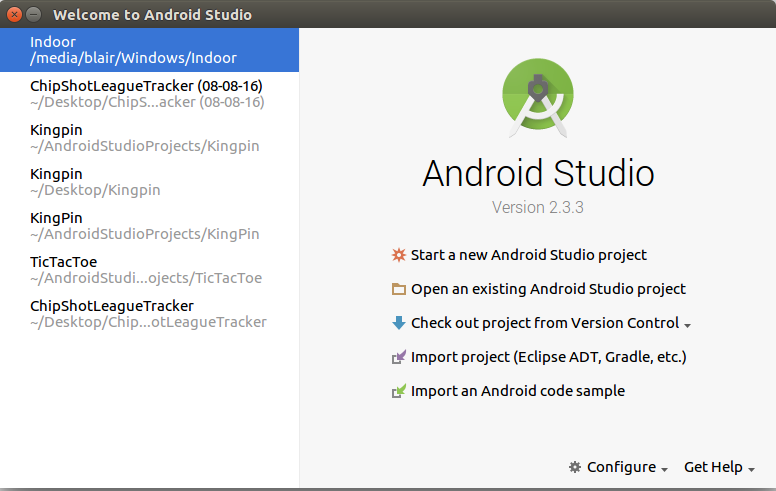
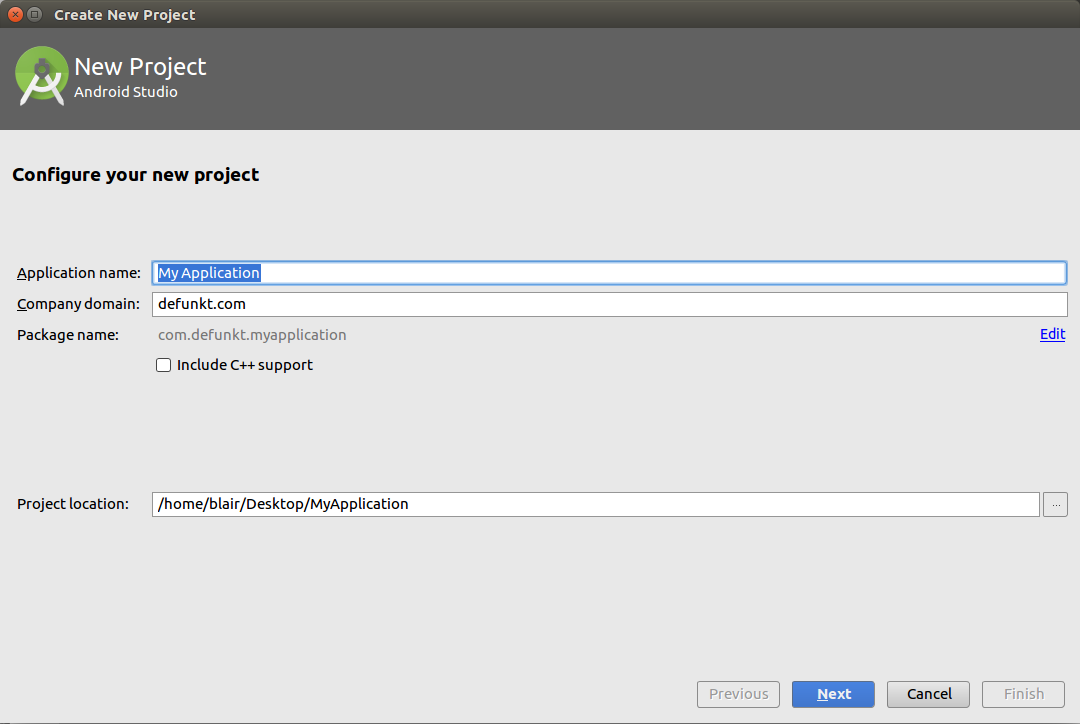
1. Download Android Studio 2.3.x if you haven’t already.

