

# Mobile Application Development

---

Produced  
by

David Drohan ([ddrohan@wit.ie](mailto:ddrohan@wit.ie))

Department of Computing & Mathematics  
Waterford Institute of Technology  
<http://www.wit.ie>

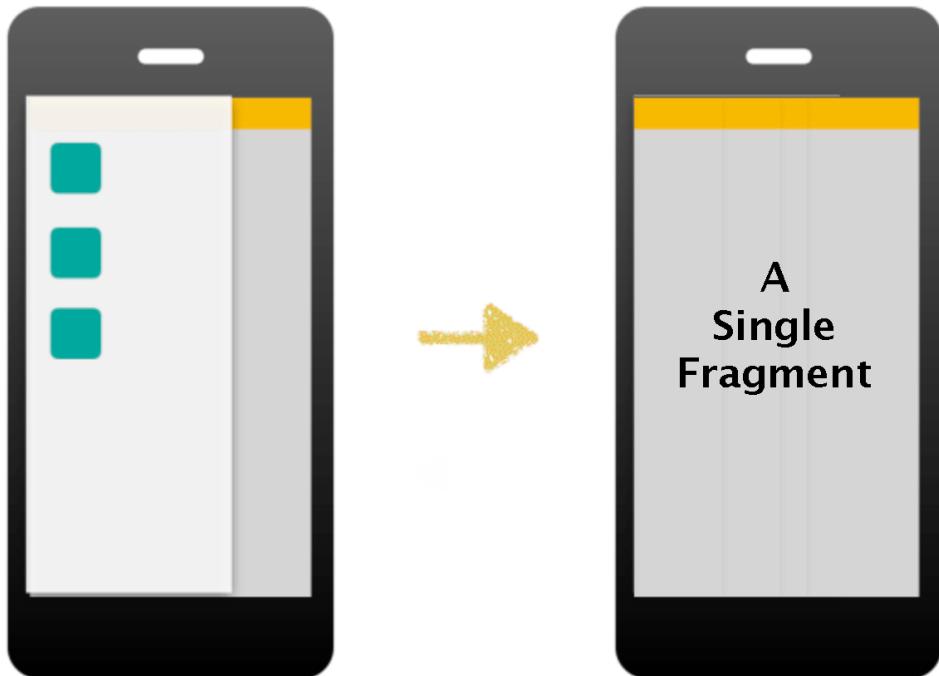


Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



# Donation Code Walkthrough

---





# Agenda

---

- ❑ Manual Nav Drawer Setup Walkthrough
- ❑ Code Extracts from Donation

# Nav Drawer in Donation

---

Manual Setup (no Navigation Component)

