# CSE2MAD Lab 3 – multiple activities

AIMS

* To explore further features of the android platform and Android studio.
* To build a multi‐activity application.
* We understand you know Java, but you need to understand how Android projects are composed and the paradigms of developing on this platform.

## BUILD a messaging application

Last week we developed an app by laying out widgets, using a layout control, responding to events. Let’s go one step further!

**Note:** This is not a cut‐and‐paste exercise (hopefully you left that behind in first year), in fact it will not work as variable names will need to be synced and customised to your solution. The code however is instructive as to the steps you need to do to satisfy the goals of the app

Today’s exercise is to build a simple text messaging app. The app will allow you to select a contact from your contact list, and send them an SMS

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Figure

The launching activity (or the main activity) will have a button that opens the contact picker.

When the user selects a contact, the user will be taken back to the main activity & the selected phone number will be displayed & the ‘send message’ button will be enabled.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

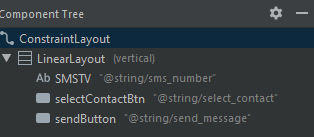
Description automatically generated

Figure 2 Figure 3

When the user clicks on the ‘send message’ button, a new activity is started. This second UI has an editable text input to type the message, and a button to send the SMS.

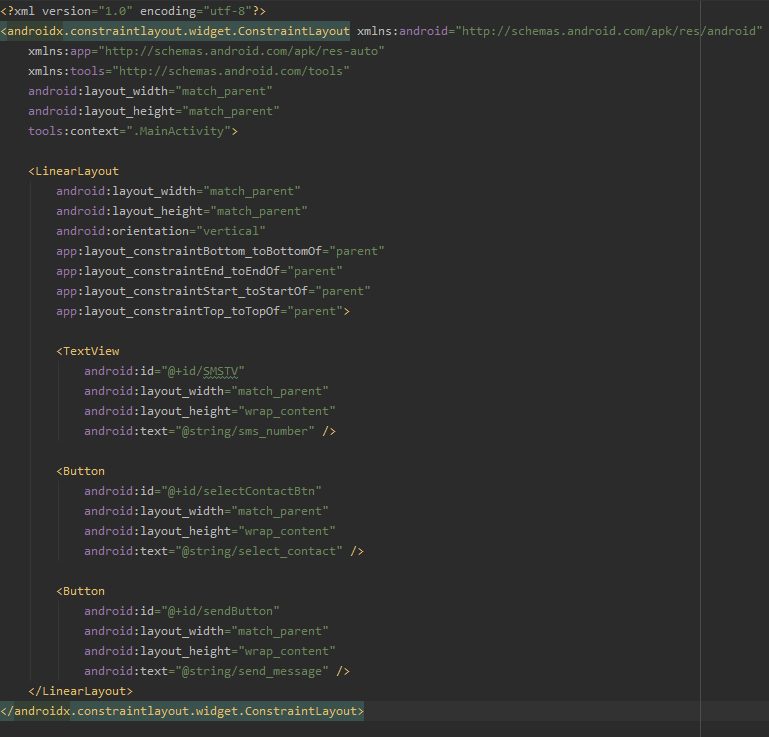
Step 1 – starting the project

1. Create a new project & choose the empty activity template. (See week 1 lab)
2. Create a new Android Project using API 25: Android 7.1.1 (Nougat)
3. Create the UI in the visual DESIGN view for the main activity as in Figure 2.



