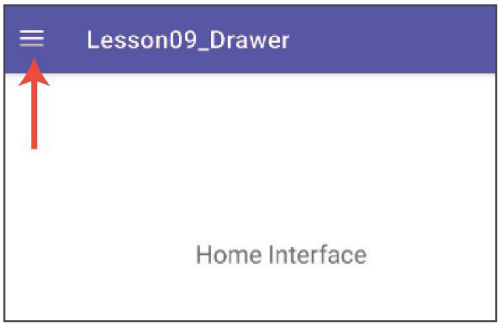
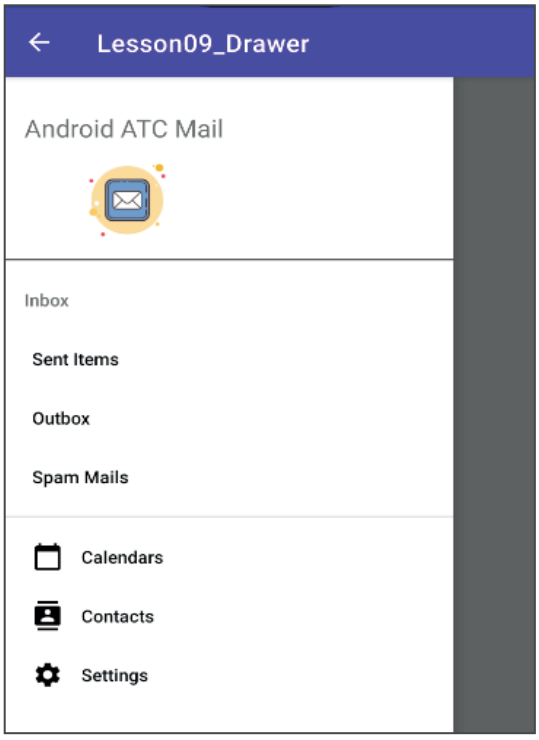
# Lab 09:

## Creating Navigation Drawer in Android App

In this lab, you will create an Android app using a drawer widget as a navigation technique for your app content. At the end of this lab you should get the following drawer menu:



The steps are as follows:

1- Open Android Studio, and then click **File → New → New Project**

2- Select **Empty Activity**, and click **Next**

3- Type: **Lab09** for the application name, then click **Finish**.

4- Before you start with typing the Kotlin code in your app, check the **build.gardle (Module: Lab09)** file and be sure it has the Kotlin plugin. If not, add the following code: **id 'kotlin-android-extensions'**
And click the **Sync Now**. The configuration should be as follows:

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
}
```

5- Open the **activity_main.xml** file in the Design mode, and then **delete** the "Hello World!" text.

6- Open the **activity_main.xml** file in the Code mode. At the top of this code, replace the following code:

```
<androidx.constraintlayout.widget.ConstraintLayout
```

With:

```
<androidx.drawerlayout.widget.DrawerLayout
```

7- Configure an id attribute value for your Drawer layout by adding the following attribute:

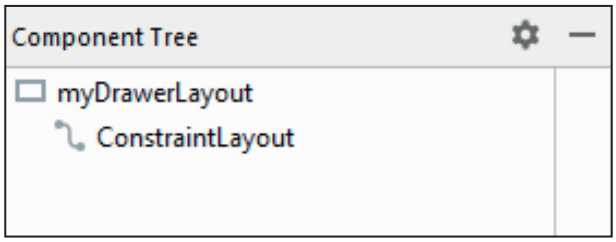
```
android:id="@+id/myDrawerLayout"
```

The full code of **activity_main.xml** file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>


<androidx.drawerlayout.widget.DrawerLayout 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/myDrawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"/>
```

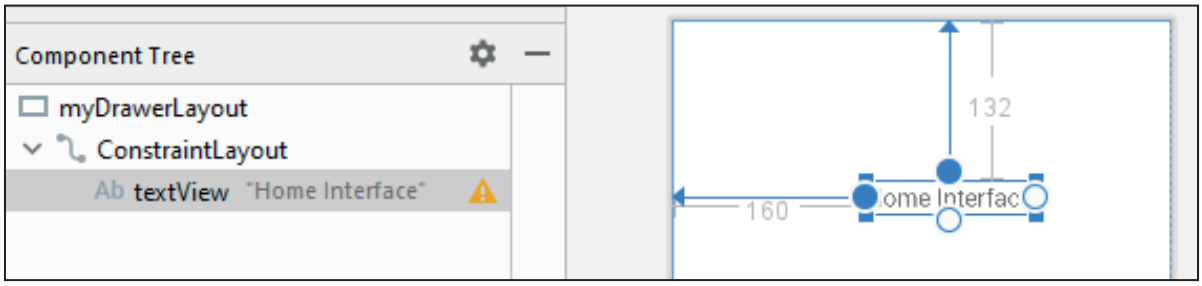
8- Open the **activity_main.xml** file in the **Design** mode. From the **Palette** panel (Layouts), add a **ConstraintLayout** widget to your activity interface using the drag and drop technique. This layout must be a child widget of your **DrawerLayout** as illustrated in the figure below for the Component Tree.

