

Android

Project overview

Georgios Papalexandrakis, Nikolaos Litsas, Nikolaos Magoulis, Panagiotis Varouktsis

Introduction

What is Android Studio?

“Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA.”

by Android

What is a project in Android Studio?

“A project in Android Studio contains everything that defines your workspace for an app, from source code and assets, to test code and build configurations.”

by Android

What is a module in Android Studio?

“A module in Android Studio is a collection of source files and build settings that allow you to divide your project into discrete units of functionality. Your project can have one or many modules and one module may use another module as a dependency. Each module can be independently built, tested, and debugged.”

by Android

Different types of modules

- Android app module
- Dynamic feature module
- Library module
- Google Cloud module

Different types of Android app modules

- Phone & Tablet Module
- Wear OS Module
- Android TV Module
- Google Cloud Module

Example

GitHub

Project's structure (0)

- Gradle Scripts (defines build configuration)
- Modules

Project's structure (1)

in example

master ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

| Author | Commit Message | Date |
|-------------|-------------------|---|
| pvarouktsis | Update README.md | d57d1ca 7 days ago 101 commits |
| | .idea | Implement expandable products 16 days ago |
| | .resources | Created .resources, add brand logo pic 8 days ago |
| | app | Implement functions to check if fields are empty 7 days ago |
| | gradle/wrapper | Initialize project 2 months ago |
| | .gitignore | Remove google services.json 8 days ago |
| | README.md | Update README.md 7 days ago |
| | TODO.md | Merge branch 'dev' of https://github.com/pvarouktsis/ShopS... 8 days ago |
| | build.gradle | Implement search (not full-text search) 14 days ago |
| | gradle.properties | Initialize project 2 months ago |
| | gradlew | Initialize project 2 months ago |
| | gradlew.bat | Initialize project 2 months ago |
| | settings.gradle | Initialize project 2 months ago |

contains “app” module

contains “Gradle Scripts”

Module's structure (0)

- **build/** (contains build outputs)
- **libs/** (contains private libraries)
- **src/** (contains code and resources)
- **build.gradle** (defines module's build configuration)

Module's structure (1)

in example

master SF-assignments / app /

pvarouktsis Implement functions to check if fields are empty 1467968 8 days ago History

..

src Implement functions to check if fields are empty 8 days ago

.gitignore Initialize project 2 months ago

build.gradle Implement search (not full-text search) 15 days ago

proguard-rules.pro Initialize project 2 months ago

contains source code

defines build configuration

| File | Description | Last Commit |
|--------------------|--|--------------|
| src | Implement functions to check if fields are empty | 8 days ago |
| .gitignore | Initialize project | 2 months ago |
| build.gradle | Implement search (not full-text search) | 15 days ago |
| proguard-rules.pro | Initialize project | 2 months ago |

src/ (0)

- **androidTest/** (contains tests that run on an Android device)
- **main/** (contains the “main” SourceSet)
- **test/** (contains tests that run on your host JVM)

src/ (1) in example

master ▾ SF-assignments / app / src / Go to file Add file ▾

| | | | | | |
|--|--|---------|------------|---|--------------|
|  pvarouktsis | Implement functions to check if fields are empty | 1467968 | 8 days ago |  | History |
| ... | | | | | |
|  androidTest/java/com/example/shopsmart | Initialize project | | | | 2 months ago |
|  main | Implement functions to check if fields are empty | | | | 8 days ago |
|  test/java/com/example/shopsmart | Initialize project | | | | 2 months ago |

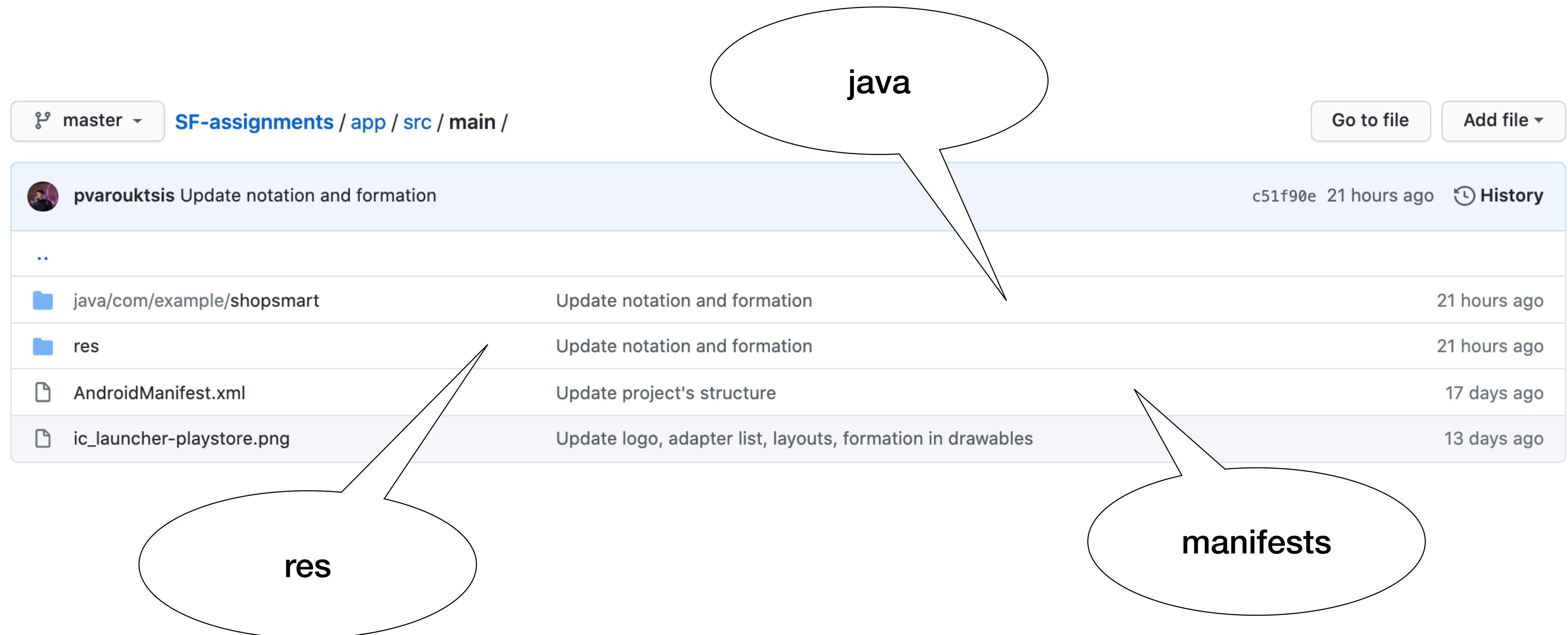
contains the app's
“main” source code

Android app structure (0)

- **manifests** (contains essential information about the application)
- **res** (contains resources such as drawables, layouts, values, etc.)
- **java** (contains Java source code)

Android app structure (1)

