Girl Geek Dinner Workshop

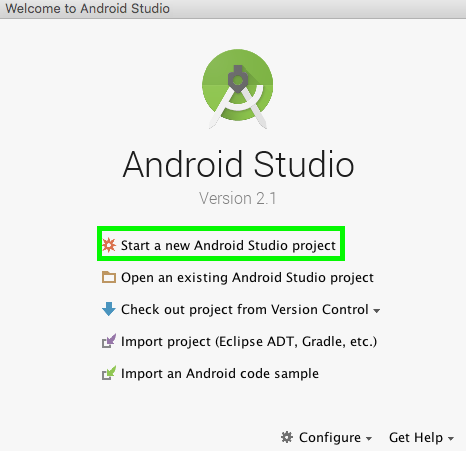
July 20, 2016

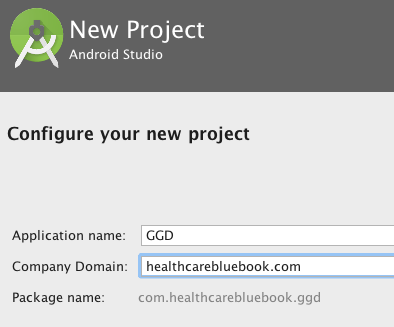
Workshop Instructions

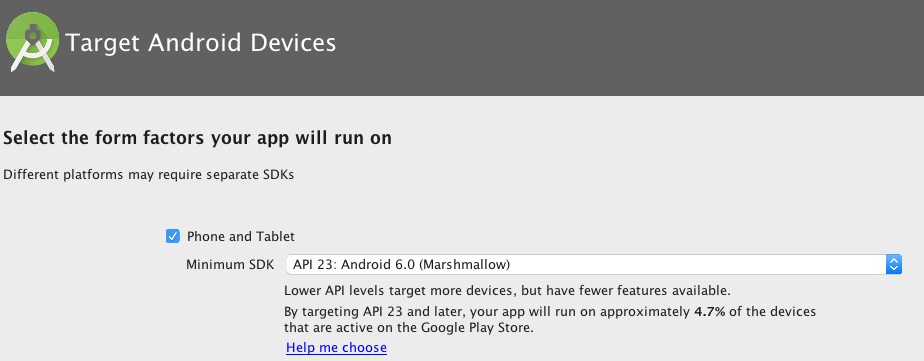
Make sure you’ve followed the steps in the Prerequisite document prior to continuing.

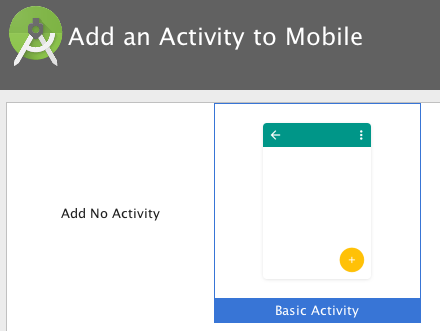
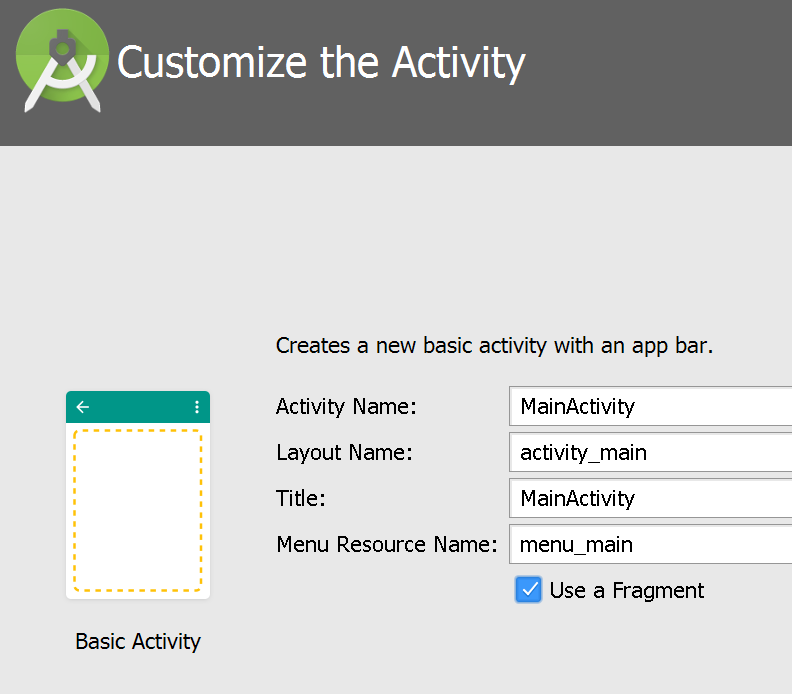
Section 1: Create new Project