2. Once it downloads you should open to something that looks like this:

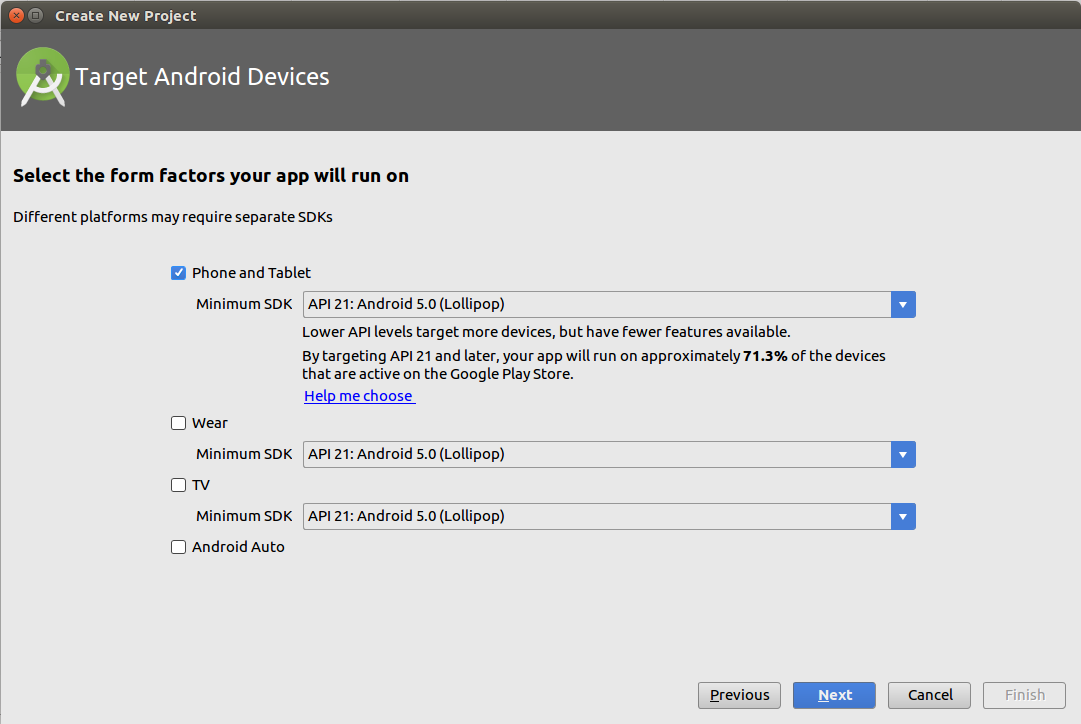


Click on, “Start a new Android Studio project.”

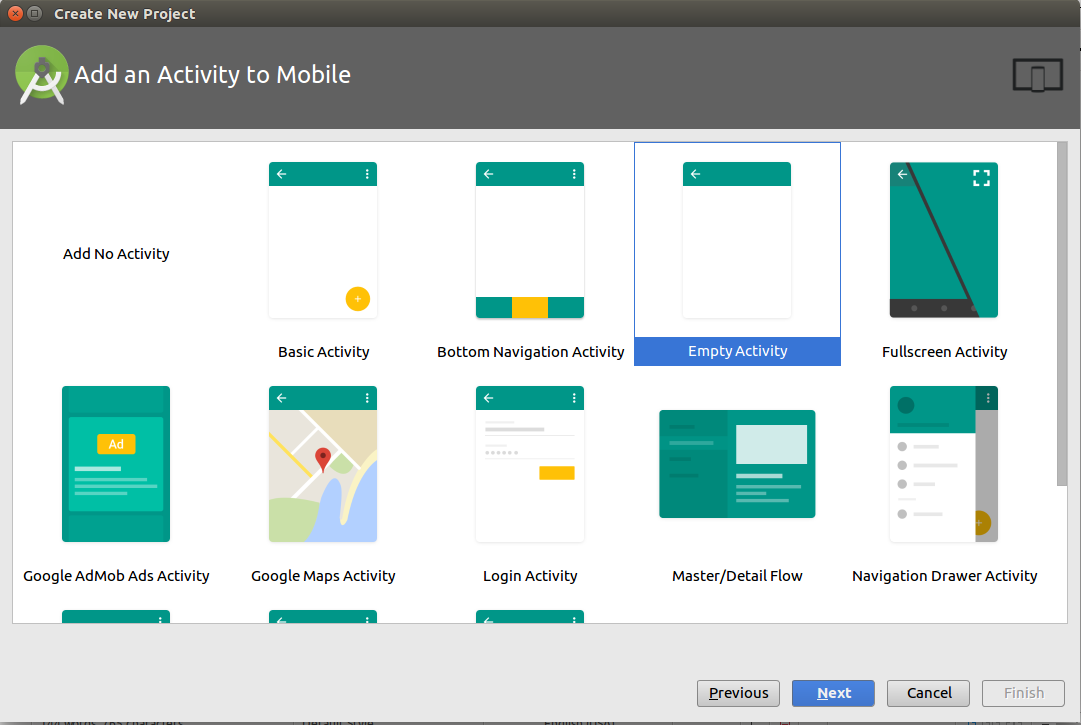
It will open to a different window featured below:



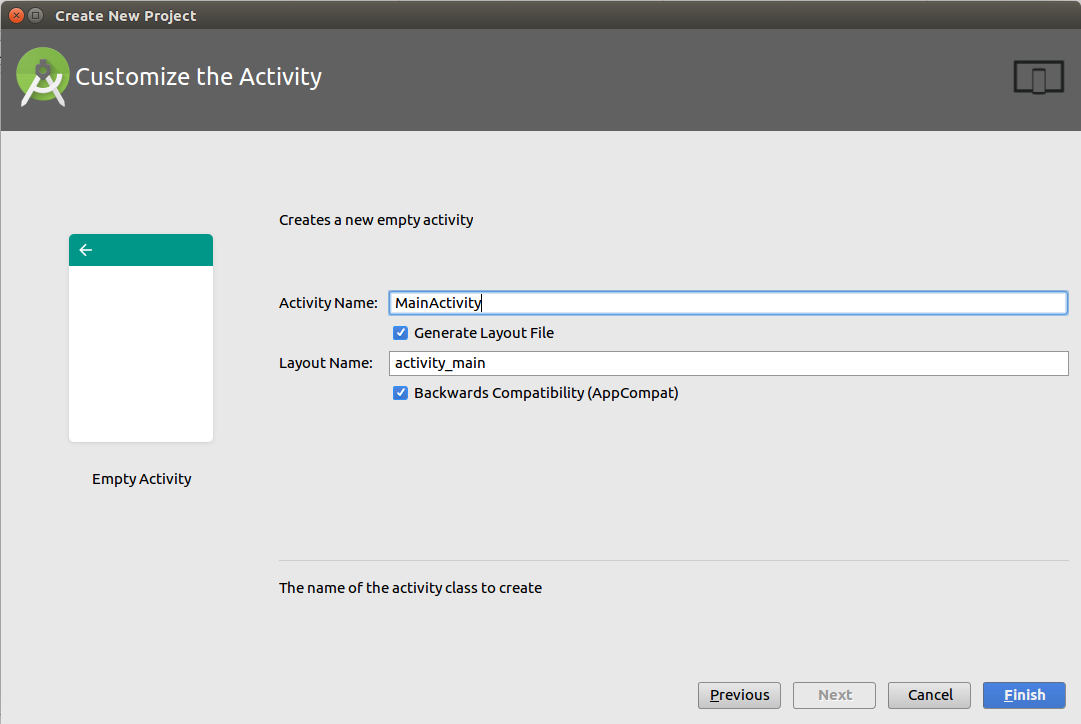
The application name is whatever you’re going to name it. In this case I used the name Indoor. The company domain is what ever you want it to be. As far as I know it’s only purpose is for the creation of the package name. Choose your own location for the file and click next.