in example

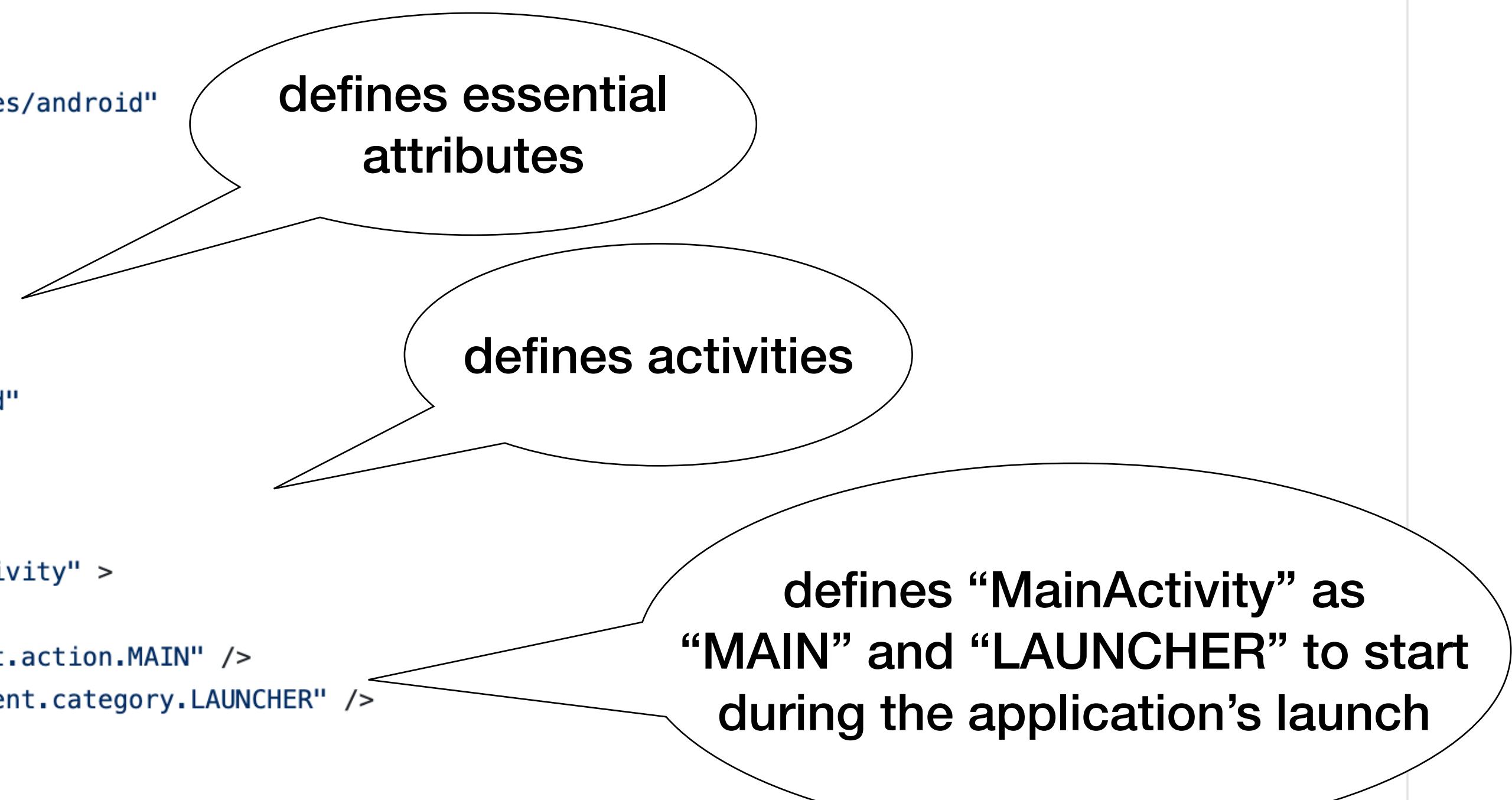


manifests

AndroidManifest.xml

```
24 lines (22 sloc) | 894 Bytes
Raw Blame   
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     package="com.example.shopsmart">
5
6     <application
7         android:allowBackup="true"
8         android:icon="@mipmap/ic_launcher"
9         android:label="@string/app_name"
10        android:roundIcon="@mipmap/ic_launcher_round"
11        android:supportsRtl="true"
12        android:theme="@style/ShopSmartTheme" >
13         <activity
14             android:name=".views.activities.MainActivity" >
15             <intent-filter>
16                 <action android:name="android.intent.action.MAIN" />
17                 <category android:name="android.intent.category.LAUNCHER" />
18             </intent-filter>
19         </activity>
20         <activity android:name=".views.activities.LoginActivity" />
21         <activity android:name=".views.activities.RegisterActivity" />
22     </application>
23
24 </manifest>
```



defines essential attributes

defines activities

defines “MainActivity” as “MAIN” and “LAUNCHER” to start during the application’s launch

Before analyzing res and java...

What is res?

“The res directory contains resources that are separated into further more packages such as drawable, layout, values and other. Resources define static re-usable content that is used by the application (e.g. drawables, colors, strings, etc.). More specifically defines from values (e.g. colors, strings, etc.) to bitmaps (e.g. vectors) and layouts that determine the UI of activities, fragments and other components.”

What is java?

“The java directory contains the Java source code which can be separated into further more packages. Java source code, can access and modify the resources and their behavior depending on specific events and implement their operations via listeners (e.g. create a View.OnClickListener object for the “btn_add” of the “layout_product” which will perform the operation of adding the requested product to the user’s cart).”

The concept is similar to HTML and JavaScript

res (0)

- **drawable/** (defines the graphics that can be drawn to the screen
e.g. a vector, a layer-list, a shape, etc.)
- **layout/** (defines the UI of an activity, fragment or other component)
- **menu/** (defines the application menu)
- **values/** (defines styles, colors, strings etc.)
- ...

res (1)

in example

| SF-assignments / app / src / main / res / | | |
|---|--|--|
|  pvarouktsis | Update notation and formation | c51f90e 21 hours ago  |
| .. | | |
|  drawable | Update notation and formation | 21 hours ago |
|  layout | Edit frame of no products | 11 days ago |
|  menu | Implement edit account functionality | 14 days ago |
|  mipmap-anydpi-v26 | Update logo, adapter list, layouts, formation in drawables | |
|  mipmap-hdpi | Update logo, adapter list, layouts, formation in drawables | |
|  mipmap-mdpi | Update logo, adapter list, layouts, formation in drawables | |
|  mipmap-xhdpi | Update logo, adapter list, layouts, formation in drawables | |
|  mipmap-xxhdpi | Update logo, adapter list, layouts, formation in drawables | 13 days ago |
|  mipmap-xxxhdpi | Update logo, adapter list, layouts, formation in drawables | 13 days ago |
|  values | Update cursor | 11 days ago |

contains vectors, layer-lists, etc.

contains the UI of activities, fragments and other components

contains definitions of styles, strings, colors, etc. that are used in the application

drawable/ (0)

in example

master ▾ SF-assignments / app / src / main / res / drawable /

Go to file Add file ▾

pvarouktsis Update cursor 1996ec8 10 days ago History

..

| File | Description | Last Commit |
|----------------------------|--|-------------|
| ic_down_arrow.xml | Update logo, adapter list, layouts, formation in drawables | 10 days ago |
| ic_home.xml | Update logo, adapter list, layouts, formation in drawables | 12 days ago |
| ic_launcher_foreground.xml | Update logo, adapter list, layouts, formation in drawables | 10 days ago |
| ic_logo.xml | Update cursor | 12 days ago |
| ic_logo_cart.xml | Update logo, adapter list, layouts, formation in drawables | 10 days ago |
| ic_logo_line.xml | Update cursor | 12 days ago |
| ic_my_account.xml | Update logo, adapter list, layouts, formation in drawables | 12 days ago |
| ic_my_cart.xml | Update logo, adapter list, layouts, formation in drawables | 12 days ago |
| ic_on_sale.xml | Update logo, adapter list, layouts, formation in drawables | 12 days ago |
| ic_search.xml | Update logo, adapter list, layouts, formation in drawables | 12 days ago |
| ll_active.xml | Update cursor | 10 days ago |
| ll_add_button.xml | Update cursor | 10 days ago |
| ll_bottom_navbar.xml | Update cursor | 10 days ago |
| ll_inactive.xml | Update cursor | 10 days ago |
| ll_remove_button.xml | Update cursor | 10 days ago |

icons that are used in the application

other layer-lists or shapes that are used in the application

drawable/ (1)

ll_active.xml

24 lines (24 sloc) | 798 Bytes

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <layer-list xmlns:android="http://schemas.android.com/apk/res/android">
3     <!-- shadow -->
4     <item android:top="5dp">
5         <shape android:shape="rectangle">
6             <!-- size relative to actual -->
7             <size
8                 android:width="260dp"
9                 android:height="60dp" />
10            <solid android:color="@color/shadow" />
11            <corners android:radius="30dp" />
12        </shape>
13    </item>
14    <!-- top -->
15    <item android:bottom="5dp">
16        <shape android:shape="rectangle">
17            <size
18                android:width="260dp"
19                android:height="60dp" />
20                <solid android:color="@color/white" />
21                <corners android:radius="30dp" />
22            </shape>
23        </item>
24    </layer-list>
```

defines a frame that can be re-used
in the application (e.g. as a background of a
Button, an EditText, etc.)

is already defined in
values/colors.xml

layout/ (0) in example

master SF-assignments / app / src / main / res / layout / Add file History

pvarouktsis Edit frame of no products

..

| File | Description | Last Commit |
|--------------------------------|---|-------------|
| activity_login.xml | Update cursor | 11 days ago |
| activity_main.xml | Update notation, layouts, fragments, activities and adapter | 15 days ago |
| activity_register.xml | Update cursor | 11 days ago |
| fragment_home.xml | Update cursor | 11 days ago |
| fragment_my_account.xml | Update textSize and edit welcome message | |
| fragment_my_account_delete.xml | Update cursor | |
| fragment_my_account_edit.xml | Update cursor | |
| fragment_my_cart.xml | Update home, on sale and my cart fragments | 13 days ago |
| fragment_on_sale.xml | Implement appropriate messages for firebase exceptions | 11 days ago |
| fragment_product_list.xml | Edit frame of no products | 11 days ago |
| layout_product.xml | Implement edit account functionality | |
| layout_shopsmart_logo.xml | Update logo, adapter list, layouts, formation in drawables | |

defines the UI of activities

defines the UI of fragments

defines the UI of other components

layout/ (1)

activity_main.xml

28 lines (25 sloc) | 1.06 KB

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/rl_main"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:background="@color/aqua"
9   tools:context=".views.activities.MainActivity">
10
11   <!-- frame layout -->
12   <FrameLayout
13     android:id="@+id/fl_main_container"
14     android:layout_width="match_parent"
15     android:layout_height="match_parent" />
16
17   <!-- bottom navbar -->
18   <com.google.android.material.bottomnavigation.BottomNavigationView
19     android:id="@+id/bottom_navbar_view"
20     android:layout_width="match_parent"
21     android:layout_height="60dp"
22     android:layout_alignParentTop="false"
23     android:layout_alignParentBottom="true"
24     android:background="@drawable/ll_bottom_navbar"
25     app:itemTextColor="@color/aqua"
26     app:menu="@menu/menu_bottom_navbar" />
27
28 </RelativeLayout>
```