Here, this constraint layout represents your main or home activity interface.




9- From the **Palette** panel (Text), add a **TextView** widget to your activity. It should be a child widget of your **ConstraintLayout** widget. Set its constraints, set its **text** attribute value to **Home Interface**, and set its **textSize** to **20sp**.

Your interface should have the following design:

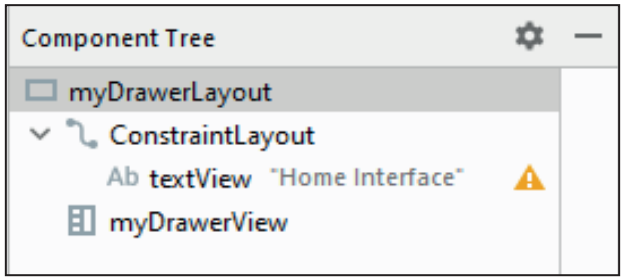



The **activity_main.xml** file in the **Code** mode is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout 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/myDrawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:layout_marginStart="160dp"
            android:layout_marginTop="132dp"
            android:text="Home Interface"
            android:textSize="20sp"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.drawerlayout.widget.DrawerLayout>
```

10- Open the **activity_main.xml** file in the **Design** mode. From the **Palette** panel (Containers), drag and drop a **NavigationView** widget to the **Component Tree** panel and be sure to add it as a child widget of your **DrawerLayout** widget. The component structure of your widgets must have the following design:




**Importance Note**: In the components tree above, the **ConstraintLayout** (or any interface activity container) must be before your **myDrawerView** (NavigationView) widget, otherwise, the navigation list items will not respond to any user's taps.

11- Open the **activity_main.xml** file in the **Code** mode and add the following attributes to your **NavigationView** tag:

```
android:id="@+id/myDrawerView"
android:layout_gravity="start"
android:fitsSystemWindows="true"
```

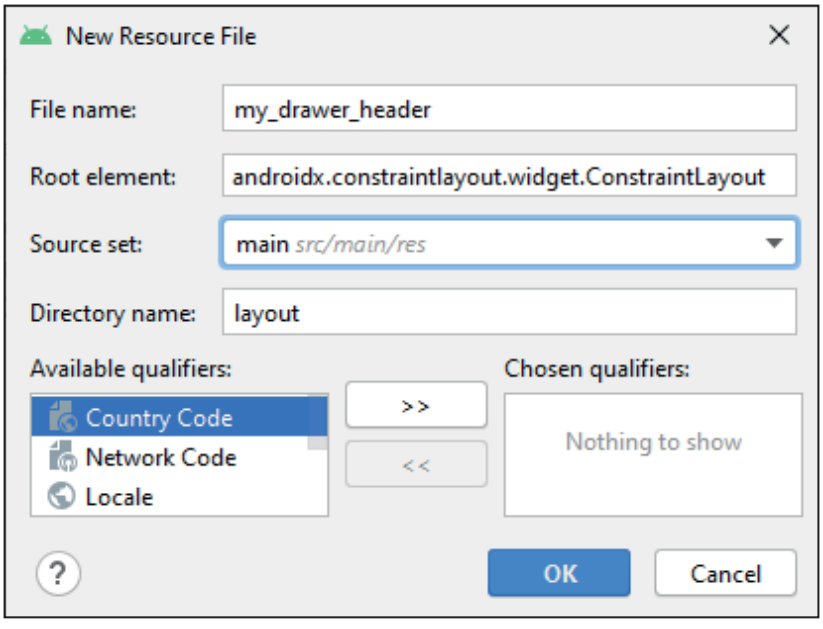
The full code of your **activity_main.xml** file as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout 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/myDrawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="160dp"
            android:layout_marginTop="132dp"
            android:text="Home Interface"
            android:textSize="20sp"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

    </androidx.constraintlayout.widget.ConstraintLayout>

    <com.google.android.material.navigation.NavigationView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/myDrawerView"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        android:clipToPadding="false"
        app:headerLayout="@layout/my_drawer_header"
        app:menu="@menu/my_drawer_items"/>

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

12- Your Drawer navigation list consists of a header and navigation items. In this step you will create your Drawer navigation header which consists of an image and text. To do that, right click **layout → New Resource File**

As illustrated in the dialog box below, type: **my_drawer_header** for the **File name** and click **OK**.

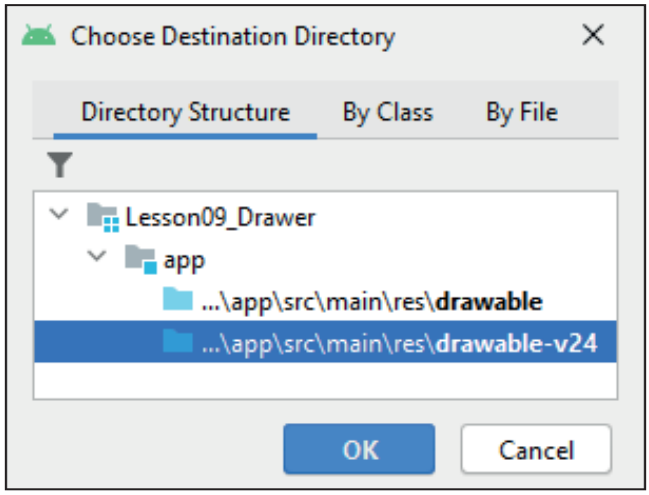


13- We want to get the following design for the **my_drawer_header.xml** file. It consists of text, icon and horizontal line.

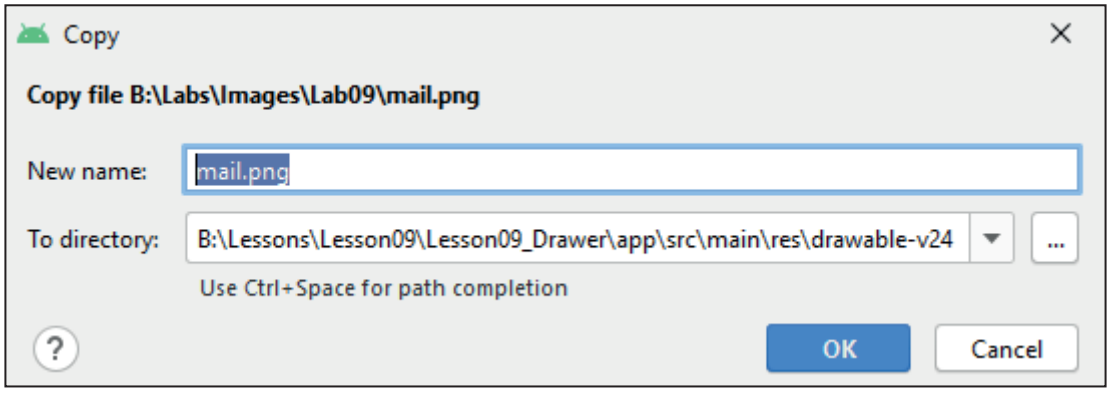

To get this design, open the **my_drawer_header.xml** file in the Design mode, add a **TextView** widget, set its constraints, set its attributes values as follows:

| | |
|---|---|
| **text**: Android ATC Mail | **textSize** :20sp |

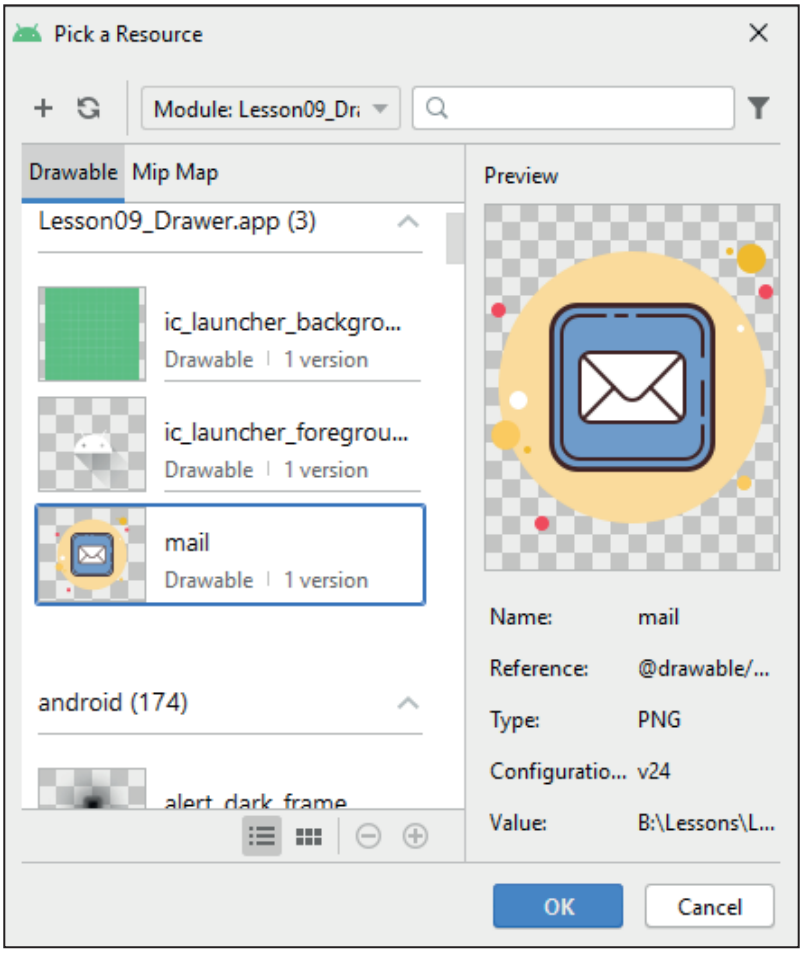
14- To add the icon from the **Labs** files which you have in your computer, open **Labs\Images\Lab09**, copy: **mail.png**. Back to your **Android Studio**, right click the **drawable** container, select **Paste**, then select the destination directory as illustrated in the figure below. Click **OK**.




15- As illustrated in the dialog box below, keep the default configurations and click **OK**



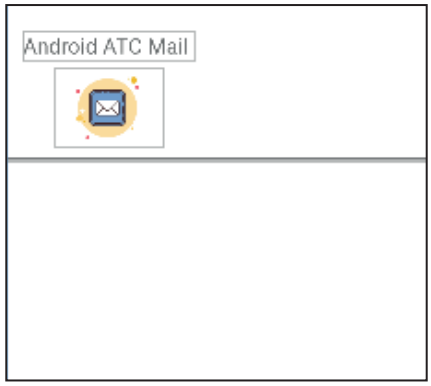
16- From the **Palette** panel (Widgets), Add an **ImageView** widget to your activity interface using the drag and drop technique. As illustrated in the **Pick a Resource** dialog box below, select your image, click **OK**. Then, set your image constraints.




17- To add a horizontal line in **my_drawer_header.mxl** file below your image (mail.png), open the file in the **Design** mode, add a **<view>** widget (from the **Palette** panel → **Widgets**) to your interface. Open your file (**my_drawer_header.mxl**) in the **Code** mode, and set the **View** tag attributes values as illustrated in the following gray highlighted XML code:

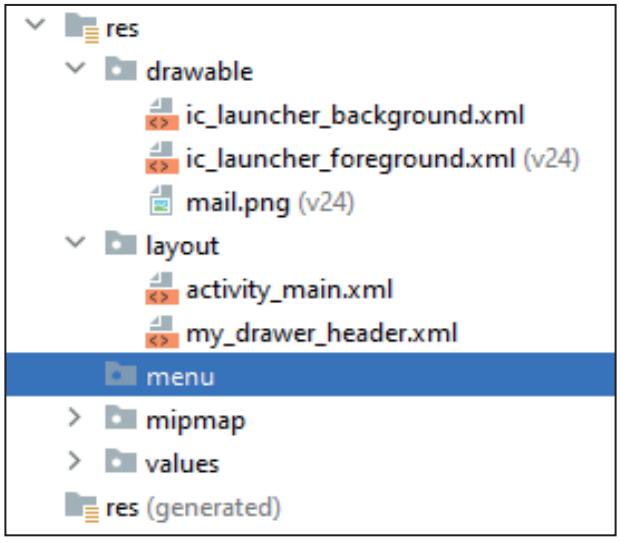
```
<View
    android:id="@+id/view"
    android:layout_width="409dp"
    android:layout_height="1dp"
    android:layout_marginTop="148dp"
    android:background="@color/black"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Your **my_drawer_header.mxl** file should have a design that looks like the following figure:



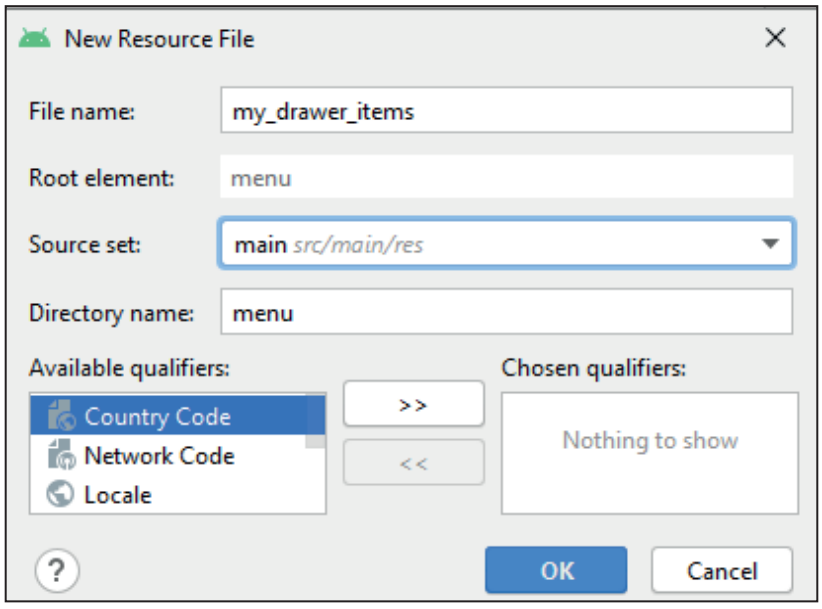

Now in the following steps, you will create the content of your drawer list items. These items consist of texts and icons. First of all, create a menu directory which will later include the file **menu.xml**. This file will include all the components of your drawer list items.