1. Open Android Studio. Select “Start a new Android Studio project”

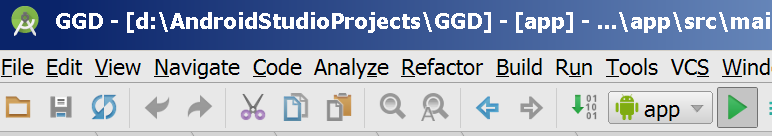
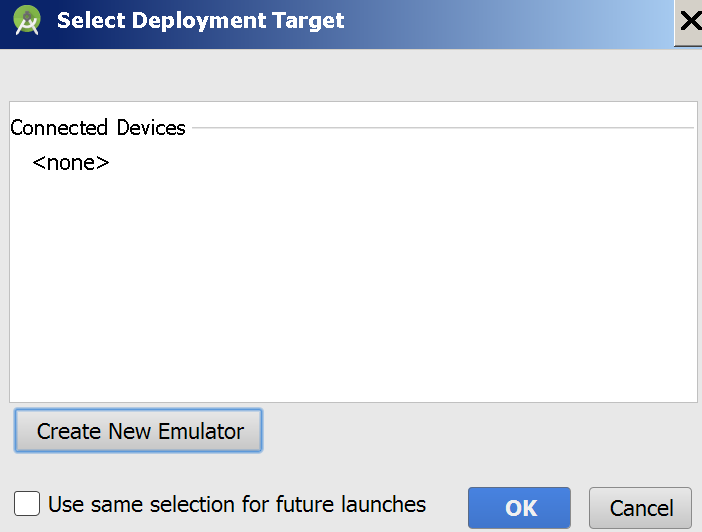
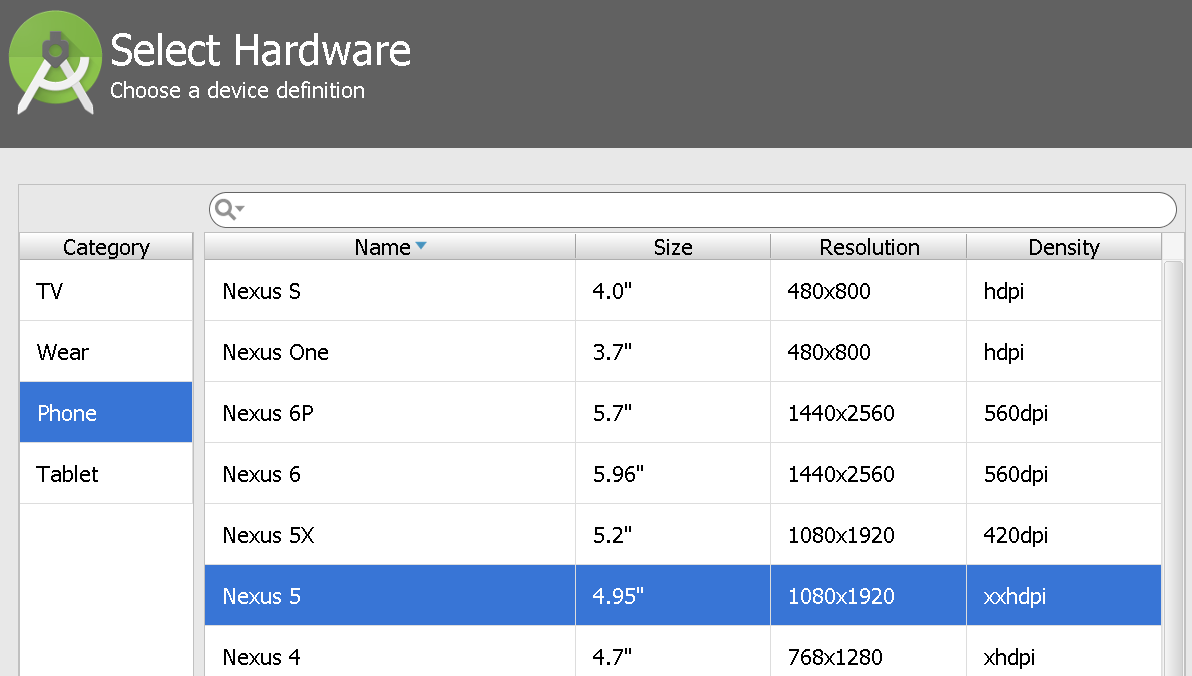
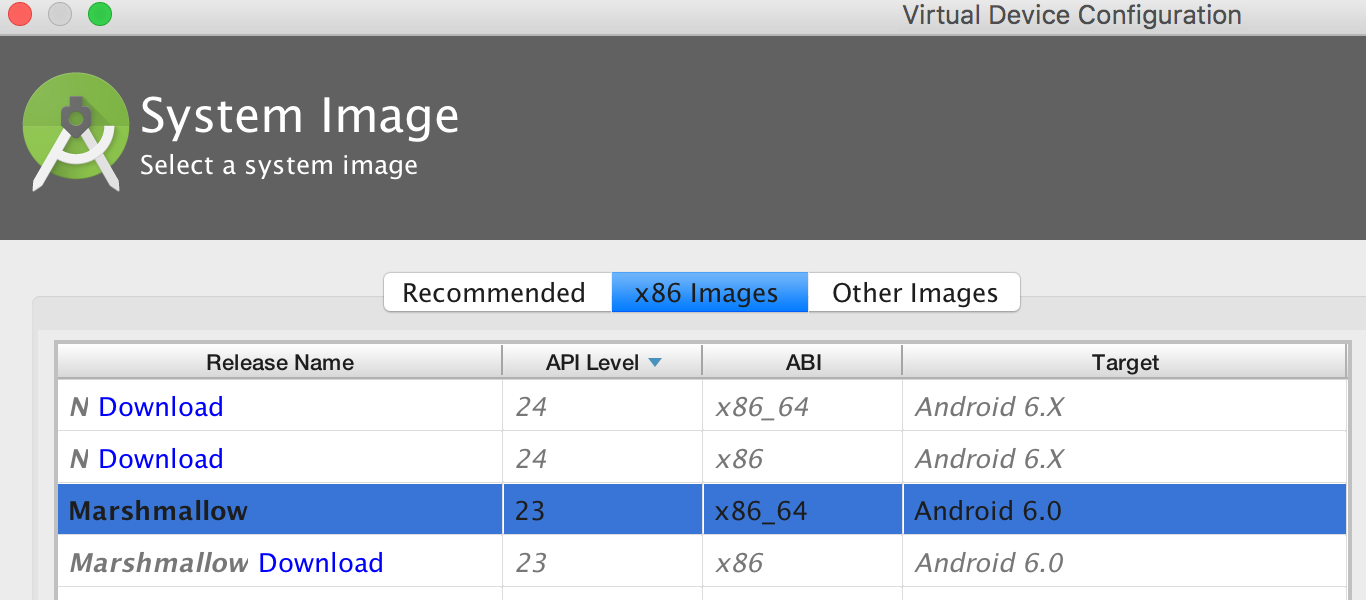
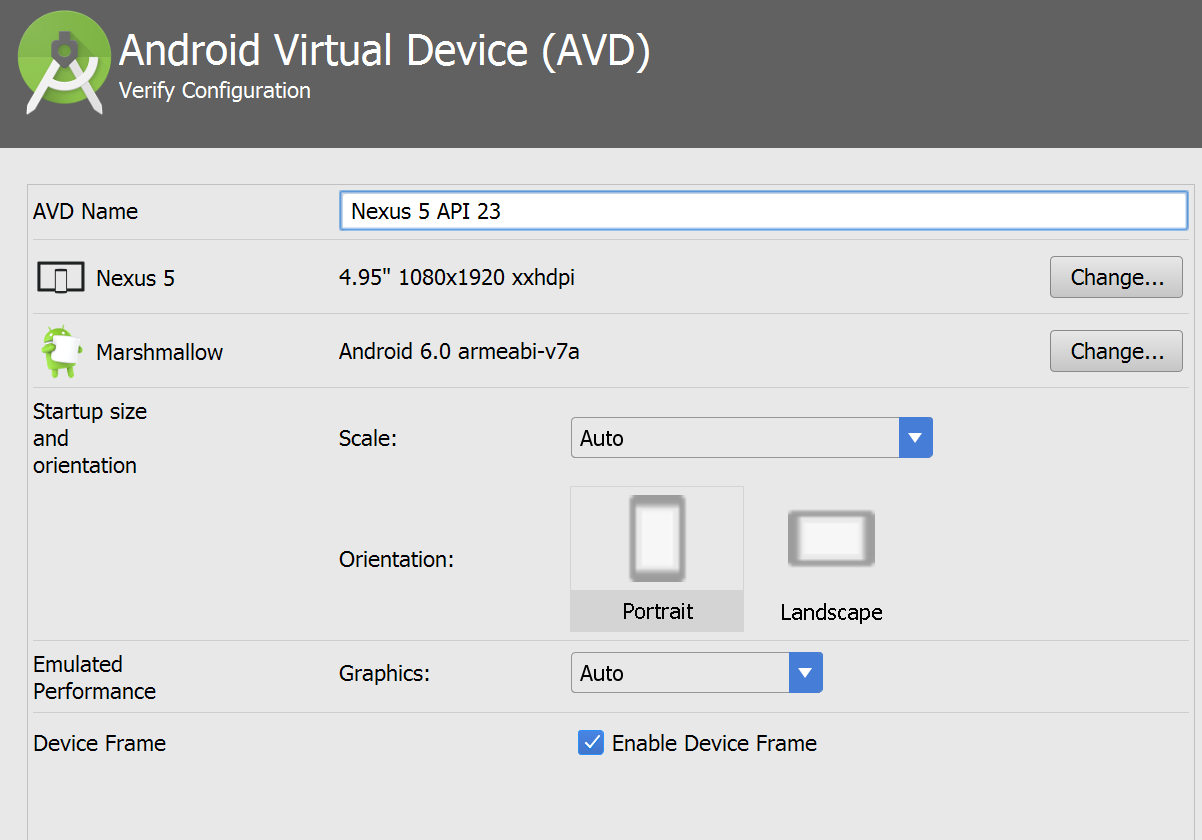