The diagram illustrates the layout structure defined in `activity_main.xml`. It features a large cyan-colored `FrameLayout` with a red border, which contains a white `BottomNavigationView` with a green border. The `BottomNavigationView` has four items: a house icon labeled "Home", a shopping cart icon, a gear icon, and a person icon. A callout bubble labeled "FrameLayout" points to the top of the cyan area. Another callout bubble labeled "BottomNavigationView" points to the bottom of the cyan area. A third callout bubble labeled "FrameLayout is extended behind BNV" points to the right edge of the cyan area, indicating that the `FrameLayout` extends beyond the boundaries of the `BottomNavigationView`.

layout/ (2) fragment_home.xml

```
89 lines (79 sloc) | 3.32 KB
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/r_l_home_container"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@color/transparent"
8     android:gravity="center"
9     android:clickable="true"
10    android:focusable="true"
11    android:focusableInTouchMode="true"
12    tools:context=".views.activities.MainActivity">
13
14    <RelativeLayout
15        android:id="@+id/r_l_search_container"
16        android:layout_width="wrap_content"
17        android:layout_height="wrap_content"
18        android:gravity="center"
19        android:paddingStart="20dp"
20        android:paddingLeft="20dp"
21        android:paddingTop="20dp"
22        android:paddingEnd="20dp"
23        android:paddingRight="20dp"
24        android:paddingBottom="140dp">
25
26        <RelativeLayout
27            android:id="@+id/r_l_shopsmart_logo"
28            android:layout_width="wrap_content"
29            android:layout_height="wrap_content"
30            android:layout_centerHorizontal="true"
31            android:layout_centerVertical="false"
32            android:gravity="center">
33
34            <include layout="@layout/layout_shopsmart_logo" />
35
36        </RelativeLayout>
37
38        <RelativeLayout
39            android:id="@+id/r_l_search_form"
40            android:layout_width="360dp"
41            android:layout_height="70dp"
42            android:layout_below="@+id/r_l_shopsmart_logo"
43            android:layout_centerInParent="true"
44            android:layout_marginTop="20dp"
45            android:layout_marginBottom="0dp"
46            android:gravity="center">
47
48            <EditText
49                android:id="@+id/et_search"
50                android:layout_width="match_parent"
51                android:layout_height="match_parent"
52                android:background="@drawable/ll_active"
53                android:ems="10"
54                android:gravity="center_vertical"
55                android:hint="@string/text_product_search"
56                android:inputType="textFilter"
57                android:paddingStart="40dp"
58                android:paddingLeft="40dp"
59                android:paddingTop="0dp"
60                android:paddingEnd="80dp"
61                android:paddingRight="80dp"
62                android:paddingBottom="5dp"
63                android:textSize="18sp" />
64
65            <ImageButton
66                android:id="@+id/ibtn_search"
67                android:layout_width="70dp"
68                android:layout_height="70dp"
69                android:layout_alignTop="@+id/et_search"
70                android:layout_alignBottom="@+id/et_search"
71                android:layout_alignParentStart="false"
72                android:layout_alignParentLeft="false"
73                android:layout_alignParentEnd="true"
74                android:layout_alignParentRight="true"
75                android:background="@color/transparent"
76                android:paddingStart="20dp"
77                android:paddingLeft="20dp"
78                android:paddingTop="20dp"
79                android:paddingEnd="20dp"
80                android:paddingRight="20dp"
81                android:paddingBottom="22dp"
82                android:scaleType="centerInside"
83                android:src="@drawable/ic_search" />
84
85        </RelativeLayout>
86
87    </RelativeLayout>
88
89 </RelativeLayout>
```



“rl_search_form”

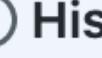
“rl_shopsmart_logo”

menu/ (0)

in example

master ▾ SF-assignments / app / src / main / res / menu /

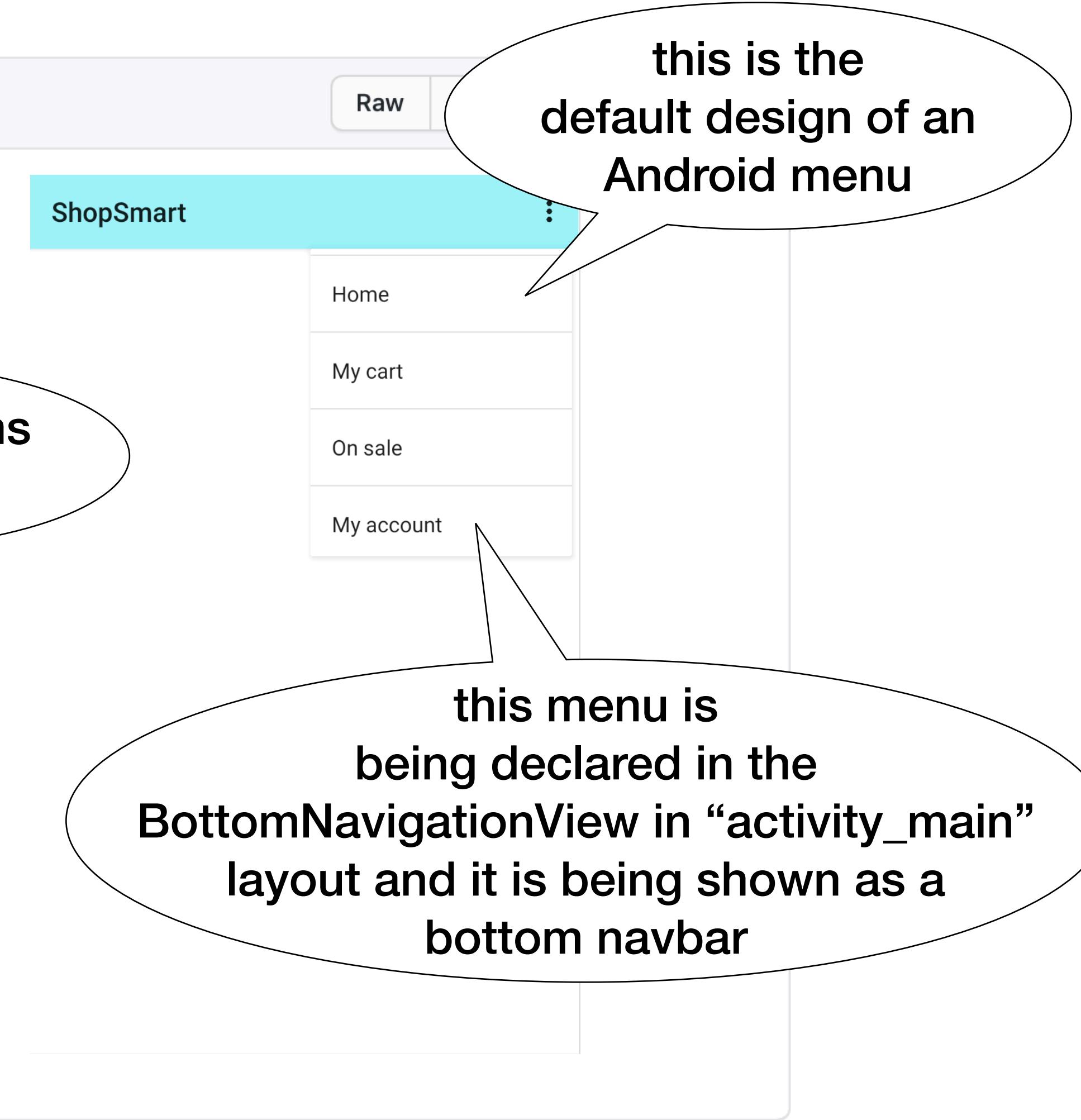
Go to file Add file ▾

| | | | |
|--|--------------------------------------|---------------------|---|
|  pvarouktsis | Implement edit account functionality | 299fed6 13 days ago |  |
| .. | | | History |
|  menu_bottom_navbar.xml | Implement edit account functionality | 13 days ago | |