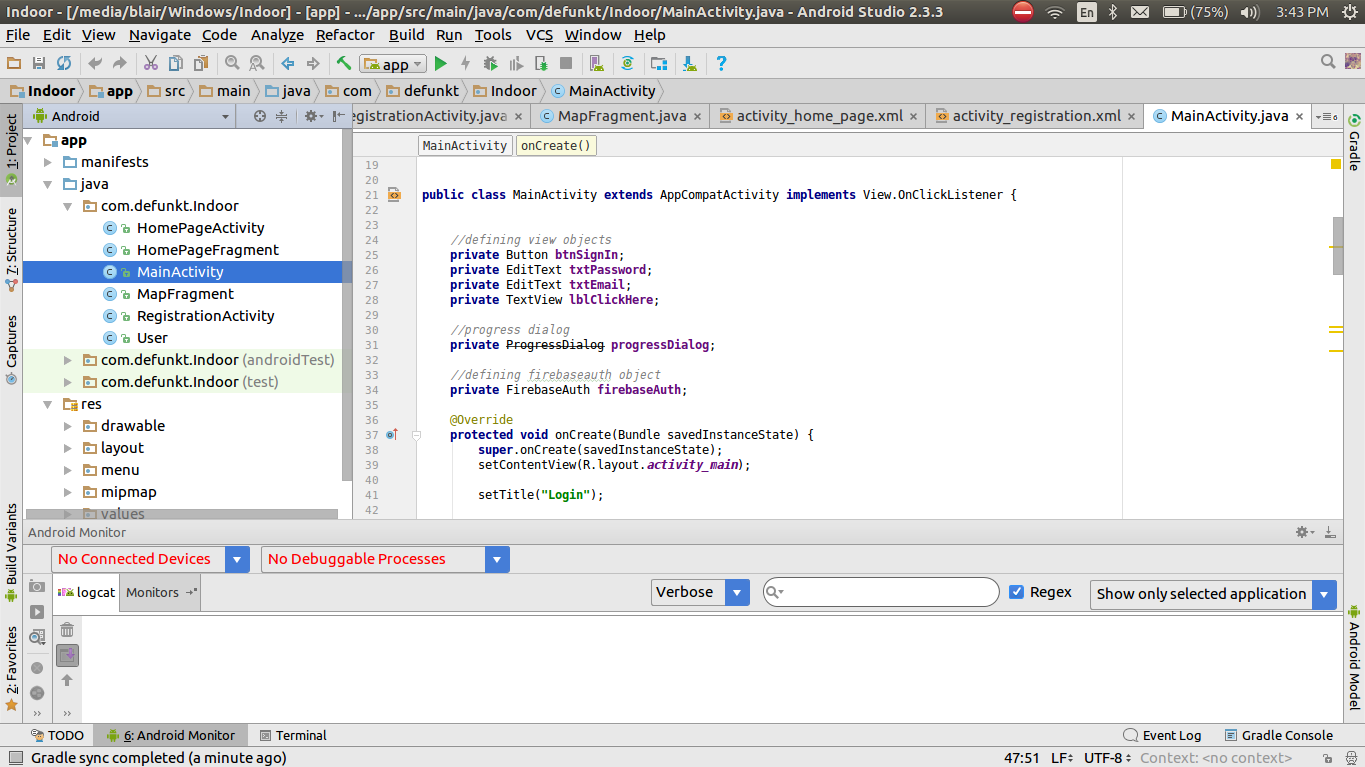
Here we have the next window where we choose what platform we want our app to run on. I’ve chosen Android 5.0 Lollipop which is 71% of the android market. Make sure you have phone and tablet checked in the checkbox as pictured above and click next.