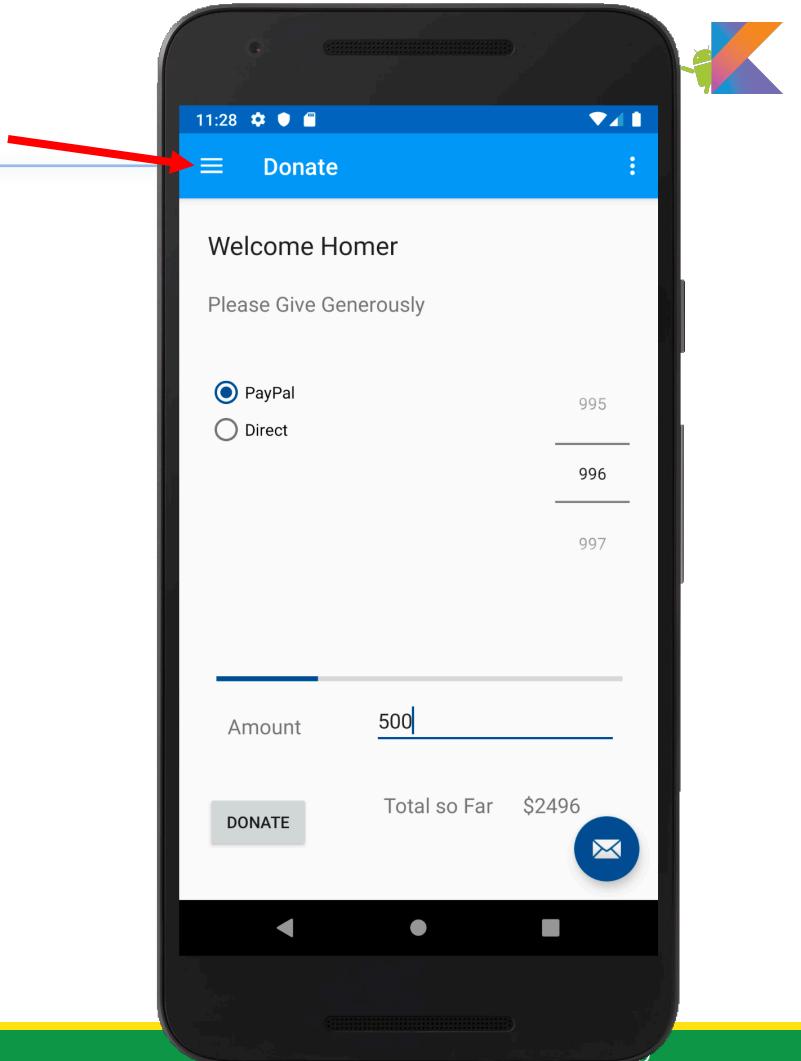


# Case Study

❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

## ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar

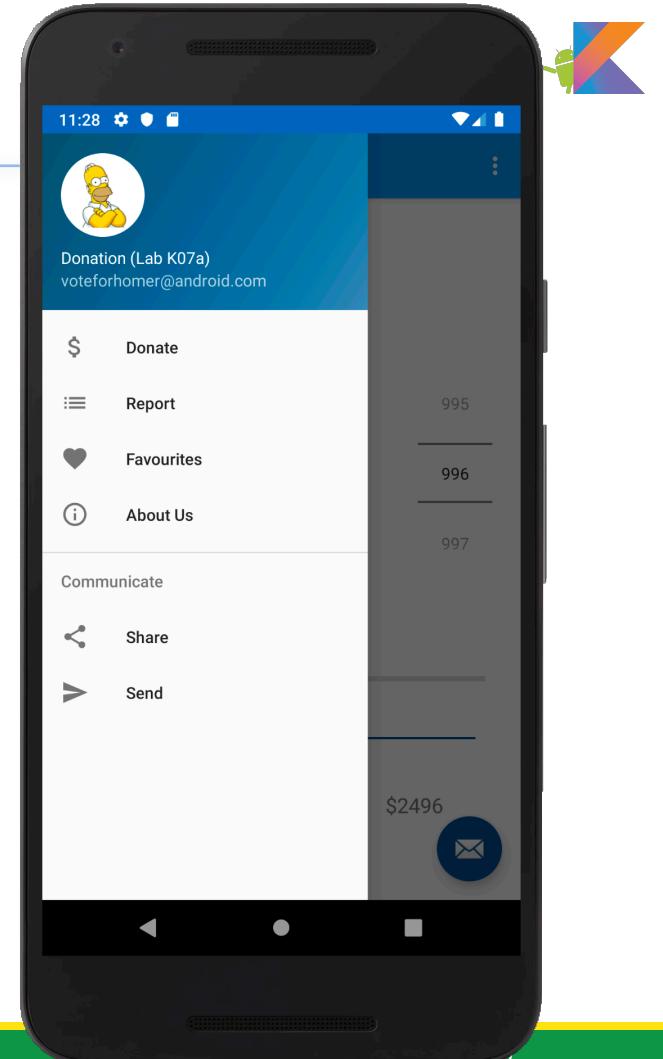


# Case Study

❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

## ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar

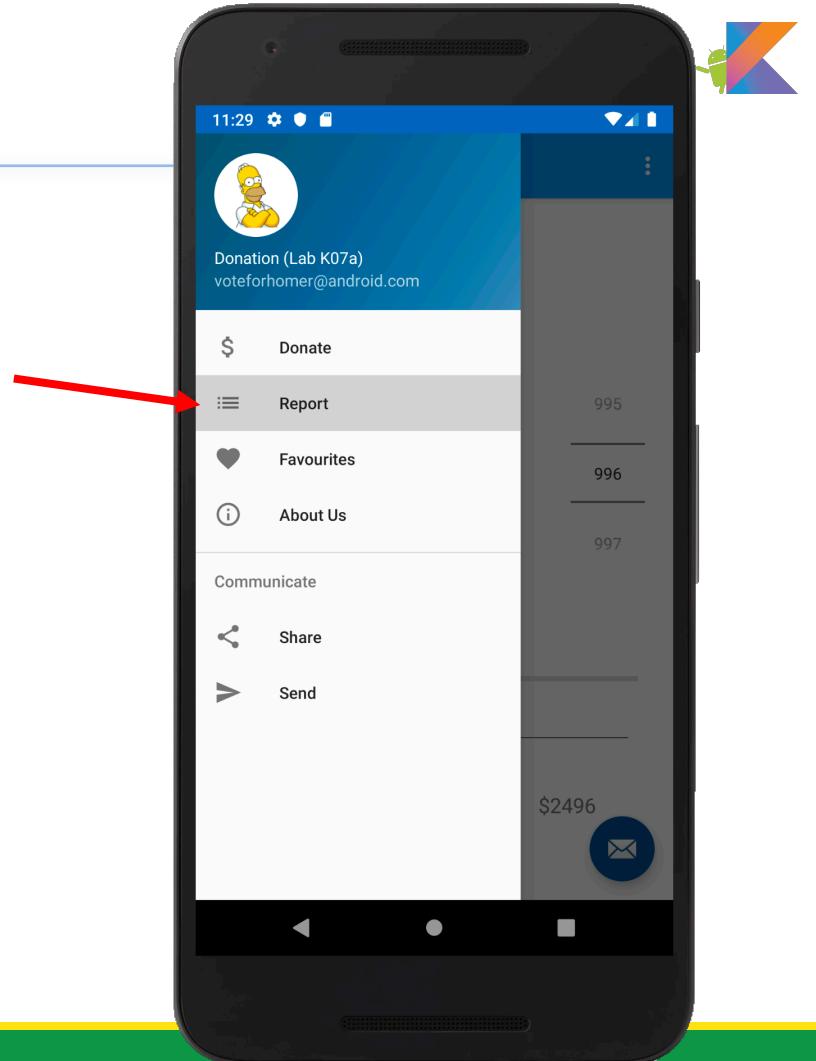


# Case Study

- ❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

- ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar

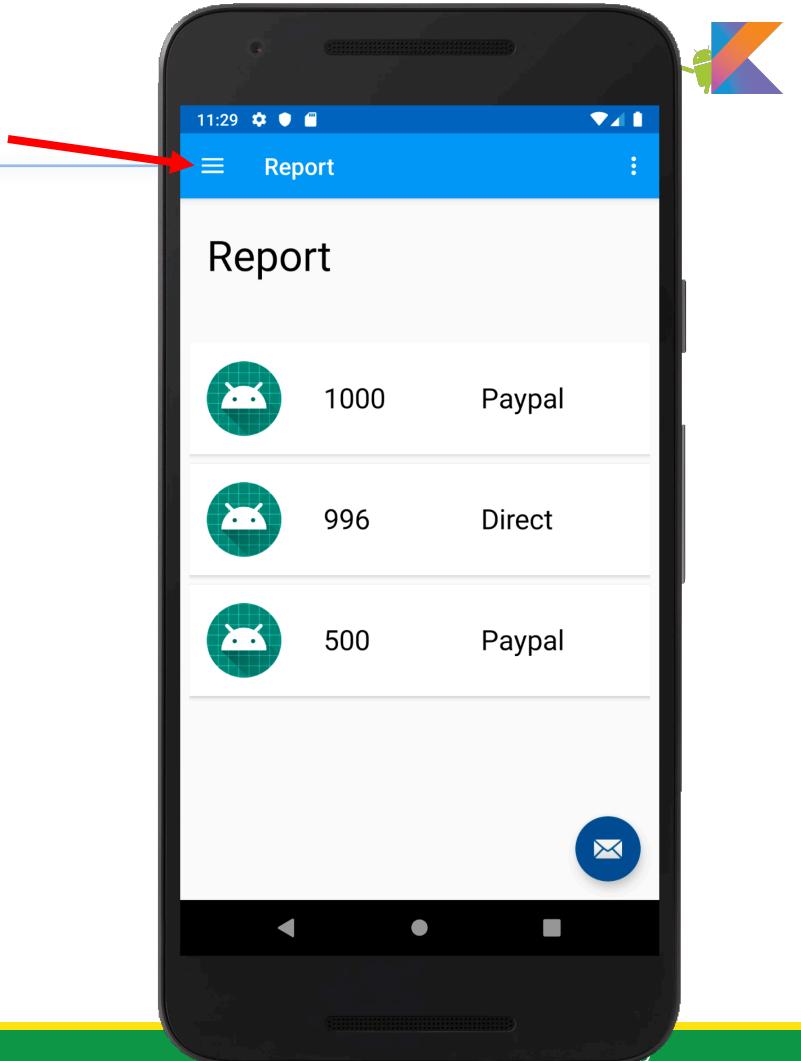


# Case Study

❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

## ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar

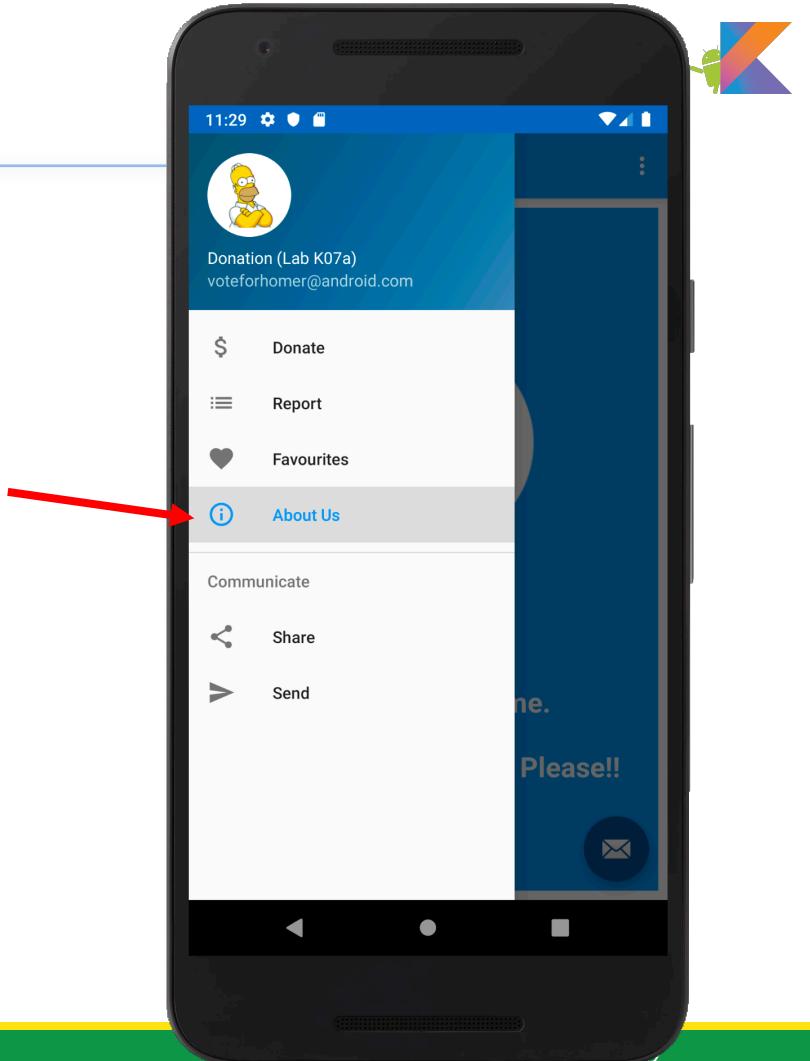


# Case Study

❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

## ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar

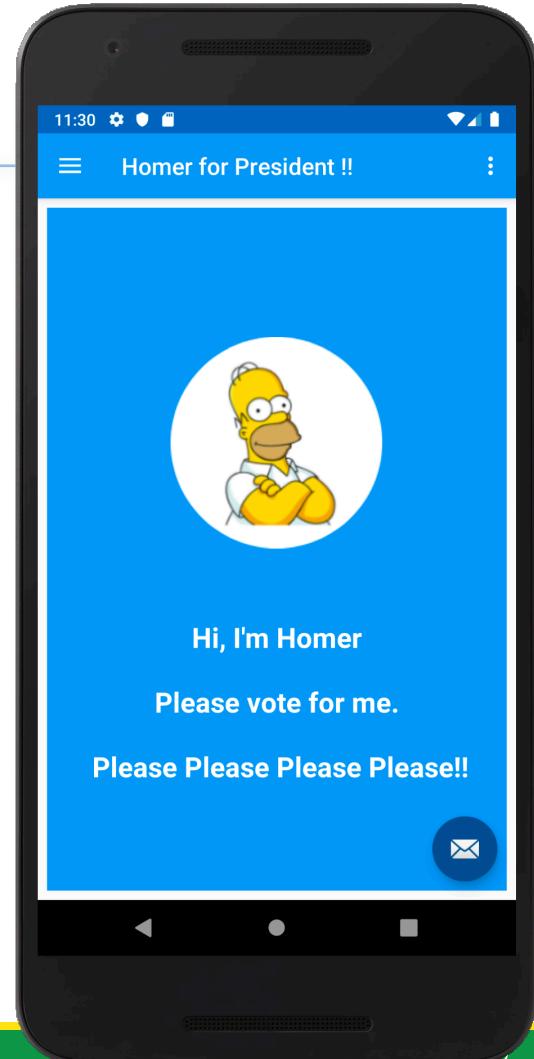


# Case Study

❑ An Android App to keep track of donations made to ‘Homers Presidential Campaign ’.

## ❑ App Features

- Accept donation via number picker or typed amount
- Keep a running total of donations
- Display report on donation amounts and types
- Display running total on progress bar





# Navigation Drawer Overview

---

- ❑ Android Studio (and the Navigation Library) does a lot of the heavy lifting for you, but generally the following steps are necessary to add a Navigation Drawer to your app
  - *Create drawer layout*
  - *Bind to navigation drawer layout*
  - *Handle navigation drawer click and*
  - *Update content based on user selection*



## Overview - *Create Drawer Layout*

---