menu/ (1)

menu_bottom_navbar.xml

```
24 lines (19 sloc) | 706 Bytes  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">  
3  
4     <item  
5         android:id="@+id/navigation_home"  
6         android:icon="@drawable/ic_home"  
7         android:title="@string/text_home" />  
8  
9     <item  
10        android:id="@+id/navigation_my_order"  
11        android:icon="@drawable/ic_my_cart"  
12        android:title="@string/text_my_cart" />  
13  
14     <item  
15        android:id="@+id/navigation_on_sale"  
16        android:icon="@drawable/ic_on_sale"  
17        android:title="@string/text_on_sale" />  
18  
19     <item  
20        android:id="@+id/navigation_my_account"  
21        android:icon="@drawable/ic_my_account"  
22        android:title="@string/text_my_account" />  
23  
24 </menu>
```



values/ (0)

in example

The screenshot shows a GitHub repository interface for the 'SF-assignments' project. The path 'SF-assignments / app / src / main / res / values /' is selected in the sidebar. The repository has one commit from 'pvarouktsis' titled 'Update cursor'. The commit details show it was pushed 10 days ago. Below the commit, there are four files listed:

- colors.xml**: Update activities, fragments, adapters. Last modified 11 days ago.
- ic_launcher_background.xml**: ~~Update logo, adapter list, layouts, formation in drawables~~. Last modified 12 days ago.
- strings.xml**: Update textSize and edit welcome message. Last modified 10 days ago.
- styles.xml**: Update cursor. Last modified 10 days ago.

Annotations with callouts point to specific files:

- An annotation labeled "defining colors used by the application" points to the **colors.xml** file.
- An annotation labeled "defining strings used by the application" points to the **strings.xml** file.
- An annotation labeled "defining styles used by the application" points to the **styles.xml** file.

values/ (1) strings.xml

30 lines (30 sloc) | 1.53 KB

Raw Blame   

```
1 <resources>
2     <!-- app's name -->
3     <string name="app_name">ShopSmart</string>
4     <!-- texts -->
5     <string name="text_default">None</string>
6     <string name="text_home">Home</string>
7     <string name="text_my_cart">My cart</string>
8     <string name="text_on_sale">On sale</string>
9     <string name="text_my_account">My account</string>
10    <string name="text_user_username">Username</string>
11    <string name="text_user_email">Email</string>
12    <string name="text_user_password">Password</string>
13    <string name="text_user_login">Sign in</string>
14    <string name="text_user_logout">Sign out</string>
15    <string name="text_user_register">Sign up</string>
16    <string name="text_user_update_short">Update</string>
17    <string name="text_user_update_long">Update your account</string>
18    <string name="text_user_delete_short">Delete</string>
19    <string name="text_user_delete_long">Delete your account</string>
20    <string name="text_product_search">Search</string>
21    <string name="text_product_filter">Filter</string>
22    <string name="text_product_name">Product_name</string>
23    <string name="text_product_price">Product_price</string>
24    <string name="text_product_supermarket">Product_supermarket</string>
25    <string name="text_product_button_add">Add</string>
26    <string name="text_product_button_add_extended">Add product to your cart</string>
27    <string name="text_product_button_remove">Remove</string>
28    <string name="text_product_none">There are no products</string>
29    <string name="text.Please_wait">Please wait...</string>
30 </resources>
```

38 <RelativeLayout
39 android:id="@+id/rl_search_form"
40 android:layout_width="360dp"
41 android:layout_height="70dp"
42 android:layout_below="@+id/r_l_shopsmart_logo"
43 android:layout_centerInParent="true"
44 android:layout_marginTop="20dp"
45 android:layout_marginBottom="0dp"
46 android:gravity="center">
47
48 <EditText
49 android:id="@+id/et_search"
50 android:layout_width="match_parent"
51 android:layout_height="match_parent"
52 android:background="@drawable/ll_active"
53 android:ems="10"
54 android:gravity="center_vertical"
55 android:hint="@string/text_product_search"
56 android:inputType="textFilter"
57 android:paddingStart="40dp"
58 android:paddingLeft="40dp"
59 android:paddingTop="0dp"
60 android:paddingEnd="80dp"
61 android:paddingRight="80dp"
62 android:paddingBottom="5dp"
63 android:textSize="18sp" />
64
65 <ImageButton
66 android:id="@+id/ibtn_search"
67 android:layout_width="70dp"
68 android:layout_height="70dp"
69 android:layout_alignTop="@+id/et_search"
70 android:layout_alignBottom="@+id/et_search"
71 android:layout_alignParentStart="false"
72 android:layout_alignParentLeft="false"
73 android:layout_alignParentEnd="true"
74 android:layout_alignParentRight="true"
75 android:background="@color/transparent"
76 android:paddingStart="20dp"
77 android:paddingLeft="20dp"
78 android:paddingTop="20dp"
79 android:paddingEnd="20dp"
80 android:paddingRight="20dp"
81 android:paddingBottom="22dp"
82 android:scaleType="centerInside"
83 android:src="@drawable/ic_search" />
84
85 </RelativeLayout>

“text_product_search”

fragment_home.xml

java (0)

“The java directory does not have a standard structure. It depends on the developer what model and patterns will be applied. There are different approaches about the java directory architecture (e.g. Model-View-Presenter).”

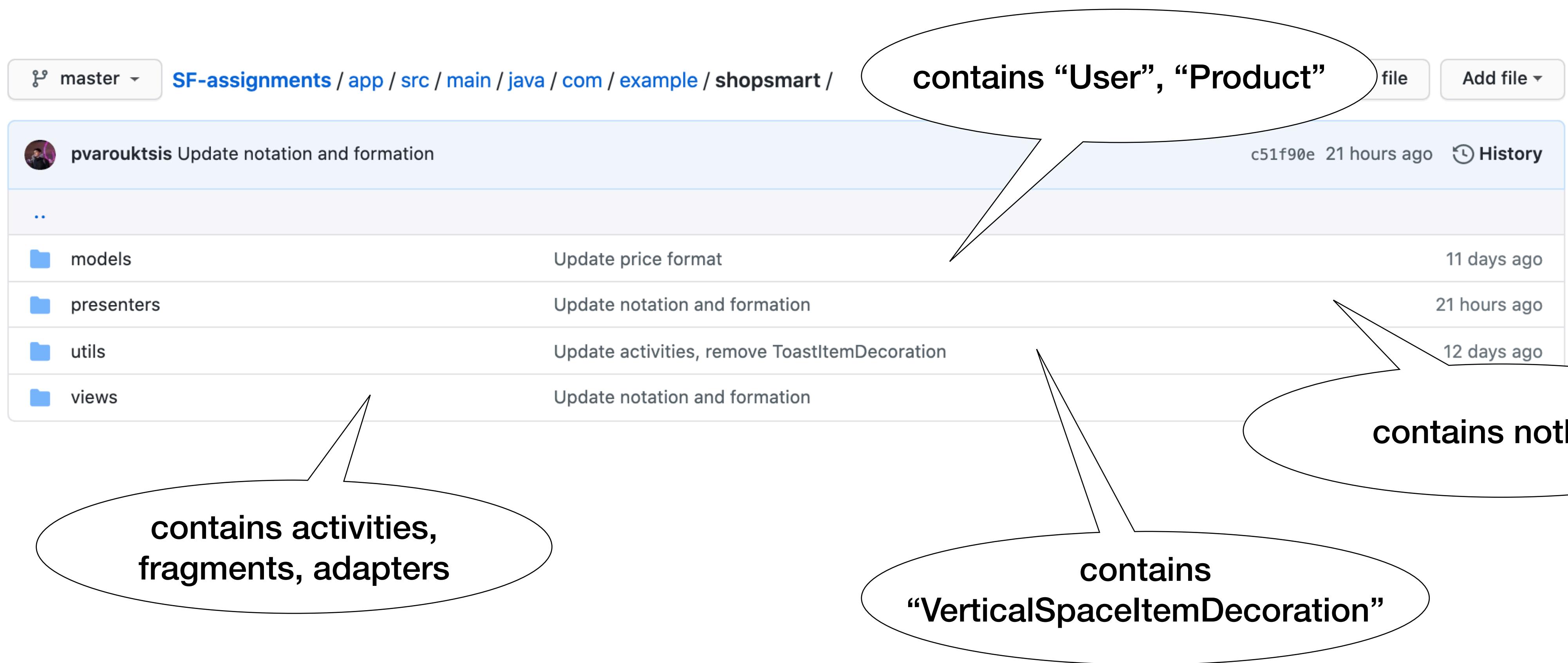
java (1)

in example

- **models/** (contains the data models e.g. User, Product)
- **views/** (contains the views e.g. activities, fragments, adapters)
- **presenters/** (contains nothing but it should accommodate an intermediate layer between models and views for retrieving, managing and showing data but also update the view on specific events via listeners)
- **utils/** (contains other “helper” classes e.g. VerticalSpaceItemDecoration)

java (2)

shopsmart/



models/ (0)

master ▾ SF-assignments / app / src / main / java / com / example / shopsmart / models /