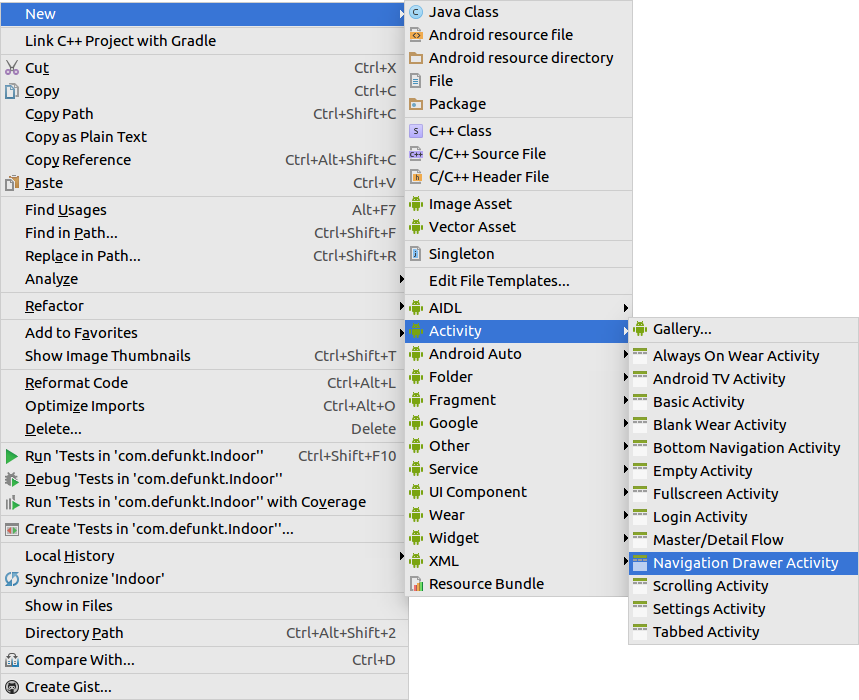
We’ll come to this screen. This is where we choose what kind of activity we want to create. We’re doing the login screen here. We could choose login activity, but it comes with a ton of features that we may/may not even implement. Instead I have chosen empty activity and implement the login screen from scratch. Make sure Empty activity is highlighted and click next.



Leave the defaults and just click finish. Android Studio should open up and start you out in the main activity.



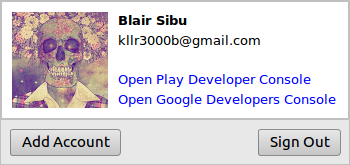
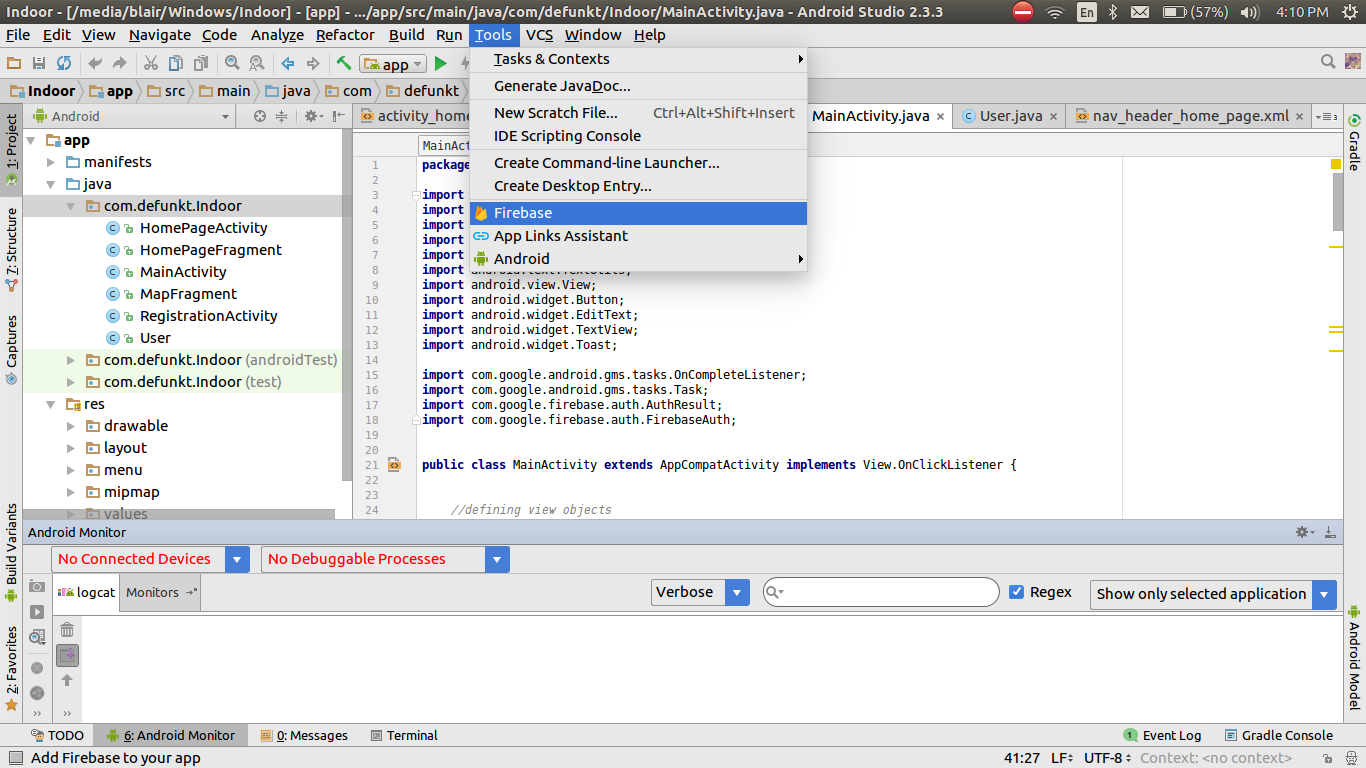
It’s not going to look like this right off the bat but that’s what this is for, the set up. So let’s get to it. Right click on whatever you decided to name your package. In this case we would right click on, “com.defunkt.Indoor.” Make sure that it’s the first one that your right click on, otherwise you would be adding an activity in the wrong directory.



