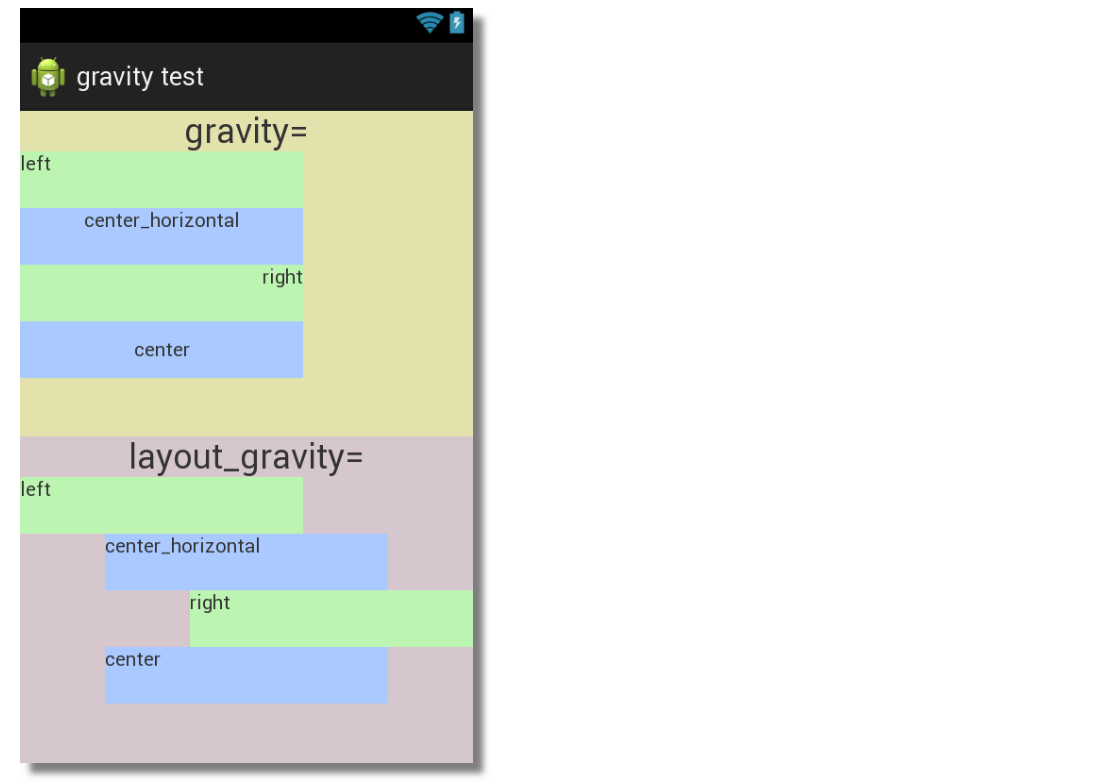
**Apomee eCommerce Beta(Doc)**

1. **Connecting with Firebase**: Uses androidX artifacts. Start with empty activity.

Created firebase project with hash no(Gradle>app…) and kept the Json file in the app directory of the project. Added dependency on gradle app/project for Firebase connection. Created a local database in firebase.

1. **Creating main page:** Along background image, two buttons for log in and sign up which are created by using resource file and used as background for those buttons. Layout->drawable…. Logo in the upper site along the slogan.
2. **Creating login page:** Added dependencies to import material designs. For checkbox style: style="@style/Material.Drawable.CheckBox"

Cbd\_strokecolor from <https://github.com/rey5137/Material/>, the dependencies for this was imported in this step.. Linear layout used within a relative layout. android:visibility="invisible" to hide a textview.



4) **Creating Registration Page:** Created registration page in almost the same manner as login in page.

5) **Registration Activity connecting with FireBase: Elements:**

\*loadingbar = new ProgressDialog(this); //Declaring dialog bar. loadingbar is a random name>>

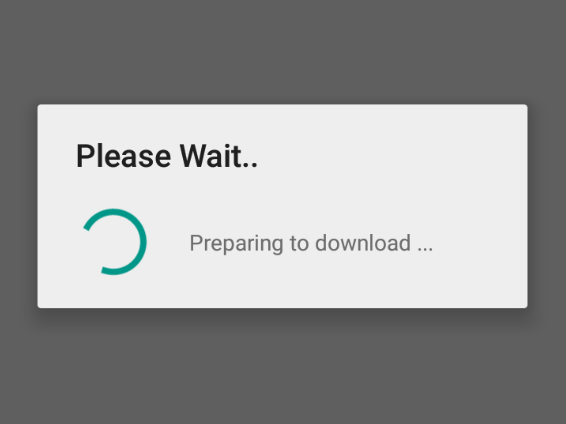
loadingbar.setTitle("Create Account"); *//Set loading bar title*

loadingbar.setMessage("Please wait, information are being checked");  *//set inside message of the loading bar*

loadingbar.setCanceledOnTouchOutside(false); *//terminates the loading bar if it is touched outside(true/false)*

loadingbar.show(); *//shows the loading bar*

loadingbar.dismiss() *//stops the loading bar*

**

String name= InputName.getText().toString();  *//converts the inputname variable after getting the text by getText and converts to string via toString.*