Go to file Add file ▾

 pvarouktsis Update price format History

..

| File | Description | Time |
|--------------|--|-------------|
| Product.java | Update price format | 12 days ago |
| User.java | Update logo, adapter list, layouts, formation in drawables | 14 days ago |

contains “User”, “Product”

models/ (1)

User.java

84 lines (66 sloc) | 1.77 KB

Raw Blame   

```
1 package com.example.shopsmart.models;
2
3 import com.google.firebaseio.Exclude;
4
5 import java.io.Serializable;
6 import java.util.ArrayList;
7
8 public class User implements Serializable {
9     @Exclude
10    private String id;
11    @Exclude
12    private String username;
13    @Exclude
14    private String email;
15    @Exclude
16    private String password;
17    private ArrayList<Product> cart;
18
19    public User() {
20        // Required
21    }
22
23    public User(String email, String password) {
24        this.email = email;
25        this.password = password;
26        this.cart = new ArrayList<>();
27    }
28
29    public User(String username, String email, String password) {
30        this.username = username;
31        this.email = email;
32        this.password = password;
33        this.cart = new ArrayList<>();
34    }
35
36    public User(String id, String username, String email, String password) {
37        this.id = id;
38        this.username = username;
39        this.email = email;
40        this.password = password;
41        this.cart = new ArrayList<>();
42    }
43
44    @Exclude
45    public String getId() { return id; }
46
47    @Exclude
48    public void setId(String id) { this.id = id; }
49
50    @Exclude
51    public String getUsername() {
52        return username;
53    }
54
55    @Exclude
56    public void setUsername(String username) {
57        this.username = username;
58    }
59
```

many fields are excluded from Firebase Firestore

many constructors to initialize each time different fields of the object

presenters/

master ▾ SF-assignments / app / src / main / java / com / example / shopsmart / presenters / Go to file Add file ▾

 pvarouktsis Update notation and formation c51f90e 2 days ago ⏲ History

..

 .keep Update notation and formation 2 days ago

contains an empty “.keep”
file so the “presenters” directory can be
uploaded in GitHub

utils/ (0)

master ▾ SF-assignments / app / src / main / java / com / example / shopsmart / utils /

Go to file Add file ▾

| | | |
|---|----------------------------|---|
|  pvarouktsis Update activities, remove ToastItemDecoration | c739535 13 days ago |  |
| .. | | |
|  VerticalSpaceItemDecoration.java | Update project's structure | 18 days ago |

defines the vertical
space of two items and more
specifically of two consecutively
products in the adapter list

utils/ (1)

VerticalSpaceItemDecorator.java

22 lines (16 sloc) | 664 Bytes

Raw Blame   

```
1 package com.example.shopsmart.utils;
2
3 import android.graphics.Rect;
4 import android.view.View;
5
6 import androidx.recyclerview.widget.RecyclerView;
7
8 public class VerticalSpaceItemDecoration extends RecyclerView.ItemDecoration {
9     //private static final String TAG = "VERTICAL_SPACE_ID";
10    private final int verticalSpaceHeight;
11
12    public VerticalSpaceItemDecoration(int verticalSpaceHeight) {
13        this.verticalSpaceHeight = verticalSpaceHeight;
14    }
15
16    @Override
17    public void getItemOffsets(Rect outRect, View view, RecyclerView parent,
18                               RecyclerView.State state) {
19        outRect.bottom = verticalSpaceHeight;
20    }
21
22 }
```

extends RecyclerView.ItemDecoration

Before analyzing views/...

What is Intent?

“Intent is an object that defines a message to request an action from another app component, either a specific component (explicit intent) or a specific type of component (implicit intent).”

What is Context?

“Context is an object and is being used as an interface to global information about an application status/environment. It is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.”

What is Activity?

“Activity is an object (derives from Context) that provides a UI with which users can interact. Activity is a crucial component of the Android. Apps following the programming paradigm are launched with a main method but Android apps are able to be launched with a variety of Activities (e.g. launch gmail from the app menu or launch gmail from the notification bar). In other words, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.”

What is Fragment?

“Fragment is an object and represents a behavior or a part of the UI in an Activity. It is always hosted in an Activity and the Fragment’s lifecycle is directly affected by the Activity’s lifecycle. An Activity can accommodate multiple Fragments and a Fragment can be re-used in multiple Activities. In other words, each Fragment is a modular section of an Activity, which has its own lifecycle, receives its own input events, and which it can be added or removed while the Activity is running.”

What is RecyclerView?

“RecyclerView is an object that displays a scrolling list of elements. It is a more advanced and flexible version of [ListView](#) and it is recommended on large data sets or data that frequently change.”

What is RecyclerView.Adapter?

“RecyclerView.Adapter provides a binding from an app-specific data set to views via a ViewHolder, which describes an item’s view and metadata. These views are then displayed within the RecyclerView.”

views/ (0)

- **activities/** (contains the activities of the application)
- **fragments/** (contains the fragments of the application)
- **adapters/** (contains the adapters of the application)

views/ (1)

| master | | | SF-assignments / app / src / main / java / com / example / shopsmart / views / | Go to file | Add file ▾ |
|---|---|--|--|---|------------|
|  pvarouktsis | Update notation and formation | | c51f90e 2 days ago |  History | |
| .. | | | | | |
|  activities | Update notation and formation | | 2 days ago | | |
|  adapters | Update message add product and showToast function in adapters | | 12 days ago | | |
|  fragments | Update notation and formation | | 2 days ago | | |

activities/ (0)

| SF-assignments / app / src / main / java / com / example / shopsmart / views / activities / | | | Go to file | Add file ▾ |
|---|-------------------------------|---------|------------|---|
|  pvarouktsis | Update notation and formation | c51f90e | 2 days ago |  |
| .. | | | | |
|  AbstractActivity.java | Update notation and formation | | 2 days ago | |
|  LoginActivity.java | Update notation and formation | | 2 days ago | |
|  MainActivity.java | Update notation and formation | | 2 days ago | |
|  RegisterActivity.java | Update notation and formation | | 2 days ago | |

implements an abstraction

activities/ (1)

AbstractActivity.java

implementing methods that are inherited by the subclasses (e.g. goToMainActivity(), add() and replace() fragments, etc.)

implementing goToMainActivity() by creating a new Intent and passing the “MainActivity” to be displayed on the screen

implementing replace() by creating a FragmentManager and a FragmentTransaction and replacing the “fl_main_container” of the “activity_main” layout with the new fragment

```
16  public abstract class AbstractActivity extends AppCompatActivity {  
17      protected static final String TAG = "ACTIVITY";  
18      protected ProgressDialog progressDialog;  
19  
20      protected abstract void initializeUIComponents();  
21  
22      protected void goToMainActivity(AbstractActivity aa) {  
23          Log.d(TAG, "goToMainActivity: called");  
24          startActivity(new Intent(aa, MainActivity.class));  
25          finish();  
26      }  
27  
28      protected void goToLoginActivity(AbstractActivity aa) {  
29          Log.d(TAG, "goToLoginActivity: called");  
30          startActivity(new Intent(aa, LoginActivity.class));  
31          finish();  
32      }  
33  
34      protected void goToRegisterActivity(AbstractActivity aa) {  
35          Log.d(TAG, "goToRegisterActivity: called");  
36          startActivity(new Intent(aa, RegisterActivity.class));  
37          finish();  
38      }  
39  
40      protected void add(int currentID, Fragment fragment) {  
41          Log.d(TAG, "replace: called");  
42          FragmentManager fm = getSupportFragmentManager();  
43          FragmentTransaction ft = fm.beginTransaction();  
44          ft.add(currentID, fragment).commit();  
45      }  
46  
47      protected void replace(int currentID, Fragment fragment) {  
48          Log.d(TAG, "replace: called");  
49          FragmentManager fm = getSupportFragmentManager();  
50          FragmentTransaction ft = fm.beginTransaction();  
51          ft.replace(currentID, fragment).commit();  
52      }  
53  
54      protected void showLoading(AbstractActivity aa) {  
55          Log.d(TAG, "showLoading: called");  
56          if (progressDialog == null) {  
57              progressDialog = new ProgressDialog(aa);  
58          }  
59          progressDialog.setMessage(getString(R.string.text_please_wait));  
60          progressDialog.setCancelable(false);  
61          progressDialog.show();  
62      }
```