- ❑ For creating a navigation drawer, first we need to declare the drawer layout in your main activity (our Home) where you want to show the navigation drawer
  
- ❑ You add  
**`androidx.drawerlayout.widget.DrawerLayout`** as the root view of your activity layout.
  
- ❑ We'll use **Donation** as the example to illustrate...

# Overview - Create Drawer Layout

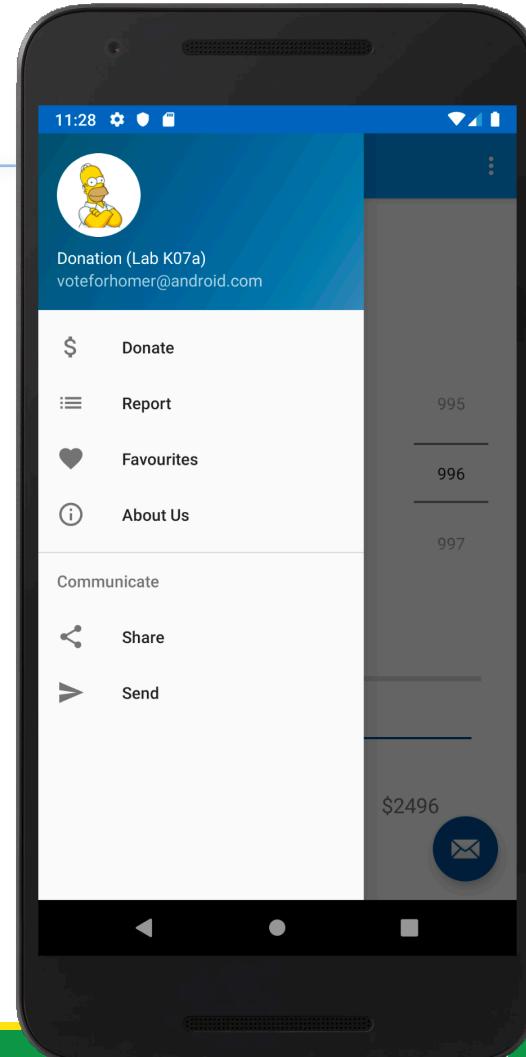


```
home.xml <?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout <!-- Line 2 -->
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_home" <!-- Line 13 -->
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView <!-- Line 17 -->
        android:id="@+id/navView"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_home" <!-- Line 23 -->
        app:menu="@menu/activity_home_drawer" />

</androidx.drawerlayout.widget.DrawerLayout>
```