18- Right click the **res** directory → **New** " **New Resource Directory**. Type **menu** for the **Directory name**, select **menu** for the **Resource type** and click **OK**. You should have the following files and directories structure:

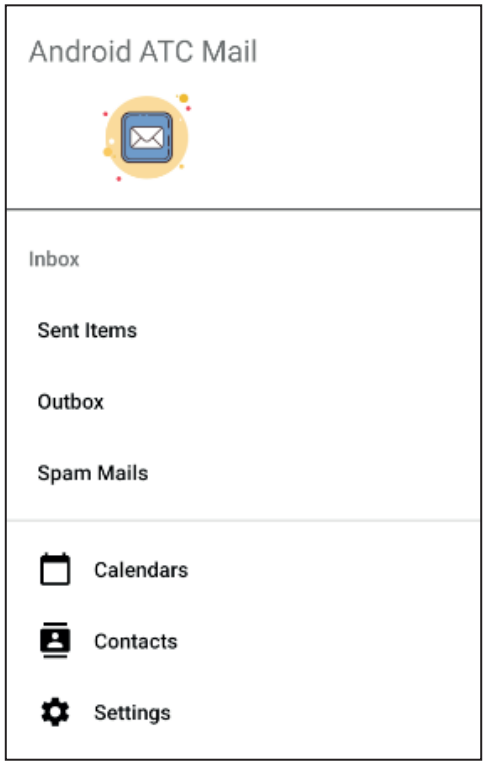



19- Right click the **menu** directory → **New** → **Menu Resource File**

As illustrated in the dialog box below, type **my_drawer_items** for the **File name**, and click **OK**.