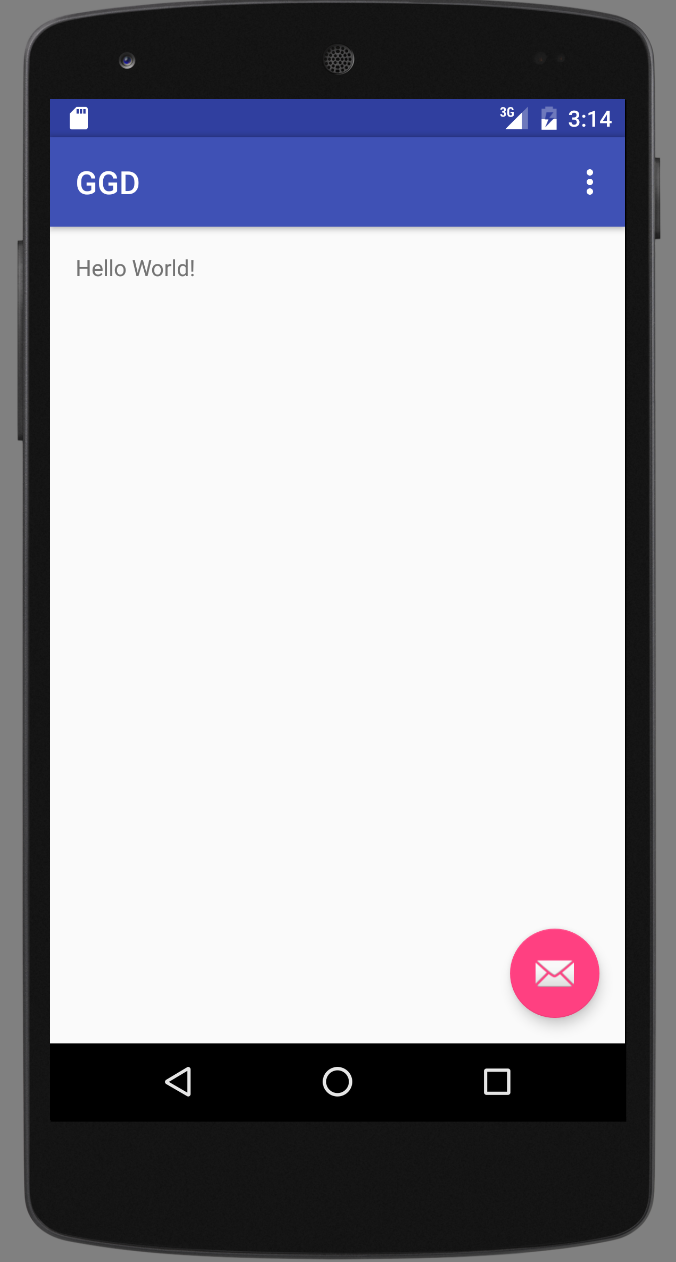
1. Name your new project “GGD” with a company domain of “healthcarebluebook.com” and select where it will live. If you choose a different name or company domain, there will be issues later on, so please choose these for simplicity’s sake.  
     
   
2. Select Phone and Tablet and API 23.



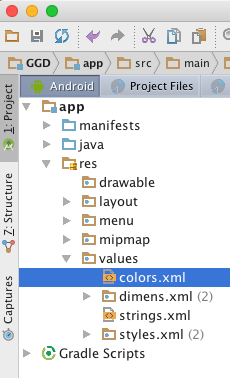
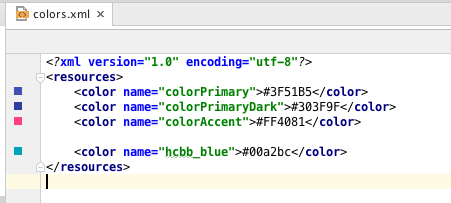
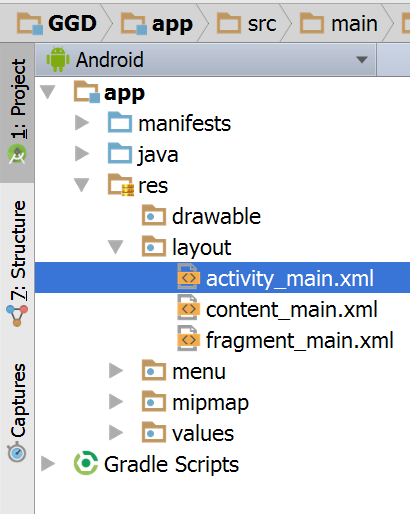
1. Select Basic Activity as the template.  
     
   
2. Make sure and check “Use a Fragment”.   
   Take the rest of the defaults as the remainder of the workshop will reference these file names.  
     
   

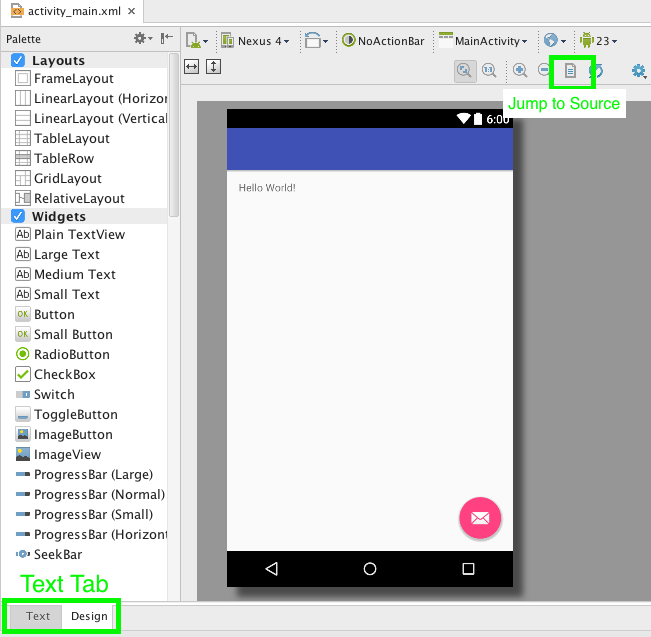
Section 2: Set up the Emulator

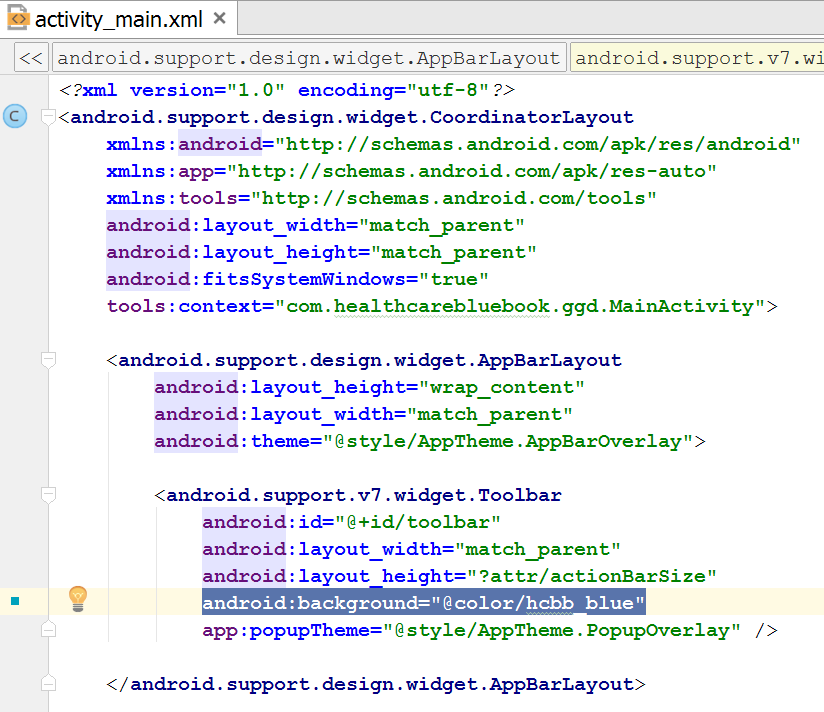
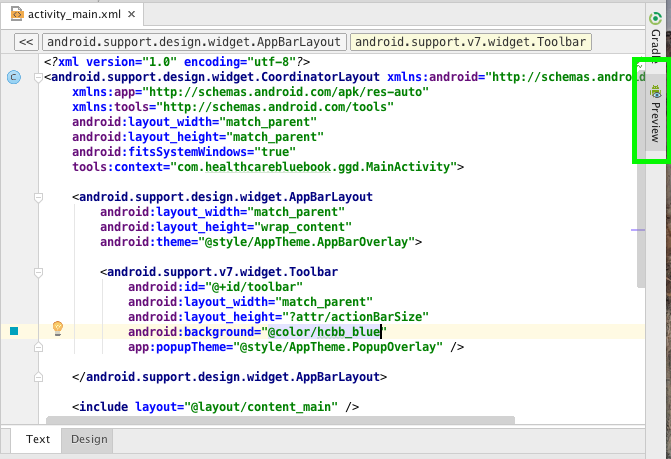
1. Click the green run button next to App on the toolbar  
     