TextUtils.isEmpty….. //Textutils is imported first from library. Can be used to check string attributes/whether empty or not and so on…

DatabaseReference XXXX; /*/A* ***DatabaseReference*** *represents a specific location in database and can be used for reading or writing data to that database location.*

FirebaseDatabase.getInstance().getReference(); //*Getting root reference from the database.*

*\*\*\*\*You can get an instance by calling* [*getInstance()*](https://firebase.google.com/docs/reference/android/com/google/firebase/database/FirebaseDatabase.html#getInstance())*. To access a location in the database and read or write data, use* [*getReference()*](https://firebase.google.com/docs/reference/android/com/google/firebase/database/FirebaseDatabase.html#getReference())

addListenerForSingleValueEvent :This listener helps you to detect the change in the data at a particular path (including the children at that particular path). This listener listens exactly once and provides you the data as per the change until this event was triggered.

\*\*public void onDataChange(@NonNull DataSnapshot dataSnapshot)

*A DataSnapshot instance contains data from a Firebase Database location. Any time you read Database data, you receive the data as a DataSnapshot.*

\*\*HashMap<String,Object> userrdataMap= new HashMap<>();

*Java HashMap is a* [*hash table*](https://en.wikipedia.org/wiki/Hash_table) *based implementation of Java’s Map interface. Map is a collection of key-value pairs. It maps keys to values.*

\*\*public abstract void onComplete (Task<TResult> task)

*Called when the Task completes.*

**6) Login Activity:** Created a class named User, used constructor and getter and setter methods under, declared from LoginActivity under parentdb in order to fetch data from firebase and set to the user class. And checks whether input matches with the database..

**7) Remember Me checkbox:** If once logged in with remember me checkedbox clicked, then the phone and password is saved on Android Memory, credits to [*https://github.com/pilgr/Paper*](https://github.com/pilgr/Paper)*.* Paper is a library to write data easily on the Android Memory.

Next time while opening the app, app checks if the data is saved using remember me. If so, then the data is checked with the server data and logged in without password or id.

**8) Admin panel:** User can login as admin from login page clicking on the I’m admin. That button will change the parentdb=”Admins”, default value of the parentdb is “Users”. So, from the firebase, Admins login data will be handled and matched with the user credentials. This little tweak to refine admins from the users.

**9) Admin Add Product Activity :** Added New category and product adding activity. Used image view and intent initially.

Afterwards:

1. Added option for admin to upload new products.
2. Products information and image can be added by admin. Images will be stored to Firebase using Firebase Storage. And image URLs are stored in firebase database in product information folder, so we can fetch URLs to show product picture in future.
3. Product key uses unique id. Generated by using date time stamp along jpg

**10) User Home :** Activity with Navigation drawer along with some options along with logo. Added username and user picture. *Picture used circleview library dependencies.*

**11) User Home Products/settings/logout:**

\*Viewholder: Created viewholder in package named viewholder to hold the product information.

\*Products: class in Model to construct and get,set the product information.

\*Picasso: to fetch image and show along the product names and descriptions.

holder.txtProductName.setText(model.getPname());

holder.txtProductDescription.setText(model.getDescription());

holder.txtProductPrice.setText("Price = " + model.getPrice() + "BDT");

Picasso.get().load(model.getImage()).into(holder.imageView)

\*User profile: User profile designed only. Navigation Drawer can be used to select settings and logout.

\*products\_layout.xml : Holds the view of Recycleview in the user home.

**Home Activity features many layouts xml files in menu and layout//:::DONE:::**

**12)Navigation DRawer view user Settings:**

* User can change his order number, not the phone number which was used as user id in the database.
* User can add address and update name
* User can upload images using firebase storage. Images are uploaded in firebase storage *Profile pictures*  folder. And image cropper available, used dependencies: [**https://github.com/ArthurHub/Android-Image-Cropper**](https://github.com/ArthurHub/Android-Image-Cropper)
* Editing settings pre information only available if the image is already uploaded, otherwise not .

**13) CartActivity:** Added one more Recycleview to show added product on the cart. Database child created on the firebase as Cart List to hold the added items.

*Used Android Arsenal Elegent number button.*

**14) Confirm Cart:** On pressing next, the price of the total products are shown

And, on click confirm the details information for shipment is asked. Upon filling up the data are saved on *‘Orders’* in the database folder.

**Need to handle Ordered items and removed item from admin view cart list.//::DONE::**

**15) Order Placed:** if the orderd once placed, user cant reorder new order, handled via if else on the product details activity. And, shipment status is introducted. User can see the status of the shipment from his cart. But, user cant add new product till the shipment is done or order placed.

**Need to allow user to create multiple orders..//::Need time to implement::**

*Order can be seen by admin using recyclerview. And, he can confirm it. on confirmation order will be removed from cart list and from order list. though user orders will be saved to the OrdersDB only with the price for future references.*