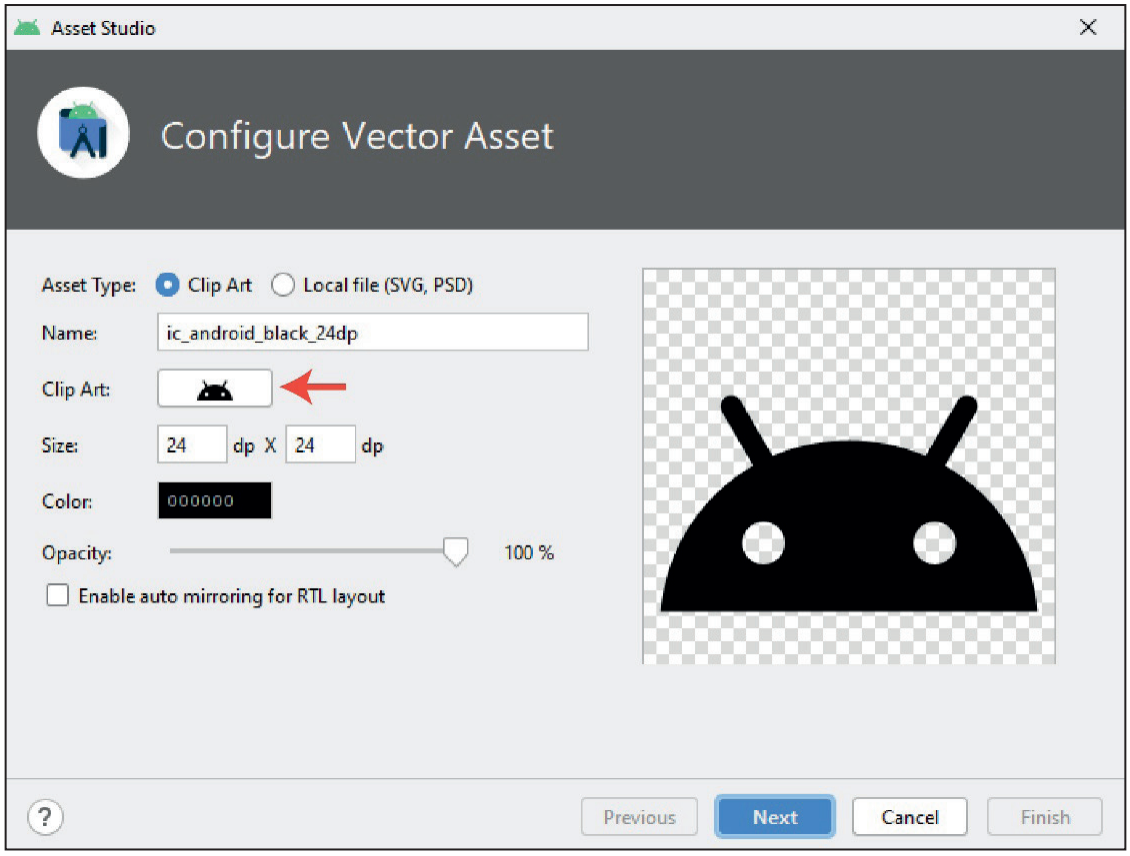



20- Our goal is to create the following drawer items list:



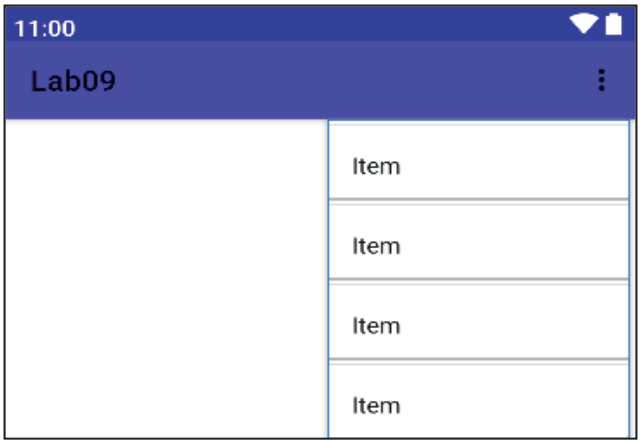

To do that, first import these icons (Calendars, Contacts, and Settings) into your Android project. To add these icons into your project, right click **drawable** container → **New** → **Vector Asset.**

From the following dialog box, click the **Clip Art** icon, type: **mail** in the search area, select the mail icon, click **OK**, click **Next** and click **Finish**




Repeat the same steps to add the following icons: **calendar today**, **contacts**, and **settings**

21- We want to add four items to this drawer list (Inbox, Calendar, Contacts, and Settings). To do that, open the **my_drawer_items.xml** file in the Design mode, and from the **Palette** panel add four **Menu Item** widgets using the drag and drop technique (you may also add them using your XML code). You should have the following design:




22- You may continue using the design mode for **my_drawer_items.xml** file to configure your drawer items by selecting each Item (Menu Item), click the **Attributes** tab, then configure the item **id**, **title**, and add the **icon** too. However, we will do this task using the XML code because we have later planned to add sub menu for the **Inbox** menu item.

Open your **my_drawer_items.xml** file in the **Code** mode, you should have the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:title="Item" />
    <item android:title="Item" />
    <item android:title="Item" />
    <item android:title="Item" />
</menu>
```

Edit this file to have the following configuration:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/inboxId"
        android:icon="@drawable/ic_baseline_mail_24"
        android:title="Inbox"/>

    <item
        android:id="@+id/calendarId"
        android:icon="@drawable/ic_baseline_calendar_today_24"
        android:title="Calendars" />

    <item android:id="@+id/contactsId"
        android:icon="@drawable/ic_baseline_contacts_24"
        android:title="Contacts" />

    <item android:id="@+id/settingsId"
        android:icon="@drawable/ic_baseline_settings_24"
        android:title="Settings" />
</menu>
```

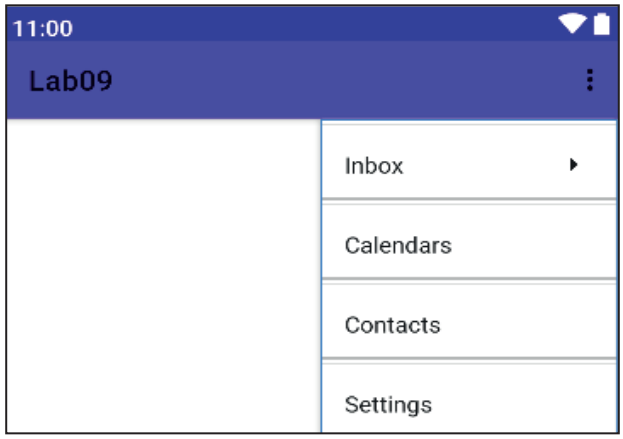
23- To add a sub menu to your **Inbox** item, add the following XML code to your **Inbox** Item tag:

```
<item
    android:id="@+id/inboxId"
    android:icon="@drawable/ic_baseline_mail_24"
    android:title="Inbox">
    <menu>
        <item
            android:id="@+id/sentId"
            android:title="Sent Items" />
```

```
        <item
            android:id="@+id/outboxId"
            android:title="Outbox" />
        <item
            android:id="@+id/spamId"
            android:title="Spam Mails" />
    </menu>
</item>
```