   
2. Choose Create New Emulator  
     
   
3. Choose the Nexus 5  
     
   
4. Select the x86 Images tab, then the Marshmallow, API 23 image. Then click Next.  
     
   
5. Take the rest of the defaults, click the finish button and start the emulator by clicking the OK button.  
     
   
6. Your emulator should pop up and show the basic app. It can take a few minutes to start up the first time. So don’t close it for the duration of the workshop.

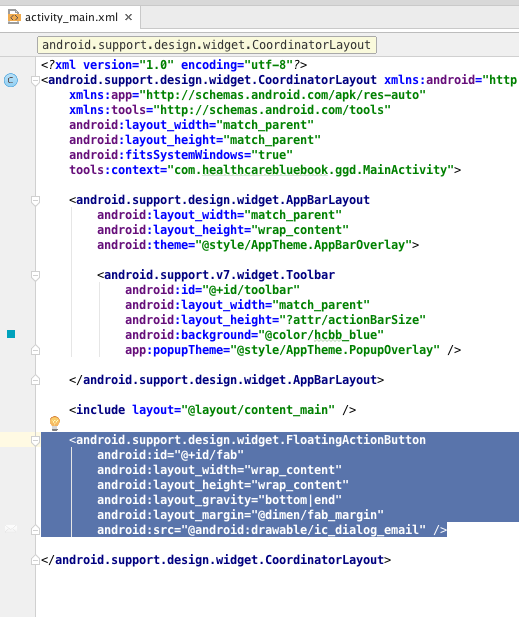
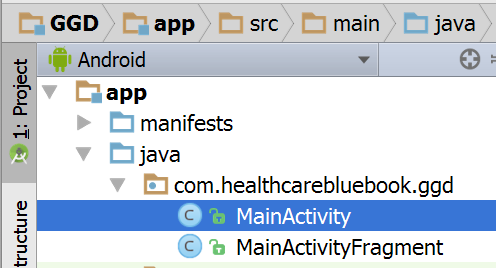


Section 3: Create the User Interface

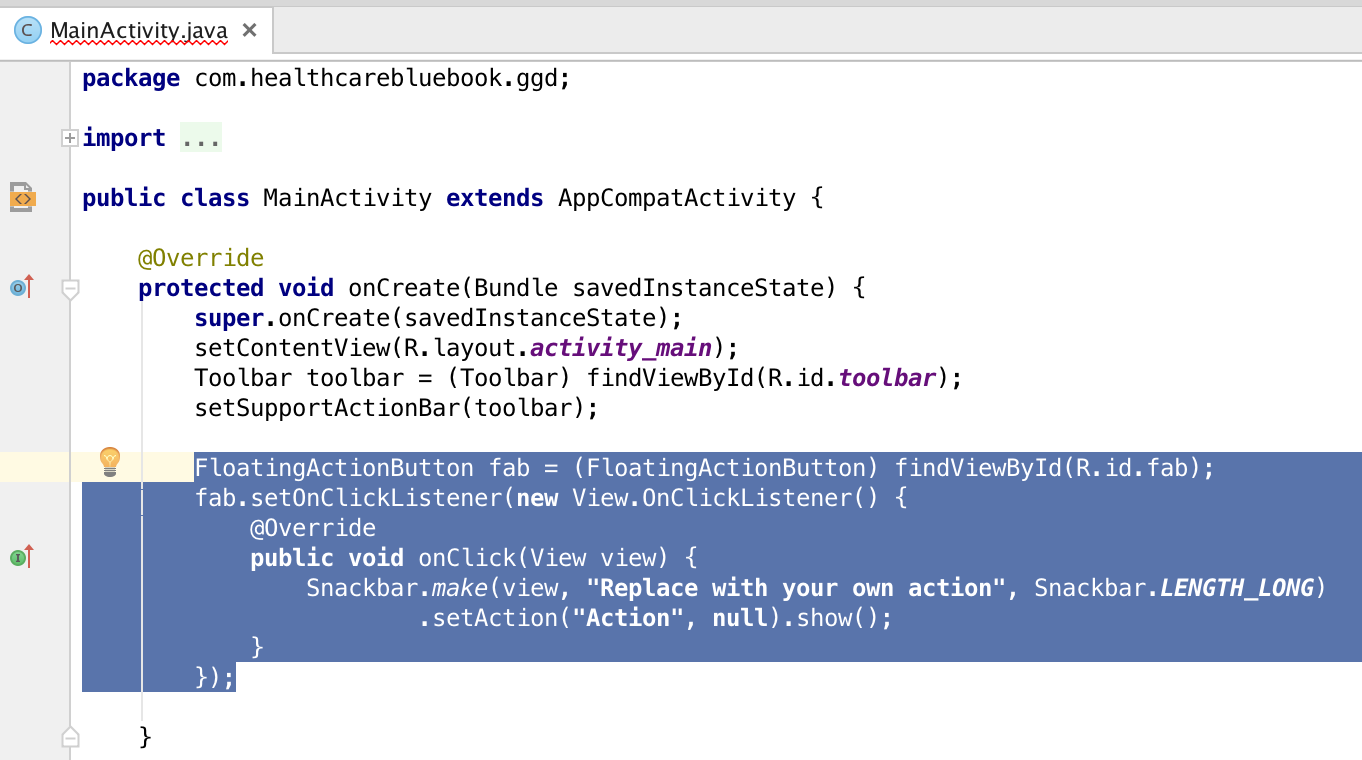
1. Change the title background to the HCBB Blue.
   1. In the Project View, open app > res > values > colors.xml  
        
      
   2. Add a new value named hcbb\_blue with a value of #00a2bc  
        
      
   3. Open the layout file  
      app > res > layout > activity\_main.xml  
        
      

If necessary, switch to the Text tab (click the Jump to Source button in the preview window).  
  


* 1. Change the value of the Toolbar background to the new hcbb\_blue color we just added.   
       
       
       
     Notice how the color changes in the preview window. (If you don’t see the preview pane, click “Preview” from the upper right region of the editor.)  
       
     

1. Remove the FloatingActionButton.
   1. In app > res > layout > activity\_main.xml, remove the FloatingActionButton xml as highlighted.  
        