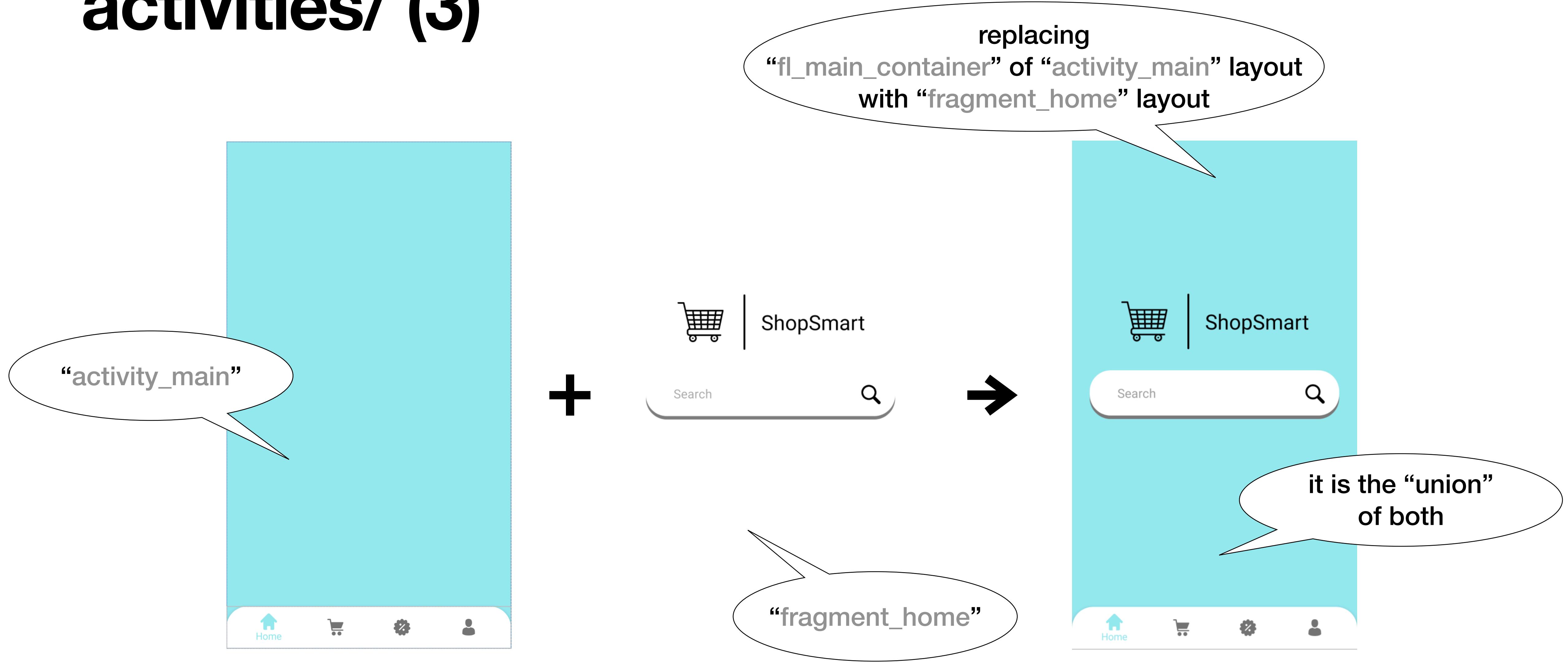
activities/ (2) MainActivity.java

```
19 public class MainActivity extends AbstractActivity {  
20     protected static final String TAG = "MAIN_A";  
21     protected FirebaseAuth fa;  
22  
23     @Override  
24     protected void onCreate(Bundle savedInstanceState) {  
25         Log.d(TAG, "onCreate: called");  
26  
27         // initialize ui  
28         super.onCreate(savedInstanceState);  
29         setContentView(R.layout.activity_main);  
30         initializeUIComponents();  
31         initializeFragment();  
32  
33         // initialize firebase  
34         fa = FirebaseAuth.getInstance();  
35     }  
36  
37     @Override  
38     protected void onStart() {  
39         Log.d(TAG, "onStart: called");  
40         super.onStart();  
41         FirebaseUser fu = fa.getCurrentUser();  
42         updateUI(fu);  
43     }  
44  
45     protected void initializeUIComponents() {  
46         Log.d(TAG, "initializeUIComponents: called");  
47  
48         // initialize bottom navigation bar  
49         BottomNavigationView bnv = findViewById(R.id.bottom_navbar_view);  
50         bnv.setOnNavigationItemSelectedListener(navigationListener);  
51     }  
52  
53     protected void initializeFragment() {  
54         Log.d(TAG, "initializeFragment: called");  
55  
56         // initialize fragment  
57         Fragment f = new HomeFragment();  
58         replace(R.id.fl_main_container, f);  
59     }  
60  
61     protected void updateUI(FirebaseUser fu) {  
62         Log.d(TAG, "updateUI: called");  
63         if (fu != null) {  
64             showToast(MainActivity.this, "Welcome " + fu.getDisplayName());  
65         } else {  
66             goToLoginActivity(MainActivity.this);  
67         }  
68     }  
69  
70     protected BottomNavigationView.OnNavigationItemSelectedListener navigationListener =  
71         new BottomNavigationView.OnNavigationItemSelectedListener() {  
72             Fragment f = null;  
73  
74             @Override  
75             public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {  
76                 Log.d(TAG, "navigationListener: called");  
77                 switch (menuItem.getItemId()) {  
78                     case R.id.navigation_home:  
79                         f = new HomeFragment();  
80                         break;  
81                     case R.id.navigation_my_order:  
82                         f = new MyCartFragment();  
83                         break;  
84                     case R.id.navigation_on_sale:  
85                         f = new OnSaleFragment();  
86                         break;  
87                     case R.id.navigation_my_account:  
88                         f = new MyAccountFragment();  
89                         break;  
90                     default:  
91                         return false;  
92                 }  
93                 replace(R.id.fl_main_container, f);  
94                 return true;  
95             }  
96         };  
97     }  
98 }
```

initializing the view
with the “activity_main” layout,
initializing the bottom navbar and replacing
“fl_main_container” with the
“HomeFragment”

implementing the
listener for the bottom navbar (e.g.
if “on sale” menu item selected replace
the current fragment with the
“OnSaleFragment”)

activities/ (3)



fragments/ (0)

| SF-assignments / app / src / main / java / com / example / shopsmart / views / fragments / | | |
|--|-------------------------------|------------|
| | | Go to file |
| | | Add file ▾ |
|  pvarouktsis | Update notation and formation | |
| .. | | |
|  AbstractFragment.java | Update notation and formation | 2 days ago |
|  HomeFragment.java | Update notation and formation | 2 days ago |
|  MyAccountDeleteFragment.java | Update notation and formation | 2 days ago |
|  MyAccountEditFragment.java | Update notation and formation | |
|  MyAccountFragment.java | Update notation and formation | |
|  MyCartFragment.java | Update notation and formation | |
|  OnSaleFragment.java | Update notation and formation | |
|  ProductListFragment.java | Update notation and formation | 2 days ago |

implementing an abstraction

every single one of these
fragments is being hosted by the
“MainActivity” and each one of them when called
are replacing the “fl_main_container”

on why implementing so many
fragments and only one activity that hosts all of them, it is because the
main layout (e.g. the bottom navbar) is not being replaced or modified, so by altering
the “fl_main_container” with the requested fragments is a better practice but, also, is
more efficiently because fragments are enough more lightweight than
activities

implementing methods that are inherited by the subclasses like add(), replace(), showLoading(), dismissLoading() etc.