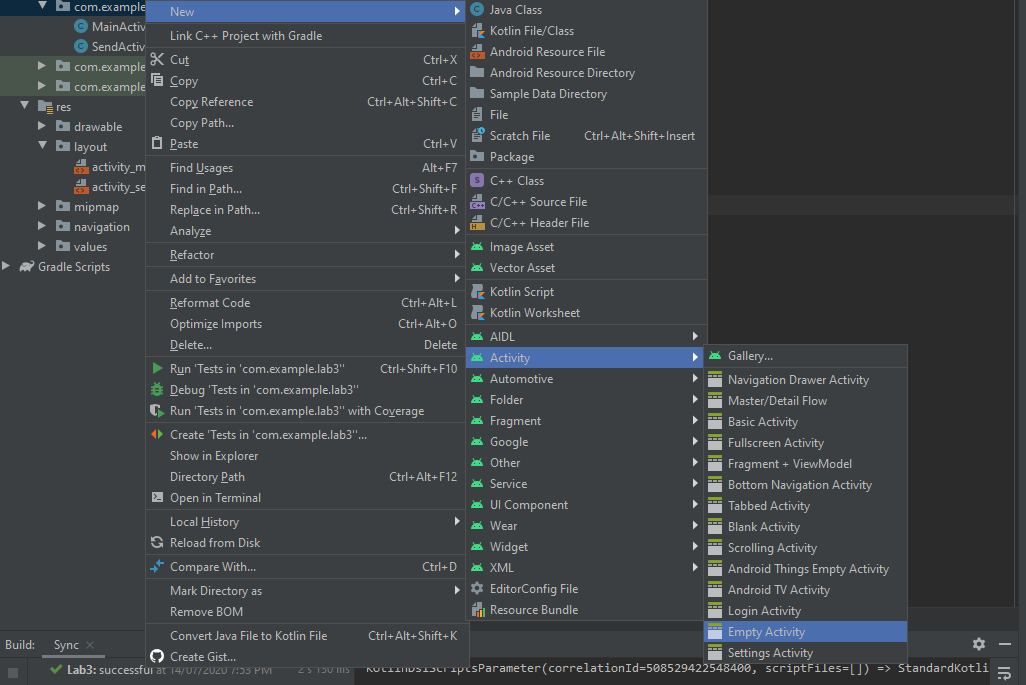
Figure

If you are stuck, here is my XML for the main activity layout, but remember it is actually easier to drag and drop from your palette into your component tree as per Figure 4!



Step 2 – Adding the Sendactivity

1. Create a new activity as per below and name it SendActivity.

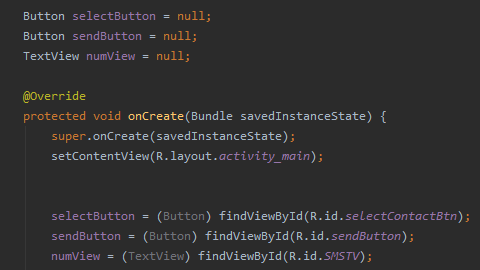


1. Create the layout as seen in Figure 3. If you are stuck here is the XML.

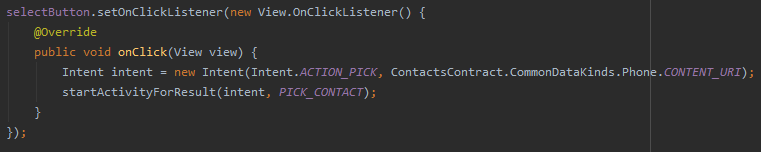


STEP 3 – code the main activity

1. Navigate in the IDE to the MainActivity.java file and add variables to access the UI elements in the activity. *Remember from last week where to declare and initialise them!* If you need a hand refer below.