      
   2. Open file app > java > com.healthcarebluebook.ggd > MainActivity.  
        
      
   3. Remove the FloatingActionButton code so we don’t have a compilation error.

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.***fab***);  
fab.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Snackbar.*make*(view, **"Replace with your own action"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Action"**, **null**).show();  
 }  
});



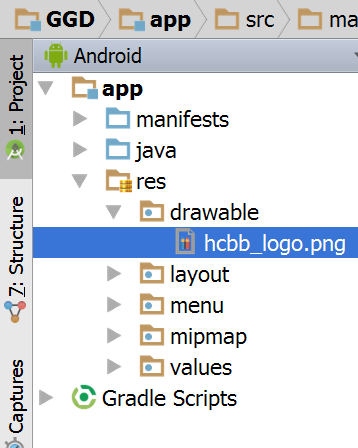
1. Checkpoint
   1. Activity\_main.xml should now look like:

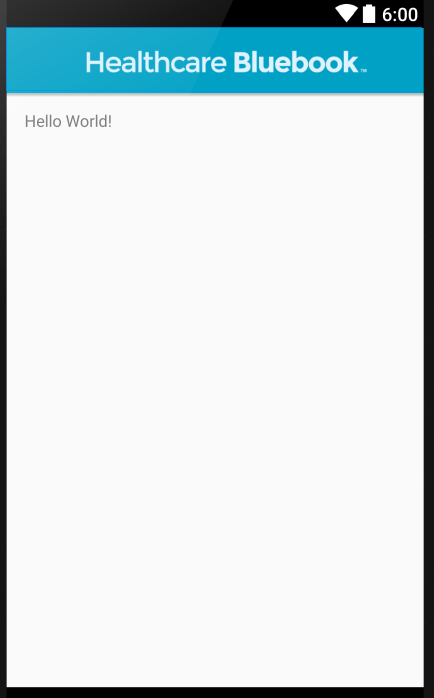
*<?***xml version="1.0" encoding="utf-8"***?>*  
<**android.support.design.widget.CoordinatorLayout**  
 **xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:app="http://schemas.android.com/apk/res-auto"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="match\_parent"**  
 **android:fitsSystemWindows="true"**  
 **tools:context="com.healthcarebluebook.ggd.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_width="match\_parent"**  
 **android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar**  
 **android:id="@+id/toolbar"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="?attr/actionBarSize"**  
 **android:background="@color/hcbb\_blue"**  
 **app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**include layout="@layout/content\_main"**/>  
  
</**android.support.design.widget.CoordinatorLayout**>

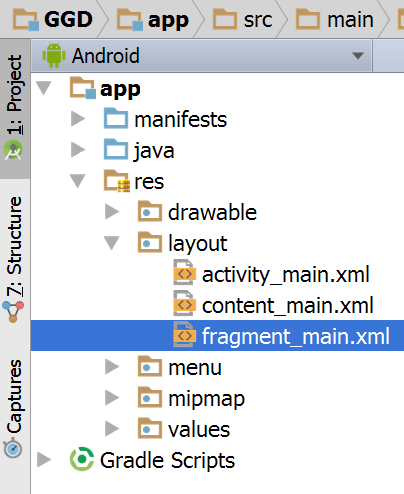
* 1. Your preview should now look like this:  
       
     

1. Add Healthcare Bluebook logo
   1. Take hcbb\_logo.png from the resources.zip file and add it to the directory:  
      {your\_project\_root\_directory}/app/src/main/res/drawable

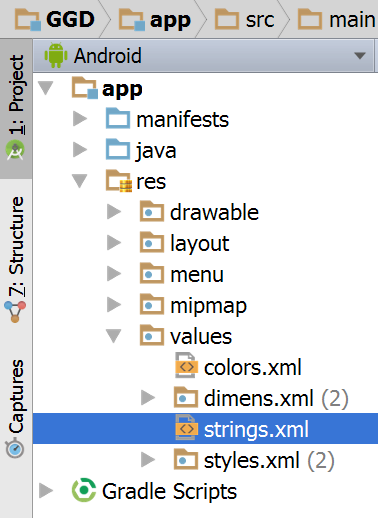
**Tip:** If you right click on the drawable folder in Android Studio, you can choose Reveal in Finder (Mac) or Show in Explorer (PC) to quickly open the directory.

After copying into the drawable folder, the png will automatically show up in Android Studio.  
  


* 1. In activity\_main.xml, add a new ImageView to the Toolbar. Use the hcbb\_logo.png as the source.  
     
  2. Preview:  
       
     

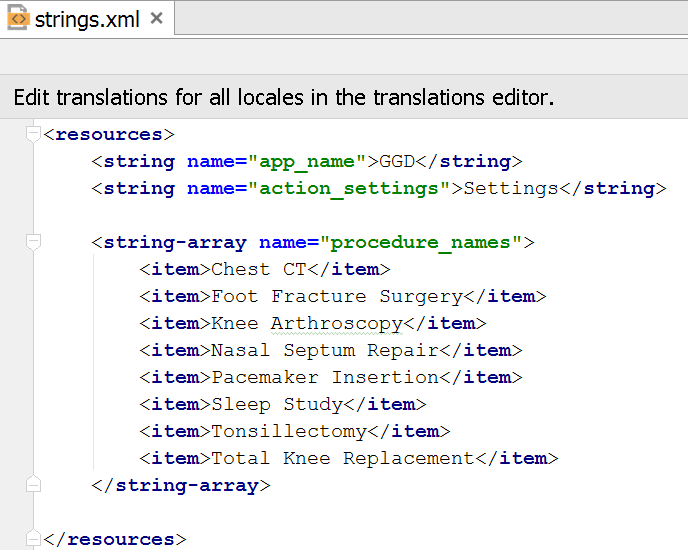
1. Slight surgery to fragment\_main.xml
   1. Open app > res > layout > fragment\_main.xml  
        
      
   2. Change RelativeLayout to “ScrollView”
   3. Remove TextView
   4. Add LinearLayout
   5. Now your fragment\_main.xml should be:

<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="match\_parent"**  
 **android:paddingLeft="@dimen/activity\_horizontal\_margin"**  
 **android:paddingRight="@dimen/activity\_horizontal\_margin"**  
 **android:paddingTop="@dimen/activity\_vertical\_margin"**  
 **android:paddingBottom="@dimen/activity\_vertical\_margin"**  
 **tools:showIn="@layout/activity\_main"**  
 **tools:context="com.healthcarebluebook.ggd.MainActivityFragment"**>  
  
 <**LinearLayout**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:orientation="vertical"**>  
  
  
 </**LinearLayout**>  
  
</**ScrollView**>

1. Add a list of procedures
   1. Open app > res > values > strings.xml  
        
      
   2. Add a string-array with the procedures we’re going to use.

<**string-array name="procedure\_names"**>  
 <**item**>Chest CT</**item**>  
 <**item**>Foot Fracture Surgery</**item**>  
 <**item**>Knee Arthroscopy</**item**>  
 <**item**>Nasal Septum Repair</**item**>  
 <**item**>Pacemaker Insertion</**item**>  
 <**item**>Sleep Study</**item**>  
 <**item**>Tonsillectomy</**item**>

<**item**>Total Knee Replacement</**item**>  
</**string-array**>



1. Add Spinner
   1. Open app > res > layout > fragment\_main.xml
   2. Add a preamble TextView to the Linear Layout with green text “The Fair Price is the price that a person can expect to pay by being a prudent healthcare consumer.”