# Overview - Create Drawer Layout

```
home.xml <?xml version="1.0" encoding="utf-8"?>
1   <androidx.drawerlayout.widget.DrawerLayout ←
2     <include
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       xmlns:app="http://schemas.android.com/apk/res-auto"
5       xmlns:tools="http://schemas.android.com/tools"
6       android:id="@+id/drawerLayout"
7       android:layout_width="match_parent"
8       android:layout_height="match_parent"
9       android:fitsSystemWindows="true"
10      tools:openDrawer="start">
11
12      <include
13        layout="@layout/app_bar_home" ←
14        android:layout_width="match_parent"
15        android:layout_height="match_parent" />
16
17      <com.google.android.material.navigation.NavigationView
18        android:id="@+id/navView" ←
19        android:layout_width="wrap_content"
20        android:layout_height="match_parent"
21        android:layout_gravity="start"
22        android:fitsSystemWindows="true"
23        app:headerLayout="@layout/nav_header_home" ←
24        app:menu="@menu/activity_home_drawer" />
25
26    </androidx.drawerlayout.widget.DrawerLayout>
```

- ❑ *home.xml* contains the Navigation Header (*nav\_header\_home*) AND the Navigation Drawer Menu (*activity\_home\_drawer*) inside a **NavigationView**.
- ❑ *home* includes *app\_bar\_home* which will display our content
- ❑ Also, note the ‘ids’ of the widgets (for later on)



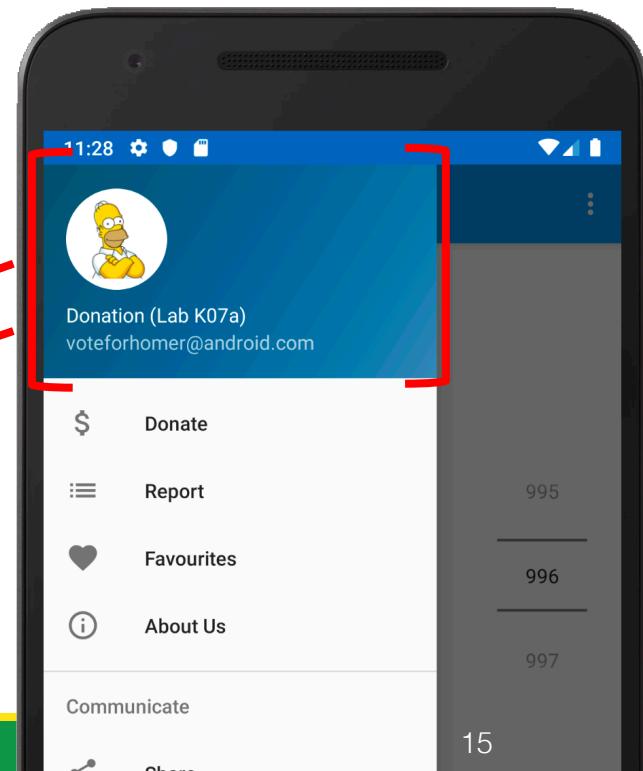
# Overview – nav\_header\_home

```
nav_header_home.xml <?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="176dp"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Navigation header"
        android:paddingTop="8dp"
        app:srcCompat="@mipmap/ic_launcher_homer" />

    <TextView...>
    <TextView...>

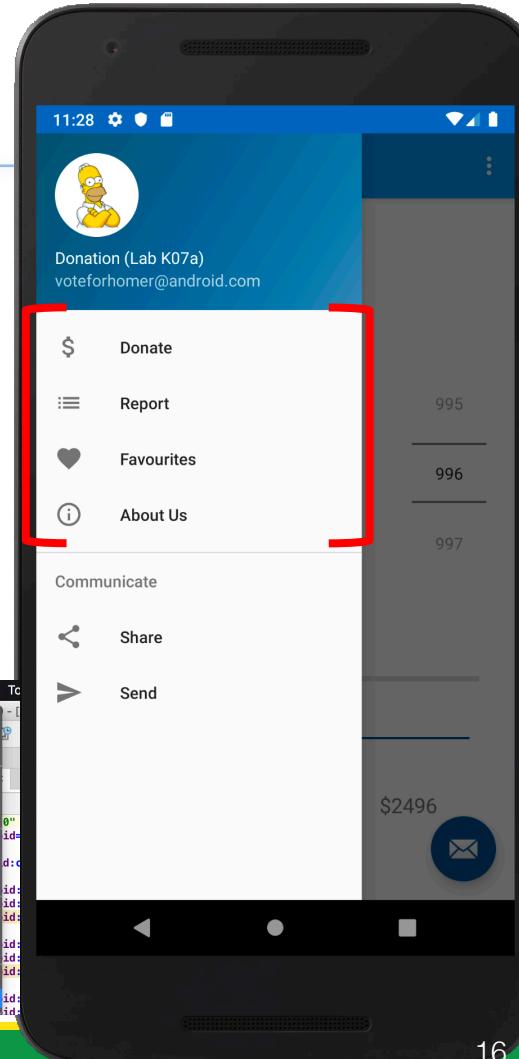
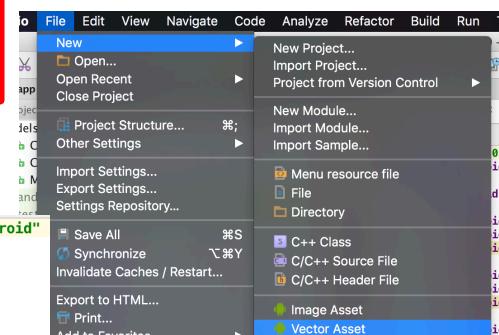
</LinearLayout>
```