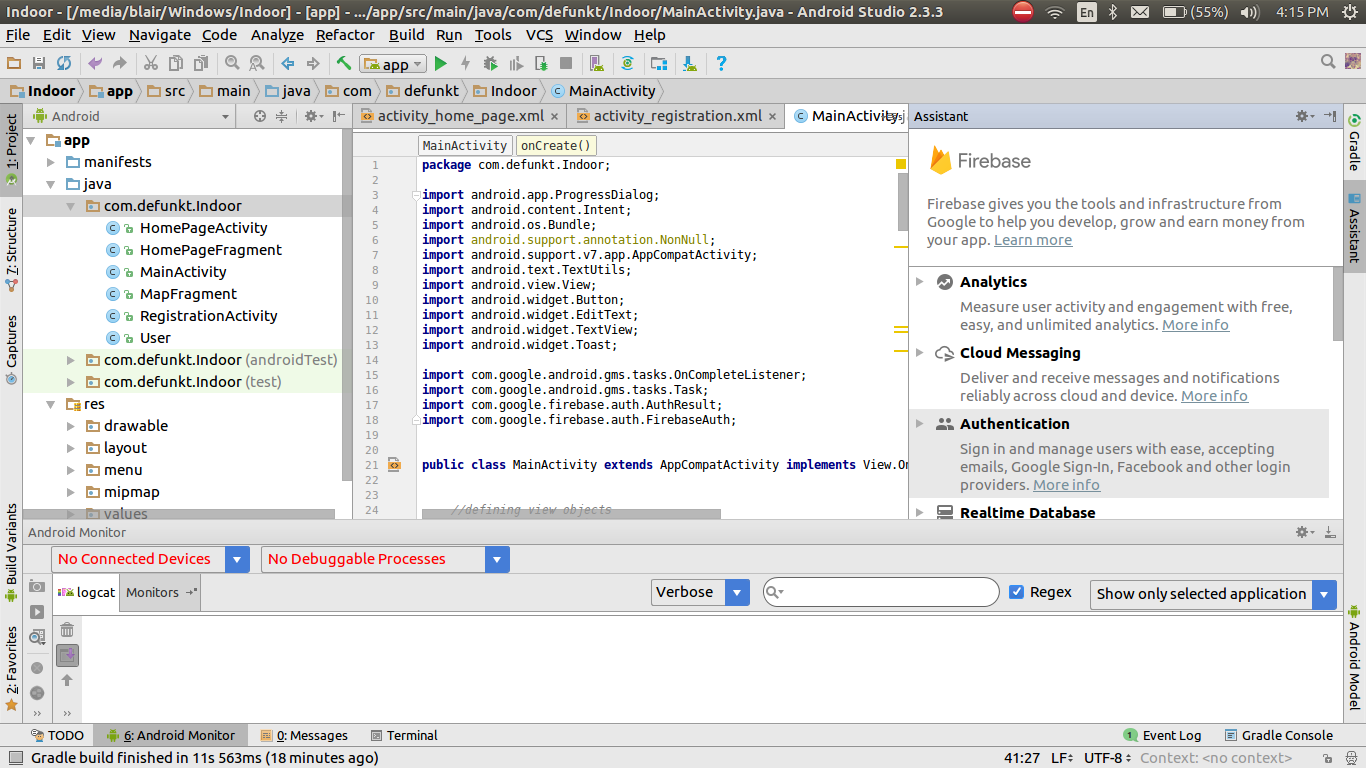
After you’ve right clicked on that package a menu will pop up. Navigate to New>Activity>Navigation Drawer. This will create the drop down menu that we need. I’ve gone ahead and named this activity HomePageActivity. Let’s almost the same thing to create the other activities. We’ll worry about adding the code and styles later. Right click on the package and bring the menu up again.

This time we’ll navigate to New>Activity>Empty Activity. This will be our RegistrationActivity. Next we’ll make the fragments. Fragments are activities that will float/overlay over our drop down menu activity. This ensures that we’ll always have a drop down menu no matter where we are in the app. Just as we created the activities we will right click on our package name.   
  
Navigate to New>Fragment>Blank Fragment. I named the first fragment HomePageFragment. Let’s also make the MapFragment in the same fashion. You’ll notice that it will generate a bunch of junk code inside the fragment, we’ll get rid of all that later. After you’ve generated the fragments let’s generate the java class User, in the same way we will right click on the package name and navigate to New>Java Class. Now we have the structure that we will need to make the app.

Next we can set up firebase. Lucky for us the new Android Studio has a wizard to do that.