<**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginBottom="20dp"**  
 **android:textSize="20dp"**  
 **android:textColor="@android:color/holo\_green\_dark"**  
 **android:text="The Fair Price is the price that a person can expect to pay by being a prudent healthcare consumer."** />

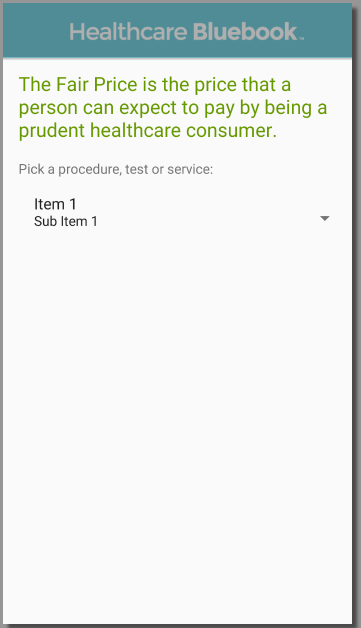
* 1. Add another TextView with instructions “Pick a procedure, test or service:”

<**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:text="Pick a procedure, test or service:"** />

* 1. Add a Spinner control and hook it up to the procedure\_names aray we created in step 6.

<**Spinner**  
 **android:id="@+id/procedure\_name\_spinner"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:spinnerMode="dialog"**  
 **android:entries="@array/procedure\_names"**  
/>

* 1. Preview should look like:



* 1. Fragment\_main.xml should be:

<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="match\_parent"**  
 **android:paddingLeft="@dimen/activity\_horizontal\_margin"**  
 **android:paddingRight="@dimen/activity\_horizontal\_margin"**  
 **android:paddingTop="@dimen/activity\_vertical\_margin"**  
 **android:paddingBottom="@dimen/activity\_vertical\_margin"**  
 **tools:showIn="@layout/activity\_main"**  
 **tools:context="com.healthcarebluebook.ggd.MainActivityFragment"**>  
  
 <**LinearLayout**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:orientation="vertical"**>  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginBottom="20dp"**  
 **android:textSize="20dp"**  
 **android:textColor="@android:color/holo\_green\_dark"**  
 **android:text="The Fair Price is the price that a person can expect to pay by being a prudent healthcare consumer."** />  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:text="Pick a procedure, test or service:"** />  
  
 <**Spinner**  
 **android:id="@+id/procedure\_name\_spinner"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:spinnerMode="dialog"**  
 **android:entries="@array/procedure\_names"**  
/>  
 </**LinearLayout**>  
  
</**ScrollView**>

1. Add zip code text box (still working in fragment\_main.xml)
   1. Add TextView with instructions just under the Spinner

<**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:text="Enter zip code:"** />

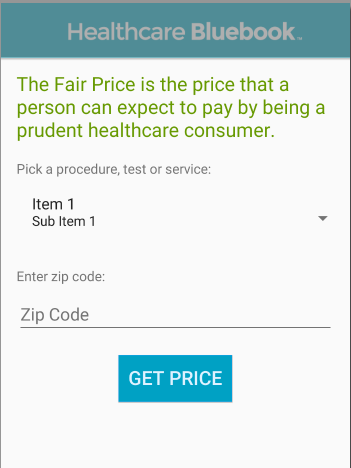
* 1. Add EditText to capture the zip code input

<**EditText**  
 **android:id="@+id/zip\_code"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:hint="Zip Code"**  
 **android:inputType="number"**/>

1. Add Button to submit the procedure and zip code

<**Button**  
 **android:id="@+id/get\_price\_button"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:layout\_gravity="center"**  
 **android:textSize="20dp"**  
 **android:text="Get Price"**  
 **android:padding="10dp"**  
 **android:background="@color/hcbb\_blue"**  
 **android:textColor="@android:color/white"**  
/>

1. Check point
   1. Preview:



* 1. Fragment\_main.xml:

<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="match\_parent"**  
 **android:paddingLeft="@dimen/activity\_horizontal\_margin"**  
 **android:paddingRight="@dimen/activity\_horizontal\_margin"**  
 **android:paddingTop="@dimen/activity\_vertical\_margin"**  
 **android:paddingBottom="@dimen/activity\_vertical\_margin"**  
 **tools:showIn="@layout/activity\_main"**  
 **tools:context="com.healthcarebluebook.ggd.MainActivityFragment"**>  
  
 <**LinearLayout**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:orientation="vertical"**>  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginBottom="20dp"**  
 **android:textSize="20dp"**  
 **android:textColor="@android:color/holo\_green\_dark"**  
 **android:text="The Fair Price is the price that a person can expect to pay by being a prudent healthcare consumer."** />  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:text="Pick a procedure, test or service:"** />  
  
 <**Spinner**  
 **android:id="@+id/procedure\_name\_spinner"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:spinnerMode="dialog"**  
 **android:entries="@array/procedure\_names"**  
/>  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:text="Enter zip code:"** />  
  
 <**EditText**  
 **android:id="@+id/zip\_code"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:hint="Zip Code"**  
 **android:inputType="number"**/>  
  
 <**Button**  
 **android:id="@+id/get\_price\_button"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:layout\_gravity="center"**  
 **android:textSize="20dp"**  
 **android:text="Get Price"**  
 **android:padding="10dp"**  
 **android:background="@color/hcbb\_blue"**  
 **android:textColor="@android:color/white"**  
/>  
  
 </**LinearLayout**>  
  
</**ScrollView**>

* 1. **Run in emulator**

1. Last UI piece: Add where we will show the Fair Price for our selected procedure and zip code.  
   1. In fragment\_main.xml, add a TextView below the Button. The text doesn’t matter at this point.  
        