# Overview – activity\_home\_drawer



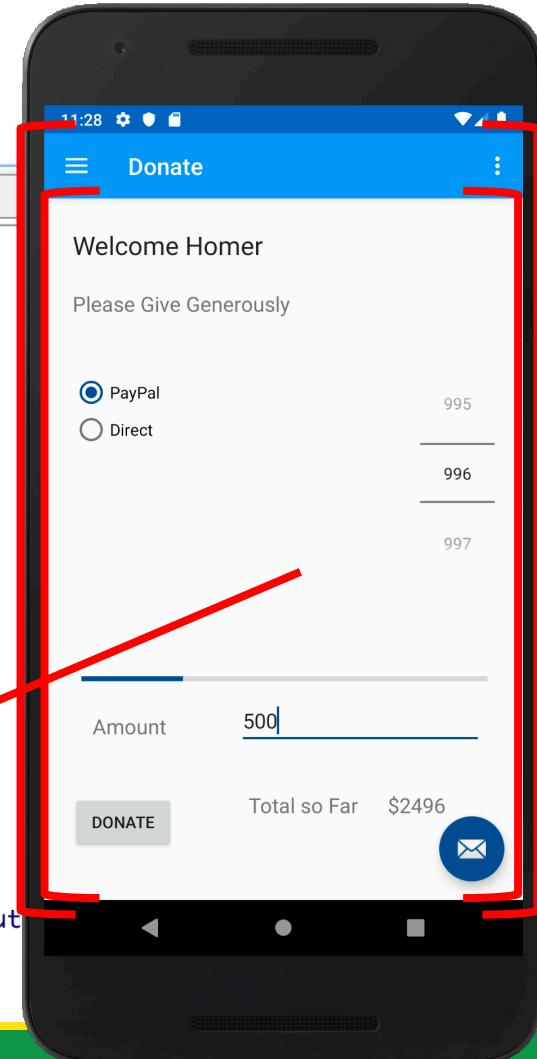
```
activity_home_drawer.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      tools:showIn="navigation_view">
5
6      <group android:checkableBehavior="single">
7          <item
8              android:id="@+id/nav_donate"
9              android:icon="@drawable/ic_menu_donate"
10             android:title="Donate" />
11
12          <item
13              android:id="@+id/nav_report"
14              android:icon="@drawable/ic_menu_report"
15              android:title="Report" />
16
17          <item
18              android:id="@+id/nav_favourites"
19              android:icon="@drawable/ic_menu_favorites"
20              android:title="@string/menu_favourites" />
21
22          <item
23              android:id="@+id/nav_aboutus"
24              android:icon="@drawable/ic_menu_aboutus"
25              android:title="About Us" />
26
27      </group>
28
29      <item android:title="Communicate" ...>
30
31          <vector xmlns:android="http://schemas.android.com/apk/res/android"
32              android:width="24dp"
33              android:height="24dp"
34              android:viewportWidth="24.0"
35              android:viewportHeight="24.0">
36
37              <path
38                  android:fillColor="#FF000000"
39                  android:pathData="M10,20v-6h4v6h5v-8h3L12,3 2,12h3v8z"/>
40
41      </vector>
42
43  </menu>
```



# Overview – *app\_bar\_home*



```
app_bar_home.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.coordinatorlayout.widget.CoordinatorLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".activities.Home">
9
10    <com.google.android.material.appbar.AppBarLayout
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:theme="@style/AppTheme.AppBarOverlay">
14
15        <androidx.appcompat.widget.Toolbar ...>
16
17    </com.google.android.material.appbar.AppBarLayout>
18
19    <include layout="@layout/content_home" /> ←
20
21    <com.google.android.material.floatingactionbutton.FloatingActionButton
22        android:id="@+id/fab"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="bottom|end"
26        android:src="@drawable/ic_fab" />
27
28</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

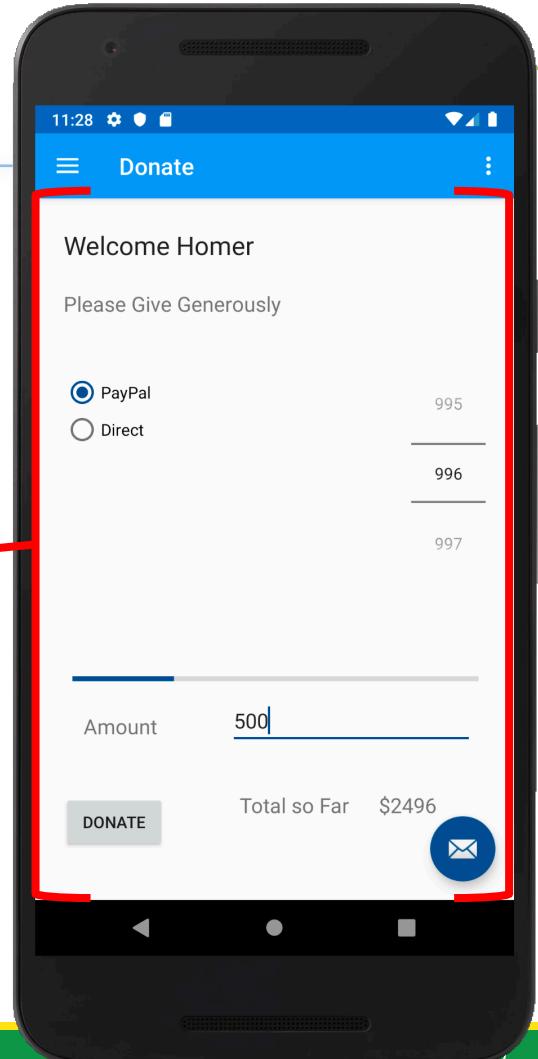


# Overview – *content\_home*