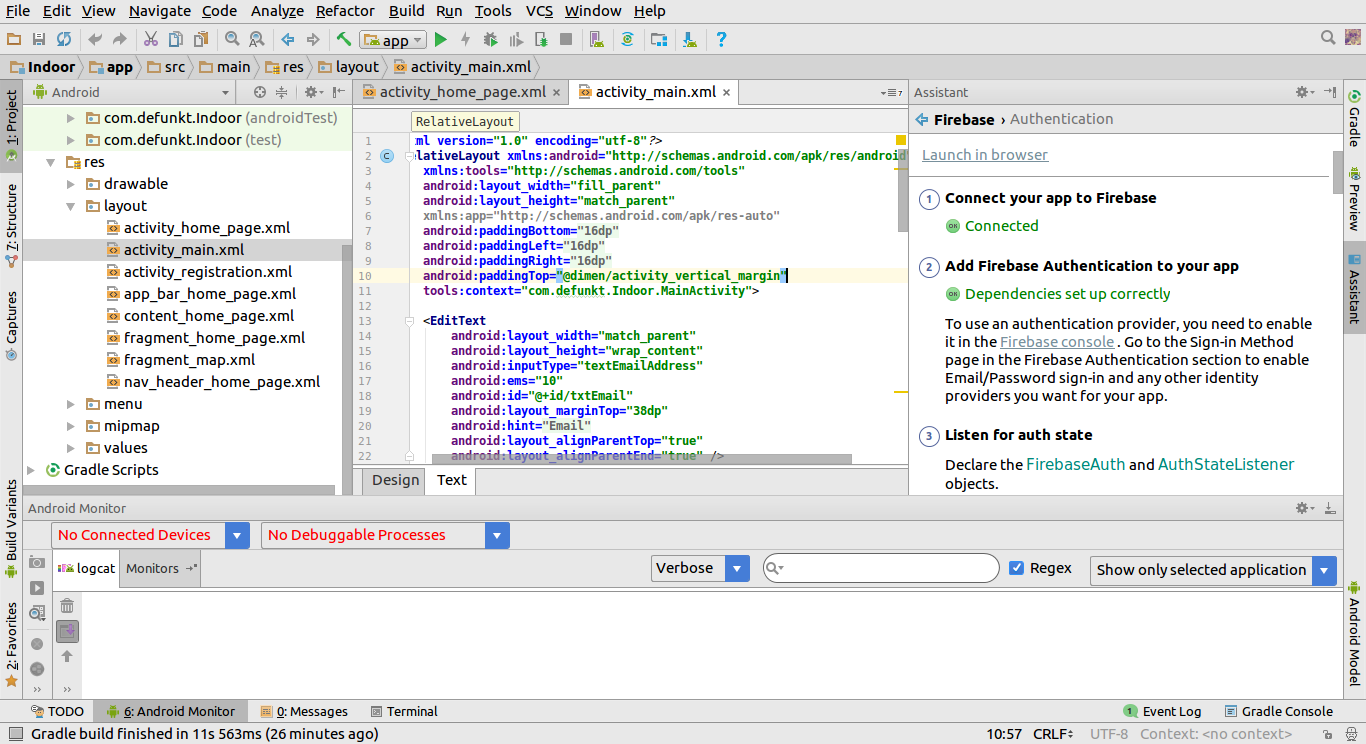


The wizard will open up in the side panel to the right. I should also note that you need to sign into your Oakland email. I invited everyone to the database that we’ll be using for the Firebase login system. Click on the corner there where the arrow points and add your account.



This menu will open up when we click on firebase. We want to choose the Authentication option. Click on it and it’ll drop down, we will then click on email and password authentication. Do steps 1 and 2. When you’re logged into your oakland email it will ask what project you want to add. It will bring up a window, Click the project that is called Chipshots and it will add that firebase database to your app.

Next we need to set up all the code. I have all the code and XML files as well as all the drawables and values files that we will need on my github branch. Just download it as a zip. Copy and paste everything for the java classes from below your package name, if it’s the same package name just copy all of it and paste it in. Do this for all Java classes.   
  
You’ll notice that there is a folder called res. In res will be all our resources that we will need for our application. In the subfolder of res is layout. Within layout are all our XML files.



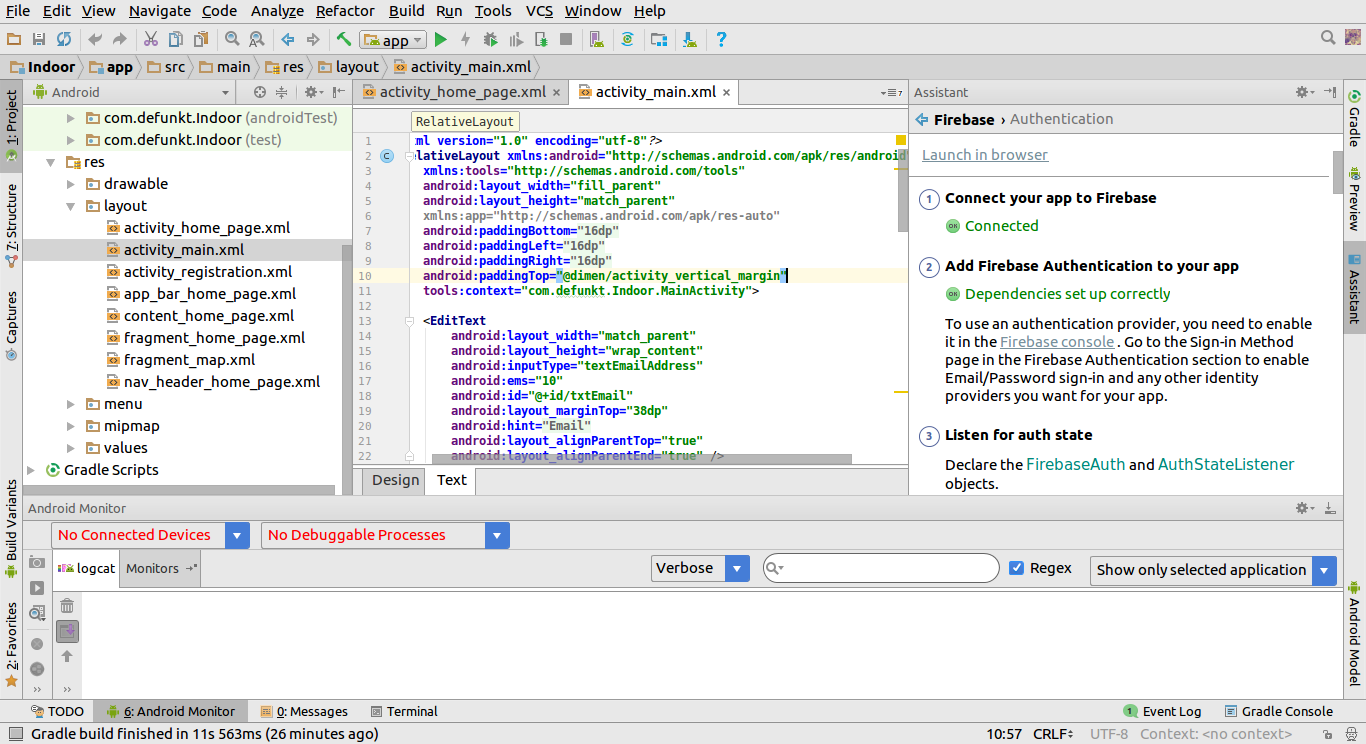
Here’s what the XML for the main activity looks like. Copy and paste everything from below the tools:context and paste it in. Do this for every XML file.