      <**TextView**

**android:id="@+id/results\_text"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_marginTop="40dp"**

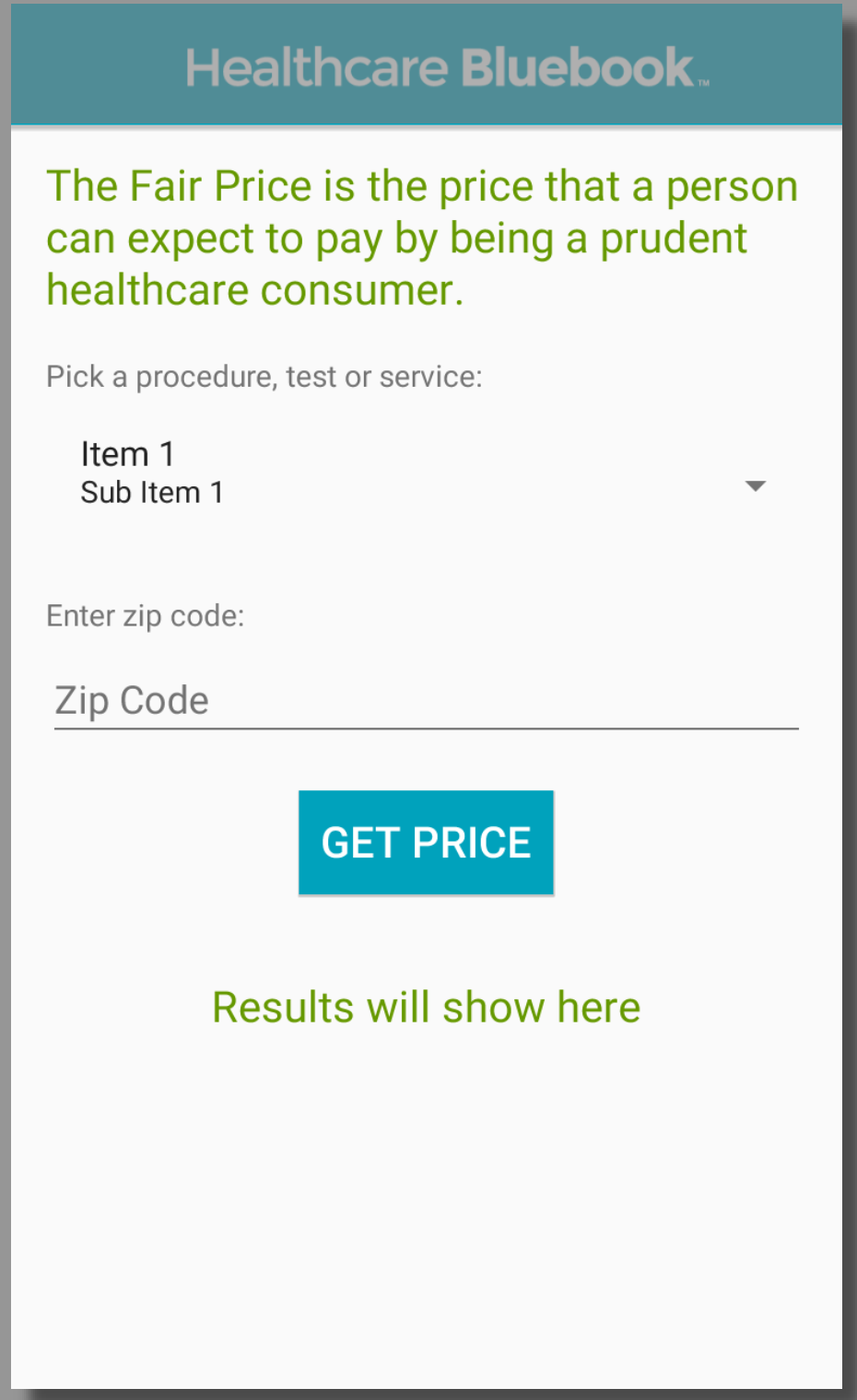
**android:textSize="20dp"**

**android:textColor="@android:color/holo\_green\_dark"**

**android:text="Results will show here"**

**android:layout\_gravity="center"**/>

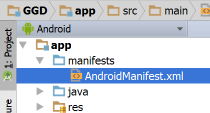
* 1. Preview:



* 1. Fragment\_main.xml

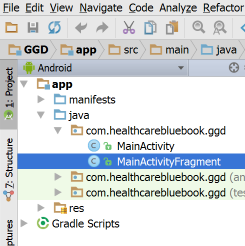
<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  
 **xmlns:tools="http://schemas.android.com/tools"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="match\_parent"**  
 **android:paddingLeft="@dimen/activity\_horizontal\_margin"**  
 **android:paddingRight="@dimen/activity\_horizontal\_margin"**  
 **android:paddingTop="@dimen/activity\_vertical\_margin"**  
 **android:paddingBottom="@dimen/activity\_vertical\_margin"**  
 **tools:showIn="@layout/activity\_main"**  
 **tools:context="com.healthcarebluebook.ggd.MainActivityFragment"**>  
 <**LinearLayout**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:orientation="vertical"**>  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginBottom="20dp"**  
 **android:textSize="20dp"**  
 **android:textColor="@android:color/holo\_green\_dark"**  
 **android:text="The Fair Price is the price that a person can expect to pay by being a prudent healthcare consumer."** />  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:text="Pick a procedure, test or service:"** />  
  
 <**Spinner**  
 **android:id="@+id/procedure\_name\_spinner"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:spinnerMode="dialog"**  
 **android:entries="@array/procedure\_names"**  
/>  
  
 <**TextView**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:text="Enter zip code:"** />  
  
 <**EditText**  
 **android:id="@+id/zip\_code"**  
 **android:layout\_width="match\_parent"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="10dp"**  
 **android:hint="Zip Code"**  
 **android:inputType="number"**/>  
  
 <**Button**  
 **android:id="@+id/get\_price\_button"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="20dp"**  
 **android:layout\_gravity="center"**  
 **android:textSize="20dp"**  
 **android:text="Get Price"**  
 **android:padding="10dp"**  
 **android:background="@color/hcbb\_blue"**  
 **android:textColor="@android:color/white"**  
/>  
 <**TextView**  
 **android:id="@+id/results\_text"**  
 **android:layout\_width="wrap\_content"**  
 **android:layout\_height="wrap\_content"**  
 **android:layout\_marginTop="40dp"**  
 **android:textSize="20dp"**  
 **android:textColor="@android:color/holo\_green\_dark"**  
 **android:text="Results will show here"**  
 **android:layout\_gravity="center"**/>  
 </**LinearLayout**>

</**ScrollView**>

1. If you notice, the keyboard opens automatically when you run the emulator. We don’t really want the keyboard to show up until the user activates the zip code box.
   1. Open the AndroidManifest.xml file  
        
      
   2. Add the following highlighted option to the activity tag:  
        
      <**activity**  
       **android:name=".MainActivity"**  
       **android:label="@string/app\_name"**  
       **android:theme="@style/AppTheme.NoActionBar"**  
       **android:windowSoftInputMode="stateHidden"**  
      >  
        
       <**intent-filter**>  
       <**action android:name="android.intent.action.MAIN"** />  
        
       <**category android:name="android.intent.category.LAUNCHER"** />  
       </**intent-filter**>  
      </**activity**>
   3. Run the app again and see that the keyboard only shows when the zip code field is tapped.

Section 4: Make the App DO something

1. Open app > java > com.healthcarebluebook.ggd > MainActivityFragment



1. Add private variables directly after the first opening curly bracket {.   
     
   Tip: If you type these in, Android Studio will add the import statements for you by popping up a dropdown box or allowing you to use the keyboard shortcut alt-Enter to add the import statement.

Tip 2: If you copy/paste, you will need to manually add the import statements or you can put your cursor over each type (Spinner, EditText, Button, TextView) and hit alt-Enter to accept an automatic import

**private** Spinner **procedureNameSpinner**;

**private** EditText **zipCodeTextView**;

**private** Button **getPriceButton**;

**private** TextView **resultsTextView**;

1. In the OnCreateView method, remove the single line of code and replace it with the following lines.

View fragmentMain = inflater.inflate(R.layout.***fragment\_main***, container, **false**);  
*// All code goes under this comment  
  
// All code goes above this comment***return** fragmentMain;

1. Assign the controls we added in fragment\_main.xml to the private variables we just created. Do this after the “View fragmentMain =” statement and before the return statement (between the comment lines).

**procedureNameSpinner** = (Spinner) fragmentMain.findViewById(R.id.***procedure\_name\_spinner***);

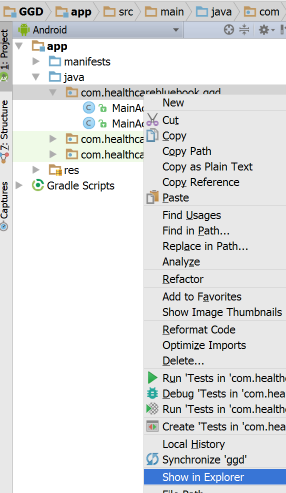
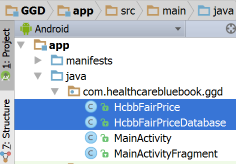
**zipCodeTextView** = (EditText) fragmentMain.findViewById(R.id.***zip\_code***);