```
content_home.xml x
```

```
1 <androidx.constraintlayout.widget.ConstraintLayout
2     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:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="com.google.android.material.appbar.AppBarLayout"
8     tools:showIn="@layout/app_bar_home">
9
10    <FrameLayout
11        android:id="@+id/homeFrame"
12        android:layout_width="0dp"
13        android:layout_height="0dp"
14        android:layout_marginBottom="8dp"
15        android:layout_marginEnd="8dp"
16        android:layout_marginStart="8dp"
17        android:layout_marginTop="8dp"
18        app:layout_constraintBottom_toBottomOf="parent"
19        app:layout_constraintEnd_toEndOf="parent"
20        app:layout_constraintStart_toStartOf="parent"
21        app:layout_constraintTop_toTopOf="parent" />
22
23 </androidx.constraintlayout.widget.ConstraintLayout>
```





## Overview – *Bind to the Drawer Layout etc.*

---

- ❑ Once you have the necessary layouts and menu in place, you then need to bind to the **Drawer** and **Navigation View** to allow you to handle the user navigation and switching content based on user selection.
- ❑ In your **onCreate ()** you'll have something like the following

```
navView.setNavigationItemSelectedListener(this)
val toggle = ActionBarDrawerToggle(this, drawerLayout, toolbar,
    "Open navigation drawer", R.string.navigation_drawer_close
)
drawerLayout.addDrawerListener(toggle)
toggle.syncState()
```



## Overview – *Bind to the Drawer Layout etc.*

---

- ❑ You'll probably want to display some kind of initial landing page once the app starts so in our example, we load up the Donate Screen (maintained in our **DonateFragment**)
- ❑ Again, in your **onCreate()** you'll have something like the following

```
ft = supportFragmentManager.beginTransaction()  
  
val fragment = DonateFragment.newInstance()  
ft.replace(R.id.homeFrame, fragment)  
ft.commit()
```

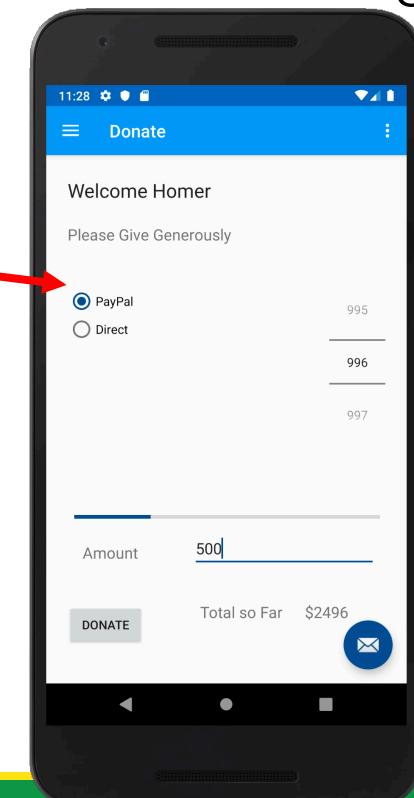
- ❑ This creates a new instance of a **DonateFragment** and replaces the fragment in our **FrameLayout** with this instance



# Overview – Handle Drawer Click & Update Content \*

- To handle users menu selection we implement the following

```
override fun onNavigationItemSelected(item: MenuItem): Boolean {  
  
    when (item.itemId) {  
        R.id.nav_donate ->  
            navigateTo(DonateFragment.newInstance())  
        R.id.nav_report ->  
            navigateTo(ReportFragment.newInstance())  
        R.id.nav_aboutus ->  
            navigateTo(AboutUsFragment.newInstance())  
  
        else -> toast("You Selected Something Else")  
    }  
    drawerLayout.closeDrawer(GravityCompat.START)  
    return true  
}
```

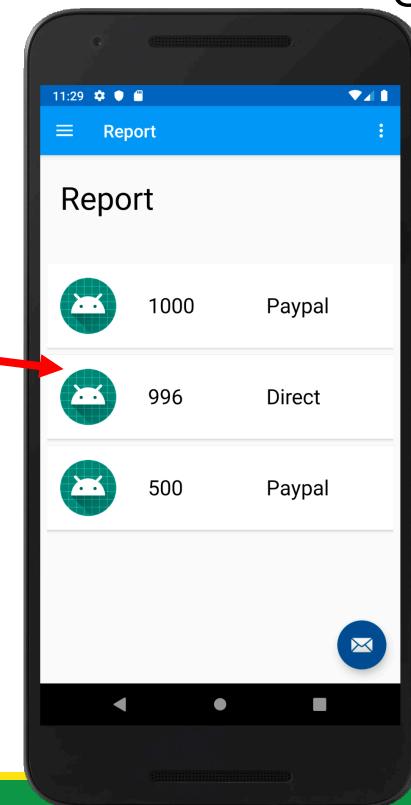




# Overview – Handle Drawer Click & Update Content \*

- To handle users menu selection we implement the following