There is a drawable subfolder in the res folder. Here all of our pictures for the application will be stored. Drag and drop all of the pictures provided there.

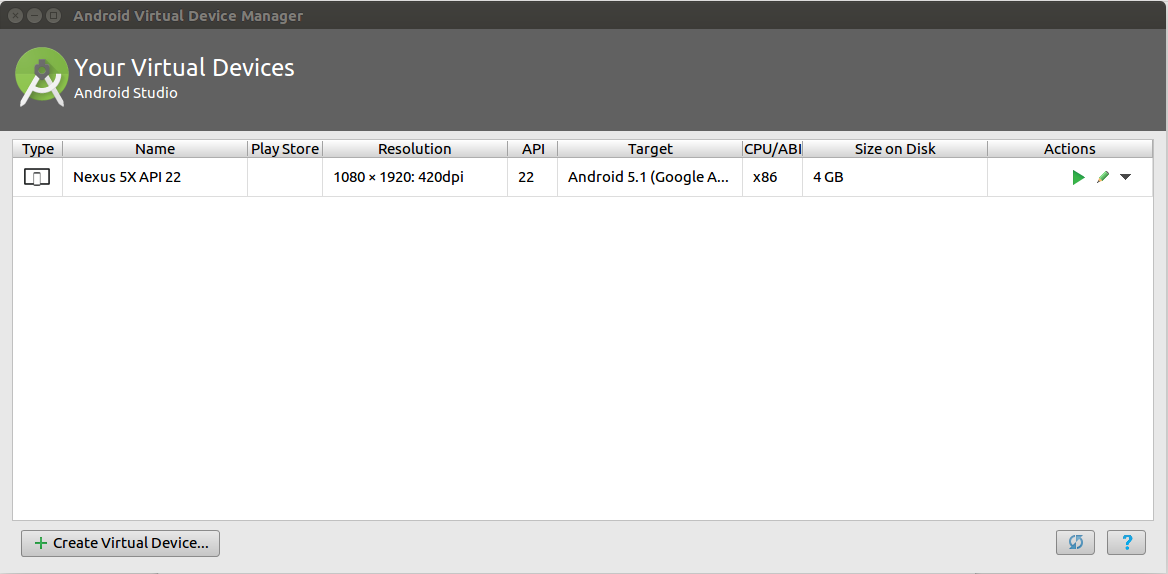
In the values subfolder of res is everything such as strings and styles values. You should be able to just drag and drop these files in since they don’t directly tie into any one project.