Note: In the code above, remove the close sign tag (/) of the first Item tag. It is beside **android:title="Inbox"**

If you check your file in the **Design** mode, you should have the following design:




The full code of the **my_drawer_items.xml** file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">


    <item
        android:id="@+id/inboxId"
        android:icon="@drawable/ic_baseline_mail_24"
        android:title="Inbox">
        <menu>
            <item
                android:id="@+id/sentId"
                android:title="Sent Items" />
            <item
                android:id="@+id/outboxId"
                android:title="Outbox" />
            <item
                android:id="@+id/spamId"
                android:title="Spam Mails" />
```

```
</menu>
    </item>

    <item
        android:id="@+id/calendarId"
        android:icon="@drawable/ic_baseline_calendar_today_24"
        android:title="Calendars" />

    <item android:id="@+id/contactsId"
        android:icon="@drawable/ic_baseline_contacts_24"
        android:title="Contacts" />

    <item android:id="@+id/settingsId"
        android:icon="@drawable/ic_baseline_settings_24"
        android:title="Settings" />

</menu>
```

24- Now, this step is very important, because you will tell your **NavigationView** widget which you have added in the **activity_main.xml** file before about which file must be used as a header for your Drawer and which one must be used for the menu list. To do this, open the **activity_main.xml** file in the **Code** mode and add the two gray highlighted attributes to your **NavigationView** tag. The code is as follows:

```
<com.google.android.material.navigation.NavigationView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/myDrawerView"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/my_drawer_header"
    app:menu="@menu/my_drawer_items"/>
```

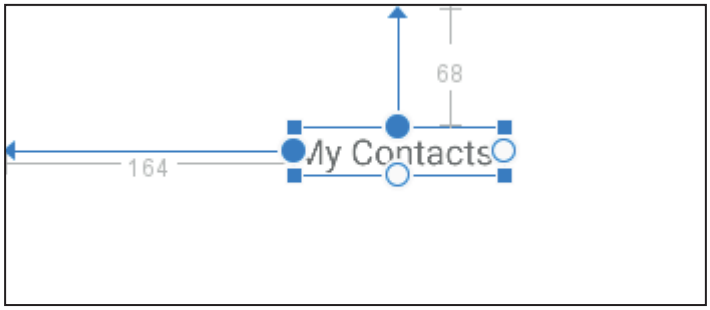
25- Our plan for the app user is for him to tap the **Contacts** in the drawer list. When this is done, another activity called **Contacts** will open. This is just to test some navigation techniques which we may use with our drawer list. Therefore, let us create a new activity by right clicking **Layouts → New → Activity → Empty Activity**

Type **Contacts** in the **File Name**, and click **Finish**

26- Open the **activity_contacts.xml** file in the Design mode, add a **TextView** widget, set its constraints, and set its attributes values as follows:

| | |
|---|---|
| **text** : Contacts | **textSize**: 20sp |

You should have the following design:




27- Open the **MainActivity** file, and add the following configuration for the **myToggle** variable:

```
class MainActivity : AppCompatActivity() {

    lateinit var myToggle: ActionBarDrawerToggle

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
}
}
```

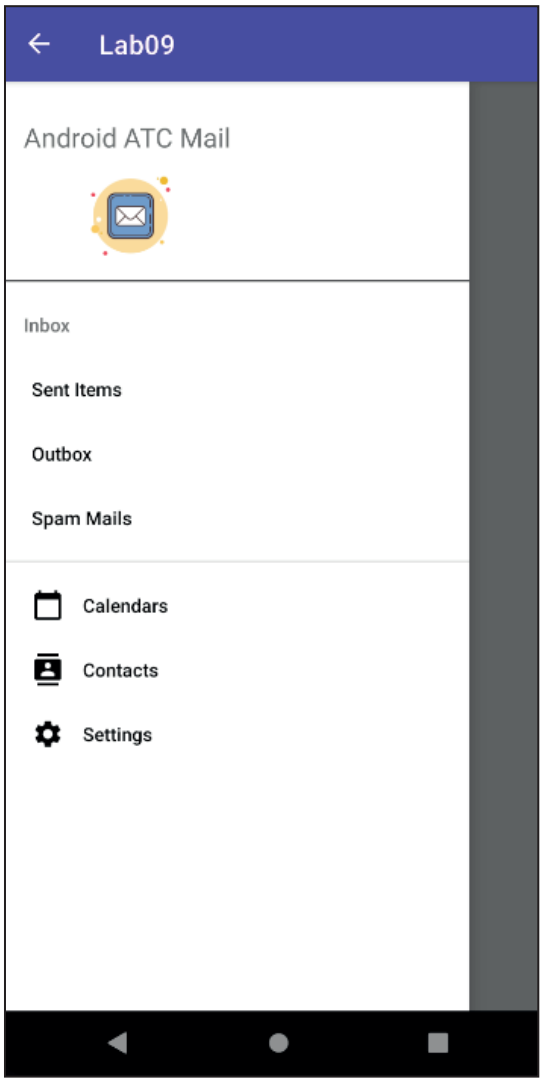
28- Then add the following configuration for the **MainActivity** file. If you get a red under line under the **`DrawerLayout`**, double click it, click the red pop-up lamp, and select **Import**.

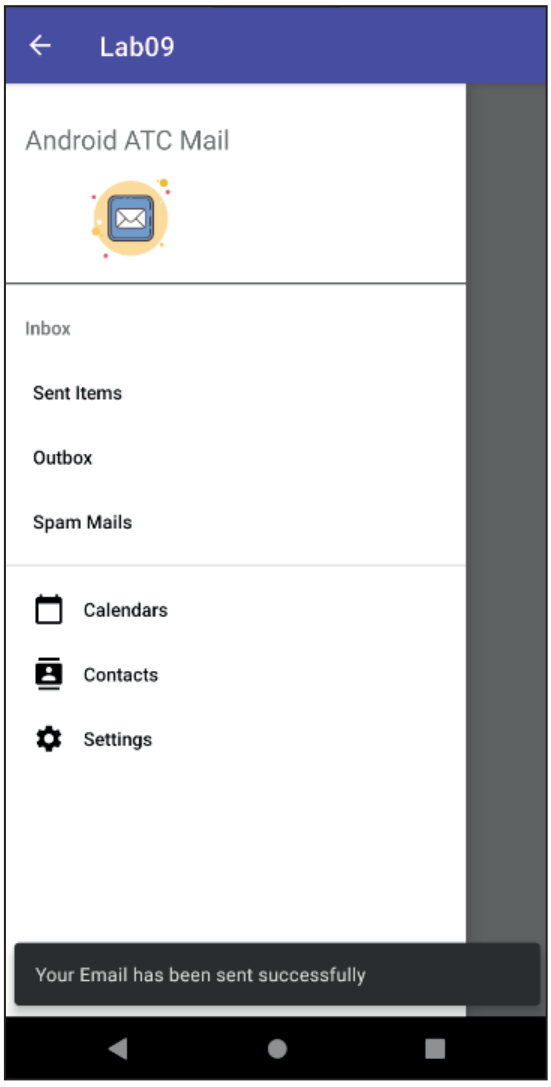
```
class MainActivity : AppCompatActivity() {

    lateinit var myToggle: ActionBarDrawerToggle

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

   val drawerLayout: DrawerLayout = findViewById(R.id.myDrawerLayout)
   val navView: NavigationView = findViewById(R.id.myDrawerView)

   myToggle = ActionBarDrawerToggle(this, drawerLayout, 0, 0)

   myDrawerLayout.addDrawerListener(myToggle)

   myToggle.syncState()
     supportActionBar?.setDisplayHomeAsUpEnabled(true)

   navView.setNavigationItemSelectedListener {
 when (it.itemId) {
```

```
    R.id.sentId -> {
        Snackbar.make(findViewById(R.id.sentId),"Your Email has been
sent successfully",Snackbar.LENGTH_LONG).show()
                }

    R.id.outboxId -> {
        Snackbar.make(findViewById(R.id.outboxId),"This is your
OutBox folder",Snackbar.LENGTH_LONG).show()
                }

    R.id.spamId -> {
        Snackbar.make(findViewById(R.id.spamId),"This is your Spam
folder",Snackbar.LENGTH_LONG).show()
                }

    R.id.calendarId -> {
         Snackbar.make(findViewById(R.id.calendarId),"This is your
Calendar",Snackbar.LENGTH_LONG).show()
                }

     R.id.contactsId -> {
         startActivity(Intent(this, Contacts::class.java))}
                }

                true

            }
        }

 override fun onOptionsItemSelected(item: MenuItem): Boolean {

      if (myToggle.onOptionsItemSelected(item)) {
          return true
      }
      return super.onOptionsItemSelected(item)
    }

}
```

As you see in the code above, the **setNavigationItemSelectedListener** method helps you in configuring the navigation for your drawer list.

29- **Run** your app. Tap your drawer button on your app toolbar and tap your drawer items. You should have the following output results:


Lab09
Android ATC Mail
Inbox
Sent Items
Outbox
Spam Mails
Calendars
Contacts
Settings


Lab09
Android ATC Mail
Inbox
Sent Items
Outbox
Spam Mails
Calendars
Contacts
Settings
Your Email has been sent successfully