```
override fun onNavigationItemSelected(item: MenuItem): Boolean {  
  
    when (item.itemId) {  
        R.id.nav_donate ->  
            navigateTo(DonateFragment.newInstance())  
        R.id.nav_report ->  
            navigateTo(ReportFragment.newInstance())  
        R.id.nav_aboutus ->  
            navigateTo(AboutUsFragment.newInstance())  
  
        else -> toast("You Selected Something Else")  
    }  
    drawerLayout.closeDrawer(GravityCompat.START)  
    return true  
}
```

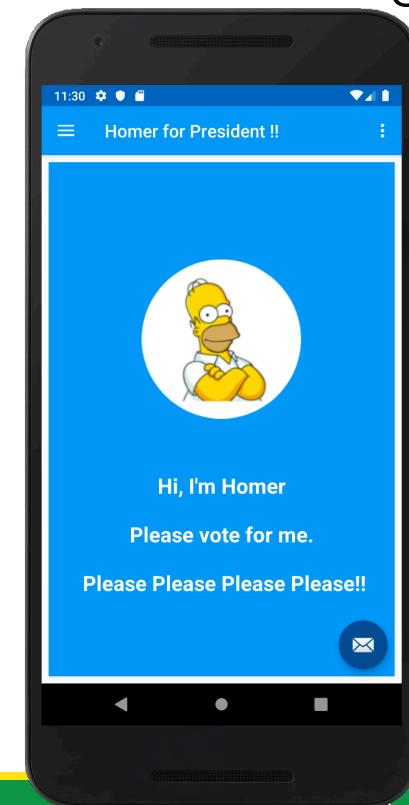




# Overview – Handle Drawer Click & Update Content \*

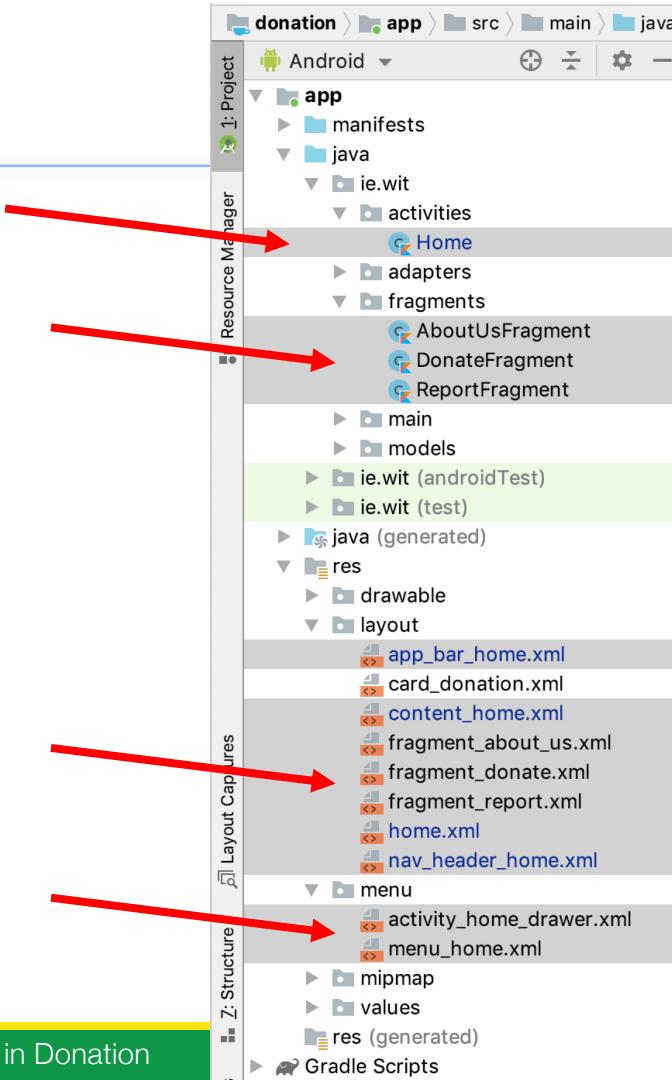
- To handle users menu selection we implement the following

```
private fun navigateTo(fragment: Fragment) {  
    supportFragmentManager.beginTransaction()  
        .replace(R.id.homeFrame, fragment)  
        .addToBackStack(null)  
        .commit()  
}
```



# Project Structure

- Added Layouts & Menus  
(Manually)
- Added 'Controller' Home  
(Manually)
- Added 'Blank Fragments'  
(Auto, but managed  
Manually)





## References

---

Sources:

- <https://blog.mindorks.com/android-navigation-drawer-in-kotlin>
- <https://medium.com/@umang.burman.micro/navigation-drawer-with-navigation-component-4f032bfdeae6>
- <https://developer.android.com/guide/navigation>
- <https://medium.com/@myric.september/navigation-in-android-navigation-architecture-component-a1f103a6cc52>
- <https://material.io/components/navigation-drawer/#>

Thanks.