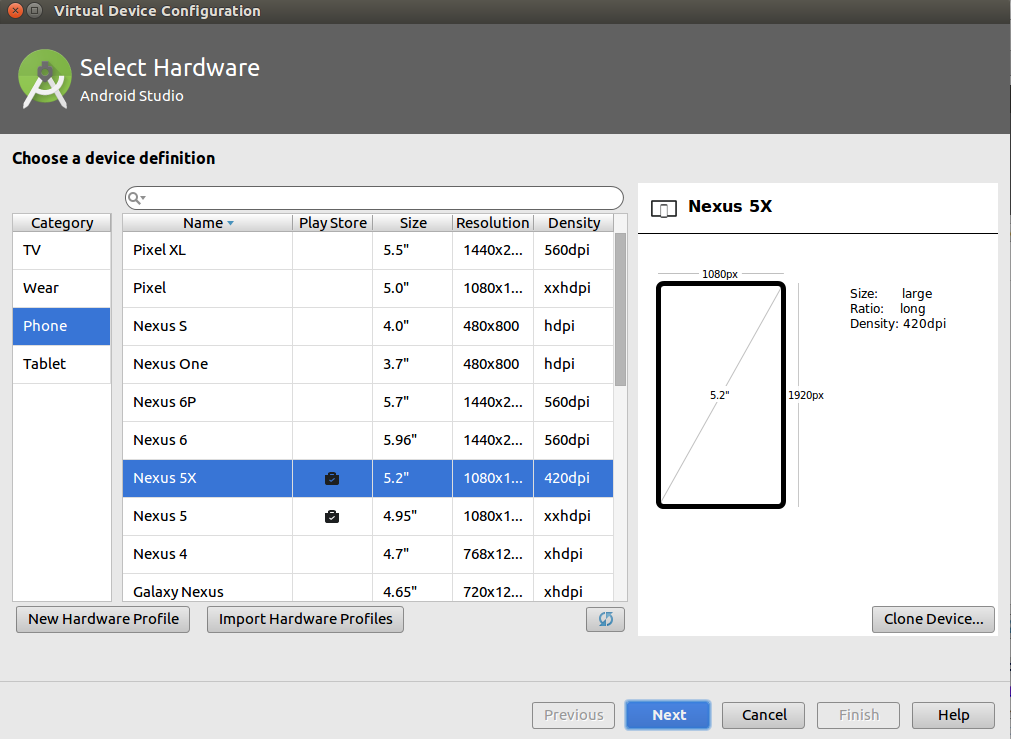
**Running the app**

You can run the app from a phone or a virtual phone which you will set up. If you want to run it on a virtual phone the set up is simple. If you choose to use your physical phone skip to page 13.

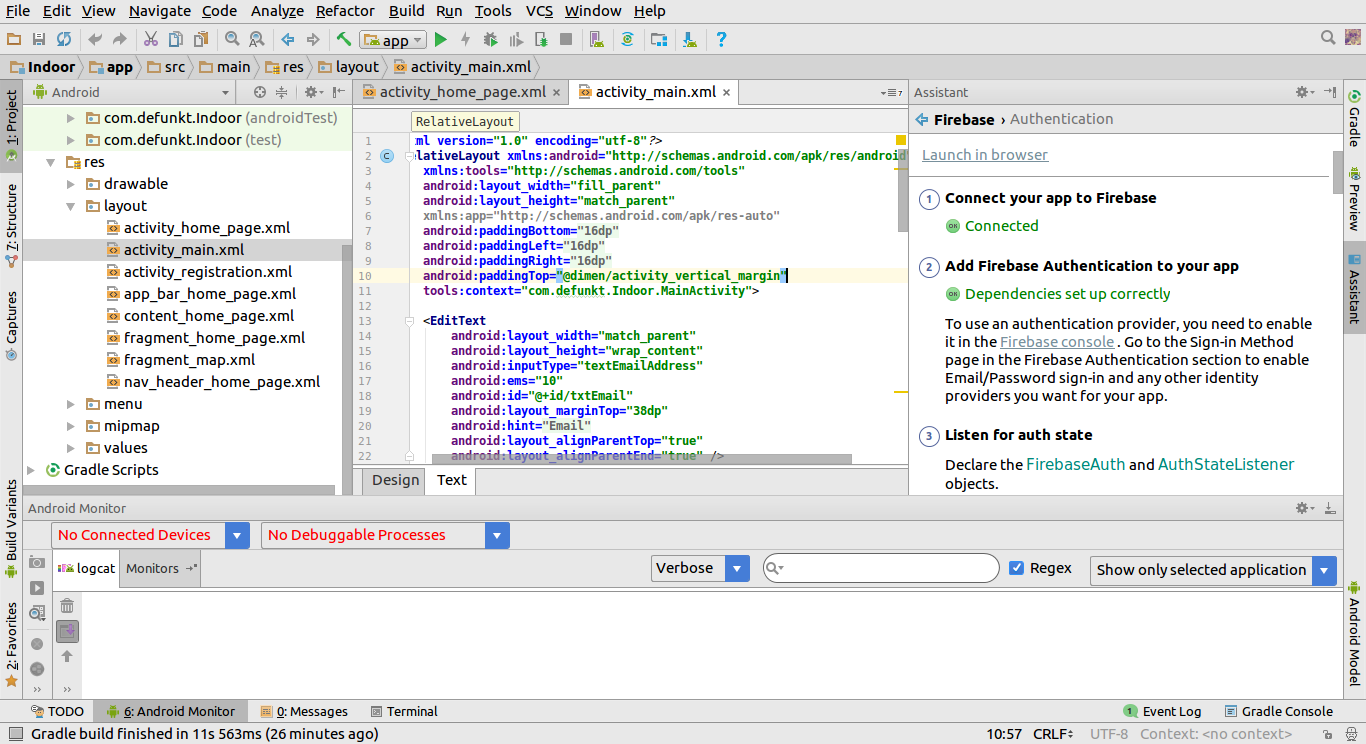


To set up your virtual phone click on the little android head that is in front of that gray rectangle. It should give the tooltip: AVD Manager. I’ve highlighted it in a red square.

This menu will pop up. Click on, “Create Virtual Device...”



You’ll see this window next. I’ve created a Nexus 5X phone. Select your phone and click next. It may take a minute to set up, but once it’s set up we’re ready to run the application.





To Run the application Click the play symbol up at the top highlighted with the red square. It will come up with this menu: Select the device you made and click OK. You device will open, but because it is the first time it has ever opened expect a delay for setup. Also this is where your android phone will show up if you choose to use a physical phone.

