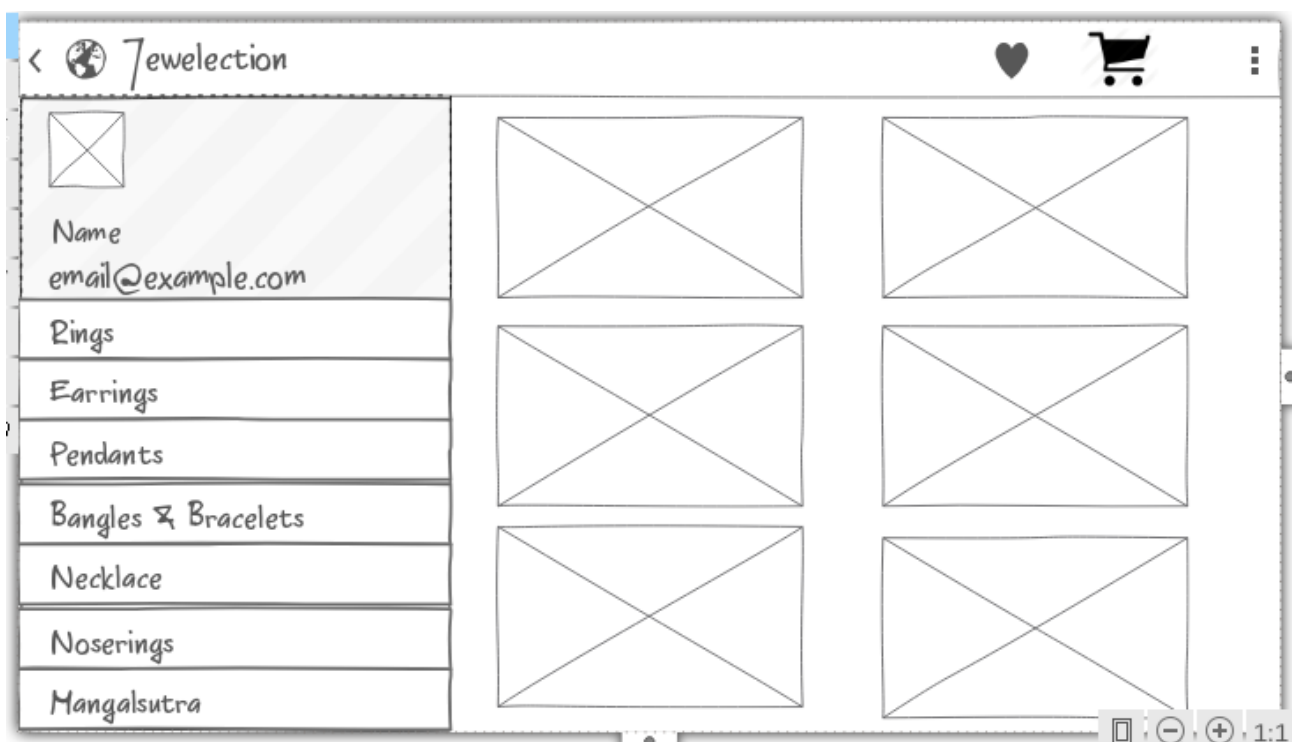


Screen 10: Reset Password Activity



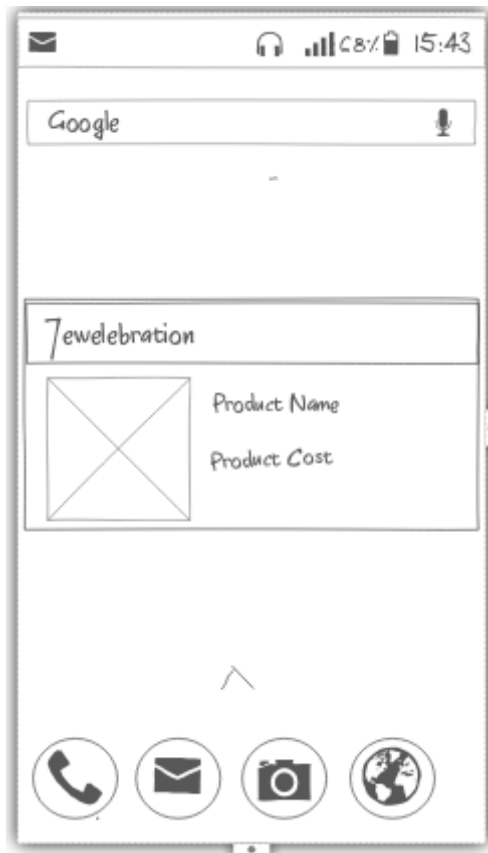
A vertical screen for password reset. At the top is a square icon with an 'X'. Below it is the title "Forgot Password?". A message states: "We just need your registered email id to sent you password reset instructions." There is an input field labeled "Email". Below the input field is a button labeled "RESET". At the bottom is a link labeled "<<Back".