fragments/ (1)

AbstractFragment.java

implementing showToast() method that shows a temporary message at the bottom of the screen

```
16 public abstract class AbstractFragment extends androidx.fragment.app.Fragment {  
17     protected static final String TAG = "FRAGMENT";  
18     protected ProgressDialog progressDialog;  
19  
20     protected abstract void initializeUIComponents(View view);  
21  
22     protected void goToMainActivity() {  
23         Log.d(TAG, "goToMainActivity: called");  
24         startActivity(new Intent(getContext(), MainActivity.class));  
25     }  
26  
27     protected void add(int cid, AbstractFragment f) {  
28         Log.d(TAG, "replace: called");  
29         if (getActivity() != null) {  
30             FragmentManager fm = getActivity().getSupportFragmentManager();  
31             FragmentTransaction ft = fm.beginTransaction();  
32             ft.add(cid, f).commit();  
33         }  
34     }  
35  
36     protected void replace(int cid, AbstractFragment f) {  
37         Log.d(TAG, "replace: called");  
38         if (getActivity() != null) {  
39             FragmentManager fm = getActivity().getSupportFragmentManager();  
40             FragmentTransaction ft = fm.beginTransaction();  
41             ft.replace(cid, f).commit();  
42         }  
43     }  
44  
45     protected void showLoading() {  
46         Log.d(TAG, "showLoading: called");  
47         if (progressDialog == null) {  
48             progressDialog = new ProgressDialog(getActivity());  
49         }  
50         progressDialog.setMessage(getString(R.string.text_please_wait));  
51         progressDialog.setCancelable(false);  
52         progressDialog.show();  
53     }  
54  
55     protected void dismissLoading() {  
56         Log.d(TAG, "dismissLoading: called");  
57         if (progressDialog != null && progressDialog.isShowing()) {  
58             progressDialog.dismiss();  
59         }  
60     }  
61  
62     protected void showToast(String message) {  
63         Log.d(TAG, "showToast: called");  
64         if (message.length() < 20) {  
65             Toast toast = Toast.makeText(getActivity(), message, Toast.LENGTH_SHORT);  
66             toast.setGravity(Gravity.BOTTOM, 0, 200);  
67             toast.show();  
68         } else {  
69             Toast toast = Toast.makeText(getActivity(), message, Toast.LENGTH_LONG);  
70             toast.setGravity(Gravity.BOTTOM, 0, 200);  
71             toast.show();  
72         }  
73     }  
74 }  
75 }
```

initializing the UI components,
implementing listeners for “et_search”,
“ibtn_search” and implementing searchProduct() and
other methods to successfully retrieve the requested
products from Firebase Firestore

fragments/ (2)

HomeFragment.java

```
24 public class HomeFragment extends AbstractFragment {  
25     protected static final String TAG = "HOME_F";  
26     protected ArrayList<Product> products = new ArrayList<>();  
27     protected String sSearch;  
28     protected EditText etSearch;  
29     protected ImageButton ibtnSearch;  
30     protected FirebaseFirestore ffdb = FirebaseFirestore.getInstance();  
31  
32     public View onCreateView(@NonNull LayoutInflater inflater,  
33                             ViewGroup container,  
34                             Bundle savedInstanceState) {  
35         Log.d(TAG, "onCreateView: called");  
36  
37         // initialize ui  
38         View homeView = inflater.inflate(R.layout.fragment_home, container, false);  
39         initializeUIComponents(homeView);  
40  
41         return homeView;  
42     }  
43  
44     protected void initializeUIComponents(View homeView) {  
45         Log.d(TAG, "initializeUIComponents: called");  
46  
47         // initialize components  
48         etSearch = homeView.findViewById(R.id.et_search);  
49         ibtnSearch = homeView.findViewById(R.id.ibtn_search);  
50  
51         // on click  
52         ibtnSearch.setOnClickListener(searchListener);  
53     }  
54  
55     protected void searchProducts() {  
56         Log.d(TAG, "searchProducts: called");  
57  
58         // TODO  
59         // full-text search  
60  
61         ffdb.collection("products")  
62             .whereEqualTo("name", sSearch)  
63             .get()  
64             .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  
65                 @Override  
66                 public void onComplete(@NonNull Task<QuerySnapshot> task) {  
67                     if (task.isSuccessful()) {  
68                         Log.d(TAG, "searchProducts: succeeded");  
69                         for (QueryDocumentSnapshot d : task.getResult()) {  
70                             addProduct(d);  
71                         }  
72                         showProducts();  
73                     } else {  
74                         Log.w(TAG, "searchProducts: failed", task.getException());  
75                         dismissLoading();  
76                         showToast("Loading products failed");  
77                     }  
78                 }  
79             });  
80     }  
81 }
```

adapters/ (0)

master SF-assignments / app / src / main / java / com / example / shopsmart / views / adapters /

| | | |
|--|---|-------------|
|  pvarouktsis | Update message add product and showToast function in adapters | |
| .. | | |
|  HomeProductListAdapter.java | Update activities, fragments, adapters | 13 days ago |
|  MyCartProductListAdapter.java | Update activities, fragments, adapters | 13 days ago |
|  OnSaleProductListAdapter.java | Update message add product and showToast function in adapters | 12 days ago |
|  ProductListAdapter.java | Update message add product and showToast function in adapters | 12 days ago |
|  ProductListAdapterFactory.java | Update logo, adapter list, layouts, formation in drawables | 14 days ago |

“HomeProductListAdapter”,
“MyCartProductListAdapter”,
“OnSaleProductListAdapter”
inherit from
“ProductListAdapter”

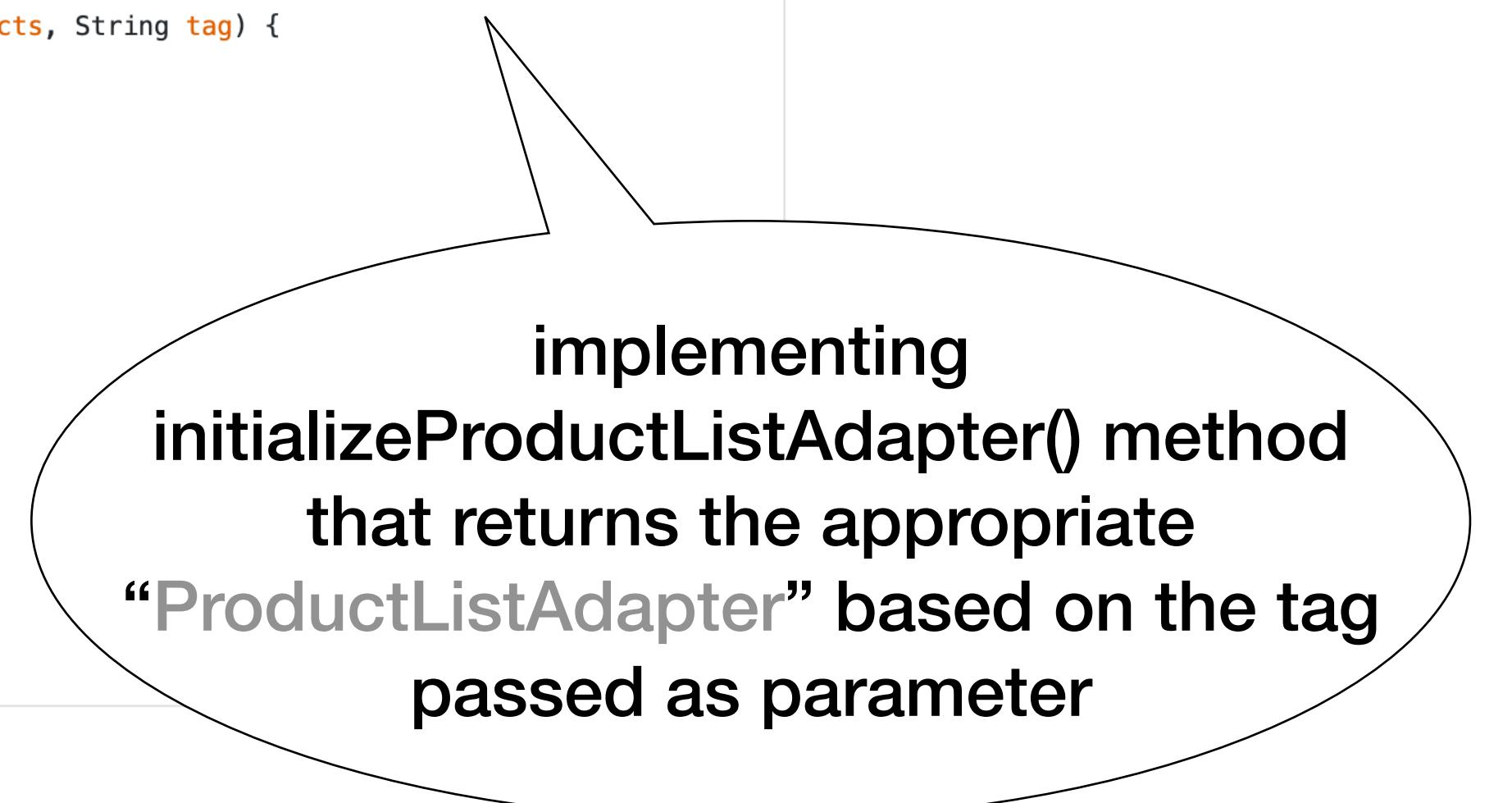
implementing a factory pattern
and an inheritance

adapters/ (1)