**getPriceButton** = (Button) fragmentMain.findViewById(R.id.***get\_price\_button***);

**resultsTextView** = (TextView) fragmentMain.findViewById(R.id.***results\_text***);

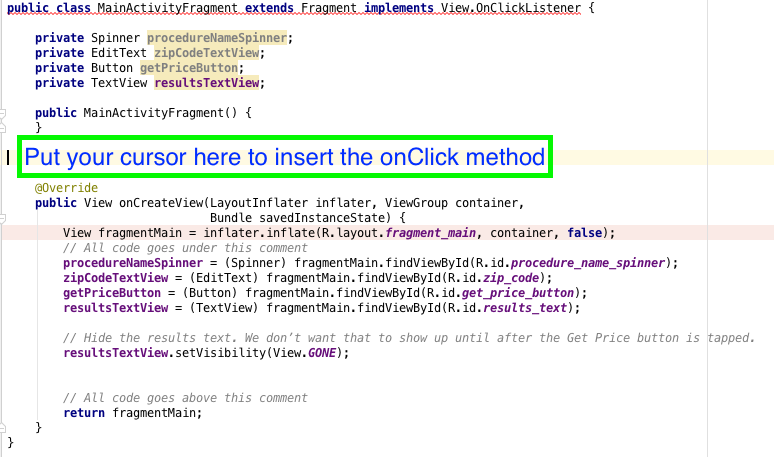
1. Hide the results text. We don’t want that to show up until after the Get Price button is tapped.

**resultsTextView**.setVisibility(View.***GONE***);

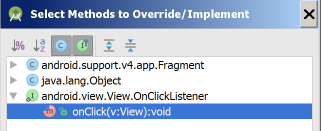
1. Install the HcbbFairPriceDatabase.  
   1. From the resources.zip file, find HcbbFairPrice.java and HcbbFairPriceDatabase.java and copy them to the folder {your\_project\_root\_directory}\ app\src\main\java\com\healthcarebluebook\ggd  
        
      You can get to this directly quickly from Android Studio by right-clicking on com.healthcarebluebook.ggd and selecting Show in Explorer (Windows) or Reveal in Finder (Mac)  
        
      
   2. Once you copy those files to the folder, Android Studio should pick them up  
        
      
2. Show the Fair Price results
   1. Open app > java > com.healthcarebluebook.ggd > MainActivityFragment
   2. Open Make MainActivityFragment implement the OnClickListener interface. This will allow our class to handle the clicks on the Get Price button. Add the highlighted text to the declaration of the class.

**public class** MainActivityFragment **extends** Fragment **implements** View.OnClickListener {

* 1. Override the onClick method to comply with the OnClickListener interface. Position your mouse between the MainActivityFragment constructor and the onCreateView method.



* 1. Use the keyboard shortcut Control-O to get a list of methods we can override. Select the onClick method for OnClickListener.



This will add the following method stub at your cursor location.

@Override  
**public void** onClick(View v) {  
   
}

* 1. Now that you have the method stub, we can start adding stuff to do when the button is tapped (clicked). The following steps should all be done within the onClick method we created in step b.
  2. Add an if statement to make sure we only do stuff if the button is clicked and not anything else. All the steps below will put code in the if block. So your onClick method should look like this:

@Override  
**public void** onClick(View v) {  
 **if** (v.getId() == R.id.***get\_price\_button***) {  
  
 }  
}

* 1. First things first. Close the keyboard since we don’t need it anymore and the user likely has it open from putting the zip code in. Be sure to create the import statements for InputMethodManager and Context by putting your cursor over a red class name and tapping alt-Enter.

InputMethodManager imm = (InputMethodManager) getActivity().getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  
imm.hideSoftInputFromWindow(**zipCodeTextView**.getWindowToken(), 0);

* 1. Get the procedure the user selected in the Spinner

String procedureName = (String) **procedureNameSpinner**.getSelectedItem();

* 1. Get the zip code the user entered

String zipCode = **zipCodeTextView**.getText().toString();

* 1. Query the HcbbFairPriceDatabase for the Fair Price for this procedure/zip combination.

HcbbFairPrice fairPrice = HcbbFairPriceDatabase.*GetHcbbFairPrice*(procedureName, zipCode);

* 1. Use String.format to put the Fair Price value into a predetermined template.

String result = String.*format*(**"An informed consumer can expect to pay %s for a %s in zip code %s."**, fairPrice.getFairPrice(), fairPrice.getProcedureName(), fairPrice.getZipCode());

* 1. Set the resultsTextView text to the String we composed in step g.

**resultsTextView**.setText(result);

* 1. Show the resultsTextView. Remember, we hid it initially, so we have to make it show it again.

**resultsTextView**.setVisibility(View.***VISIBLE***);

1. Lastly, in the **onCreateView** method (not the onClick method we’ve been working in), add the following line just before the return statement. This will tell the Get Price Button that this class' onClick method will handle any click (tap) events.

**getPriceButton**.setOnClickListener(**this**);

1. Your entire MainActivityFragment.java file should now be this:

**package** com.healthcarebluebook.ggd;  
  
**import** android.content.Context;  
**import** android.support.v4.app.Fragment;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.view.inputmethod.InputMethodManager;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
  
*/\*\**  
 *\* A fragment containing a simple view.*  
 *\*/*  
**public class** MainActivityFragment **extends** Fragment **implements** View.OnClickListener {  
  
 **private** Spinner **procedureNameSpinner**;  
 **private** EditText **zipCodeTextView**;  
 **private** Button **getPriceButton**;  
 **private** TextView **resultsTextView**;  
  
 **public** MainActivityFragment() {  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 *// make sure we are responding to the get price button and not anything else*  
**if** (v.getId() == R.id.***get\_price\_button***) {  
 *// close the keyboard*  
InputMethodManager imm = (InputMethodManager) getActivity().getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  
 imm.hideSoftInputFromWindow(**zipCodeTextView**.getWindowToken(), 0);  
 *// get the selected procedure name*  
String procedureName = (String) **procedureNameSpinner**.getSelectedItem();  
 *// get the zip code*  
String zipCode = **zipCodeTextView**.getText().toString();  
 *// query the "database" for the fair price for this procedure and this zip*  
HcbbFairPrice fairPrice = HcbbFairPriceDatabase.*GetHcbbFairPrice*(procedureName, zipCode);  
  
 String result = String.*format*(**"An informed consumer can expect to pay %s for a %s in zip code %s."**, fairPrice.getFairPrice(), fairPrice.getProcedureName(), fairPrice.getZipCode());  
  
 **resultsTextView**.setText(result);  
 **resultsTextView**.setVisibility(View.***VISIBLE***);  
 }  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 **final** View fragmentMain = inflater.inflate(R.layout.***fragment\_main***, container, **false**);  
  
 *// assign our private variables to the controls we defined in fragment\_main.xml so we can*  
 *// do stuff with them.*  
**procedureNameSpinner** = (Spinner) fragmentMain.findViewById(R.id.***procedure\_name\_spinner***);  
 **zipCodeTextView** = (EditText) fragmentMain.findViewById(R.id.***zip\_code***);  
 **getPriceButton** = (Button) fragmentMain.findViewById(R.id.***get\_price\_button***);  
 **resultsTextView** = (TextView) fragmentMain.findViewById(R.id.***results\_text***);  
  
 *// Hide the results text. We don't want that to show up until after the Get Price button is tapped*  
**resultsTextView**.setVisibility(View.***GONE***);  
  
 *// tell the button we will use this class' onClick method to handle any taps/clicks*  
**getPriceButton**.setOnClickListener(**this**);  
  
 **return** fragmentMain;  
 }  
  
  
  
}

1. Run your app in the Emulator!  
     
   Valid Zip Codes: 37211, 00901, 99501, 98174  
   Zip Code Validation is left as an exercise for the reader.  
     
     
   