Screen 11: Home Activity in landscape orientation



A horizontal screen for a jewelry store. The top bar contains a back arrow, a globe icon, the text "Jewelelection", a heart icon, a shopping cart icon, and a menu icon. On the left is a sidebar with a profile icon and the text "Name email@example.com". Below this is a list of jewelry categories: "Rings", "Earrings", "Pendants", "Bangles & Bracelets", "Necklace", "Noserings", and "Mangalsutra". The main area displays a grid of 8 placeholder images (squares with an 'X'). At the bottom right are icons for mobile, zoom in, zoom out, and a scale of 1:1.

Screen 12: Home screen with widget that shows a random product from the app



Key Considerations

How will your app handle data persistence?

Data Persistence: Data Persistence will be done with the help of SQLite Database which will be implemented with the help of content provider. It will also use a realtime remote firebase database for logging and signup processes and help the user to be kept logged in. The content provider will help to keep the images of different jewellery types in different tables locally.

The app will also uses android app architecture of Model View ViewModel which helps in data persistence between the UI.

Data Loading: The app uses Loaders to load the data from the local database with the help of the content provider.

Intent Service: The app uses an Intent Service to receive the users current location.

Describe any edge or corner cases in the UX.

When the user opens the home activity or any jewellery specific activity, the app sends an intent for launching these activities with startService function. The images are fetched from the server and updated into the local database on call upon the Intent Service class. These updated images are set to the UI using different libraries like picasso or glide.

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

Picasso: The app uses picasso to loading and caching of images into the home and specific jewelleries activity.

Firebase Realtime Database: The app will use firebase database to store user details in cloud that are sent from signup and login activities.

Fresco: It is used for image loading from the network, store them locally and display a placeholder until the image has arrived.

OkHttp3: For allowing all requests from the server and response to them.

Butterknife: This library will help for data/method binding that prevents the use of findViewById function.

Timber: For logging purpose.

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

Google Play Service : For using AdMob and Firebase Realtime Database.

Project Tasks:

Task 1: Project Setup

- 1) Configure all the libraries required for building the app in app-->build.gradle.
- 2) Define the target and min SDK required to build the app.
- 3) Define all the permissions required for the app in the manifest file.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.BROADCAST_STICKY" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_INTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- 4) Create java packages for defining classes Database, Model, View, ViewModel, Network connection, Adapters, Activity, Fragments and Interfaces.
- 5) Create dimens, drawable folder under res folders for defining dimensions, and drawable images. Vector assets for turning svg files into vector drawables along with styles normal and v24 for defining styles on api level below 24 and 24 above respectively.
- 6) Create initialization classes for databases and network activity.

Task 2: Implement UI for Each Activity and Fragment

- 1) Build UI for Signup Activity with a sign-in authentication button done with the help of firebase database.
- 2) Build UI for SignIn Activity that will check if the user has given correct credentials and then open to the HomeActivity.
- 3) Build UI for ResetPassword Activity that allow the user to reset their password incase they have forgotten their password through their email

id.

- 4) Build UI for HomeActivity with a custom appbar and a tablayout with the help of a navigation drawer.
- 5) Build UI for activities of different jewellerys each associated with fragments that helps the activities to load and set the images that are downloaded from the server.
- 6) Build UI for ProfileActivity where the user can change its profile settings and from where they can log out.
- 7) Build UI for WishlistActivity where users can save the product they have chosen and buy them later.

Task 3: Style the App

- 1) Define primary and secondary colors to be used in the app along with all the other colors in the colors.xml.
- 2) Define the default dimension sizes in the dimens.xml
- 3) Define the styles.xml with NoActionBar Theme along with the items and styles.
- 4) Define the icons to be used in the app under res->drawable.
- 5) Define the strings to be used in the app in res->values->strings.xml

Task 4: Build Reset Password through email id

- 1) Build the Reset Password through email id with the help of firebase reset password method.
- 2) Connect app to firebase through the google-services.json and google play services library.
- 3) Send reset password instructions to the email id through which the user has logged in.
- 4) On resetting the password, changes will be saved to the firebase

database.

Task 5: Build detail fragments for each of the jewellery specific activity

- 1) Use a content provider for accessing all the jewellery images on the users phone and fetch from the server the list of all the images from json url.
- 2) Display the images inside a recyclerView with good material design.

Task 6: Build Paid/Free Flavors

- 1) Setup gradle dependencies
- 2) Break up project into paid and free module
- 3) Remove ads from paid version

Task 7: Implement Widget on Home Screen

- 1) Add Widget on home screen for product view to the user
- 2) By clicking the item on the widget, the user can see the details of the product type.

Task 8: Testing and Debugging

- 1) Design Test case for valid data insertion operation in database.
- 2) Analyze the bugs.
- 3) Performance Testing.
- 4) UI test for Phone vs Tablet.

Task 9: Other Tasks

- 1) RTL support
- 2) Accessibility support