ProductListAdapterFactory.java

```
32 lines (25 sloc) | 969 Bytes
Raw Blame   
```

```
1 package com.example.shopsmart.views.adapters;
2
3 import android.app.Activity;
4 import android.util.Log;
5
6 import com.example.shopsmart.models.Product;
7
8 import java.util.ArrayList;
9
10 public class ProductListAdapterFactory {
11     protected static final String TAG = "PRODUCT_LIST_AF";
12
13     public ProductListAdapterFactory() {
14         // Required
15     }
16
17     public ProductListAdapter initializeProductListAdapter(Activity activity, ArrayList<Product> products, String tag) {
18         Log.d(TAG, "initializeProductListAdapter: called");
19         ProductListAdapter pla;
20         if (tag.equals("HOME_F")) {
21             pla = new HomeProductListAdapter(activity, products);
22         } else if (tag.equals("MY_CART_F")) {
23             pla = new MyCartProductListAdapter(activity, products);
24         } else if (tag.equals("ON_SALE_F")) {
25             pla = new OnSaleProductListAdapter(activity, products);
26         } else {
27             pla = null;
28         }
29         return pla;
30     }
31 }
32 }
```



implementing
initializeProductListAdapter() method
that returns the appropriate
“ProductListAdapter” based on the tag
passed as parameter

implementing different methods that are inherited by the subclasses e.g. `getUser()`, `showToast()`, etc. but also implementing the “ProductViewHolder” passing the view inflated with the “layout_product”

adapters/ (2)

ProductListAdapter.java

```

92     protected void getUser() {
93         Log.d(TAG, "getUser: called");
94         FirebaseUser fu = fa.getCurrentUser();
95         ffdb.collection("users")
96             .document(fu.getUid())
97             .get()
98             .addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
99                 @Override
100                public void onComplete(@NonNull Task<DocumentSnapshot> task) {
101                    if (task.isSuccessful()) {
102                        Log.d(TAG, "getUser: succeeded");
103                        DocumentSnapshot d = task.getResult();
104                        user = d.toObject(User.class);
105                    } else {
106                        Log.w(TAG, "getUser: failed", task.getException());
107                        user = null;
108                    }
109                }
110            });
111        }
112
113        protected void showToast(String message) {
114            Log.d(TAG, "showToast: called");
115            if (message.length() < 20) {
116                Toast toast = Toast.makeText(activity, message, Toast.LENGTH_SHORT);
117                toast.setGravity(Gravity.BOTTOM, 0, 200);
118                toast.show();
119            } else {
120                Toast toast = Toast.makeText(activity, message, Toast.LENGTH_LONG);
121                toast.setGravity(Gravity.BOTTOM, 0, 200);
122                toast.show();
123            }
124        }
125
126        /*
127         * ProductViewHolder class
128         */
129        protected class ProductViewHolder extends RecyclerView.ViewHolder {
130            protected RelativeLayout rlProduct;
131            protected RelativeLayout rlProductStandard;
132            protected RelativeLayout rlProductExpanded;
133            protected ImageView ivProductImage;
134            protected TextView tvProductName;
135            protected TextView tvProductPrice;
136            protected TextView tvProductSupermarket;
137            protected Button btnExtend;
138            protected Button btnAdd;
139            protected Button btnRemove;
140
141            protected ProductViewHolder(View productView) {
142                super(productView);
143                rlProduct = productView.findViewById(R.id.rl_product);
144                rlProductStandard = productView.findViewById(R.id.rl_product_standard);
145                rlProductExpanded = productView.findViewById(R.id.rl_product_expanded);
146                ivProductImage = productView.findViewById(R.id.iv_product_image);
147                tvProductName = productView.findViewById(R.id.tv_product_name);
148                tvProductPrice = productView.findViewById(R.id.tv_product_price);
149                tvProductSupermarket = productView.findViewById(R.id.tv_product_supermarket);
150                btnExtend = productView.findViewById(R.id.btn_extend);
151                btnAdd = productView.findViewById(R.id.btn_add);
152                btnRemove = productView.findViewById(R.id.btn_remove);
153            }
154        }
155    }
156 }
```

overriding `onCreateViewHolder()`,
`onBindViewHolder()` methods, implementing them
 based on the requirements (e.g. show “`btn_add`” or
 “`btn_remove`”) and, also, implementing `addToCart()`
 method for adding products to user’s cart

adapters/ (3)

HomeProductListAdapter.java

```

20 public class HomeProductListAdapter extends ProductListAdapter {
21     protected static final String TAG = "HOME_PLA";
22
23     public HomeProductListAdapter(Activity activity, ArrayList<Product> products) {
24         super(activity, products);
25     }
26
27     @NonNull
28     @Override
29     public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
30         Log.d(TAG, "onCreateViewHolder: called");
31
32         // initialize ui
33         LayoutInflater inflater = LayoutInflater.from(parent.getContext());
34         View view = inflater.inflate(R.layout.layout_product, parent, false);
35         ProductViewHolder productListView = new ProductViewHolder(view);
36         initializeUIComponents();
37
38         // initialize visibility
39         productListView.btnAdd.setVisibility(View.VISIBLE);
40
41         // initialize firebase
42         fa = FirebaseAuth.getInstance();
43
44         // initialize user
45         getUser();
46
47         return productListView;
48     }
49
50     @Override
51     public void onBindViewHolder(@NonNull ProductViewHolder pvh, final int position) {
52         Log.d(TAG, "onBindViewHolder: called");
53
54         // initialize view of product
55         Picasso.get().load(products.get(position).getImageURL()).into(pvh.ivProductImage); // printing image
56         pvh.tvProductName.setText(products.get(position).getName());
57         pvh.tvProductPrice.setText(products.get(position).getPriceToStringWithEuroSymbol());
58
59         // expand view of product
60         final boolean isExpanded = (expandedPosition == position);
61         pvh.rlProductExpanded.setVisibility(isExpanded ? View.VISIBLE : View.GONE); // expand or contract product based on current state
62         pvh.btnExtend.setVisibility(!isExpanded ? View.VISIBLE : View.GONE); // remove extend_button based on current state
63
64         pvh.rlProduct.setActivated(isExpanded);
65
66         if (isExpanded) {
67             previousExpandedPosition = position;
68         }
69
70         pvh.rlProduct.setOnClickListener(new View.OnClickListener() {
71             @Override
72             public void onClick(View v) {
73                 expandedPosition = isExpanded ? -1 : position;
74                 notifyItemChanged(previousExpandedPosition);
75                 notifyItemChanged(position);
76             }
77         });
78
79         // add product
80         pvh.btnAdd.setOnClickListener(new View.OnClickListener() {
81             @Override
82             public void onClick(View v) {
83                 addProductToCart(products.get(position));
84             }
85         });
86     }
87
88     protected void addProductToCart(Product product) {
89         Log.d(TAG, "addProductToCart: called");
90
91         // add product to user's cart
92         FirebaseUser fu = fa.getCurrentUser();
93         user.getCart().add(product);
94         ffdb.collection("users")
95             .document(fu.getUid())
96             .set(user, SetOptions.merge());
97
98         // toast for product added
99         showToast("Product added");
100    }
101
102 }
```

Tips

Getting started

- start with a diagram (e.g. decide the activities, fragments, etc. or even design the java's directory architecture)
- continue with res (e.g. build the UI of activities, fragments and other components as designed previously in a prototyping design tool)
- end with java (e.g. initialize the components of a UI and implement the operations of them via listeners)

and also...

- search, fix, repeat

Thank you