1. Add a listener for the button, here we are going to create an intent.



**Note:** you will need to add the member variable PICK\_CONTACT at the  
top of class, with the other member variables.



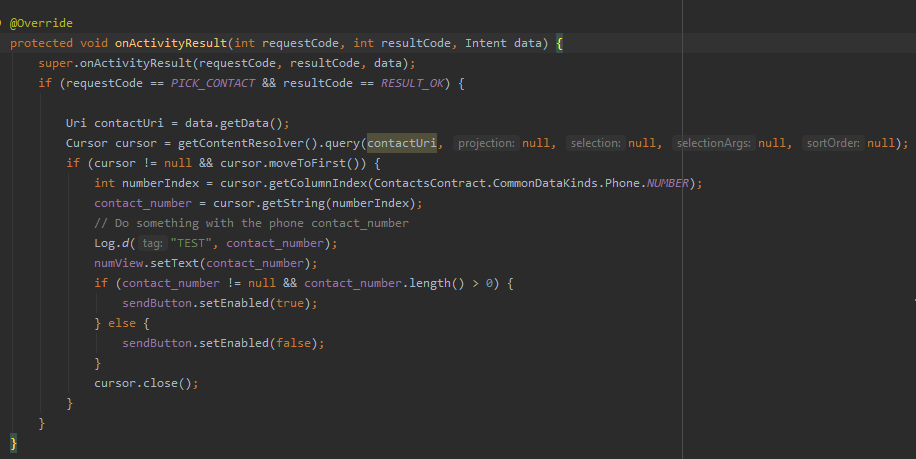
Run the app. What happens when you click on the ‘Select Contact’ button? What happens when you click on the ‘Send Message’ button? It doesn’t make sense to have the send button enabled before selecting a contact, so let’s disable it in the onCreate method.



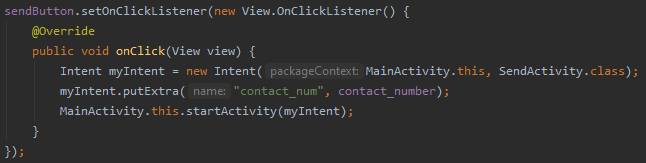
1. Now we need to get the selected contact’s phone number data into our app. First declare a member variable to store the phone number.



1. Now let’s add some code to get a contact, add this to the activity.

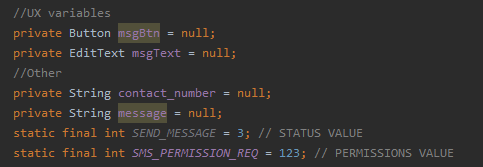


1. Guess what is next, we need to start the second activity, let’s do this via a listener to the send button in the main activity.

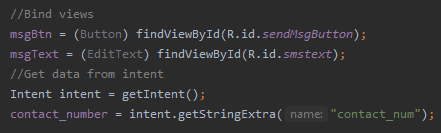


STEP 4 – code the SEND Activity

1. Let’s complete the second activity functionality. Open the SendActivity.java file. As before you are now experts in obtaining a reference to the UI widgets, for help see below.



1. As before, initialise the members and this time the intent.

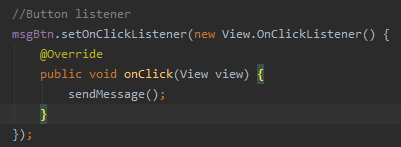


1. We are performing a few operations that need permissions (i.e. sending and receiving SMS). **Go to your AndroidManifest.xml** and notice there are two activities. Within the manifest we are going to have to ask for permission to send SMS. Add

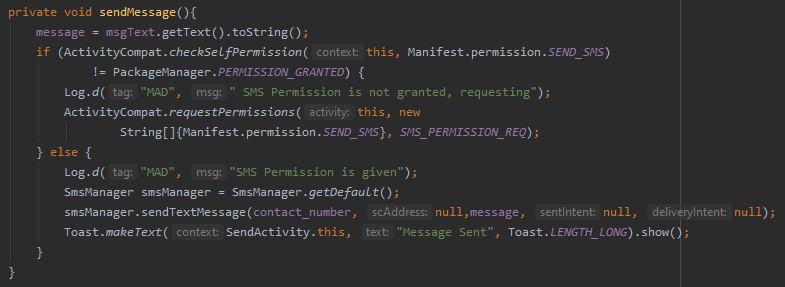
****

**above the application** specified in XML.

1. **Return to the SendActivity.java** file and handle the event on the send message button of the second activity. (Hopefully you are getting an idea where to register listeners to UI widgets. **It’s in the onCreate method**.



1. Implement the sendMessage() method as called above.



There is an OS specific permission request mechanism which we’ll talk about in the lab class for Android M (>= 6.0) and later. It has to do with permission granting at runtime or you can view chapter 74 (page 603) of the [CSE2MAD textbook online](https://search.lib.latrobe.edu.au/permalink/f/ns71rj/Almalu51373419970002146).

## Run and start texting your friends ......