CSE3MAD LAB 2 – EVENT DRIVEN DEVELOPMENT IN ANDROID

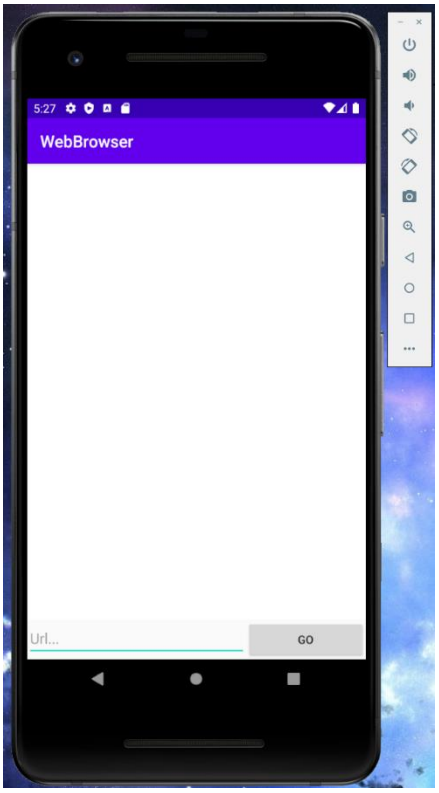
In the lab today you will develop a simple application that replicates the functionality of a web browser.

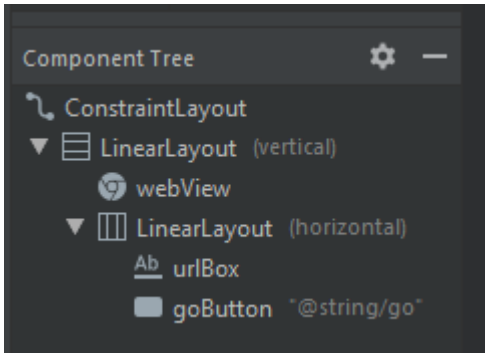
LEARNING OUTCOMES

- To create a basic UI using layouts
- How to access interactive widgets
- How to register listeners for events from those widgets

STEP 1 – CREATING THE UI

a) Create a new project & choose the empty activity template. (See week 1 lab)
b) Create a new Android Project called WebBrowser using API 27: Android Oreo
c) Create the user interface like this.





Use the Component Tree above as a start point.

I would also check out the Layout weight attribute

Useful resources

https://developer.android.com/guide/topics/ui/layout/linear

https://developer.android.com/training/keyboard-input

d) The first thing we cannot forget is we need permission for our app to access the internet!
https://developer.android.com/guide/topics/permissions/overview
Q. Where do we ask for permission?
A. The AndroidManifest.xml of course!
Insert this into the manifest **above** the 'application' block.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

e) Go to MainActivity.java

```
Button button;
WebView webView;
TextView textView;
```

Let's make some class variables for use later.
Don't forget to import the matching java libraries.

f) Edit the onCreate method to initialise the class variables.

```
button = findViewById(R.id.goButton);
webView = findViewById(R.id.webView);
textView = findViewById(R.id.urlBox);
```

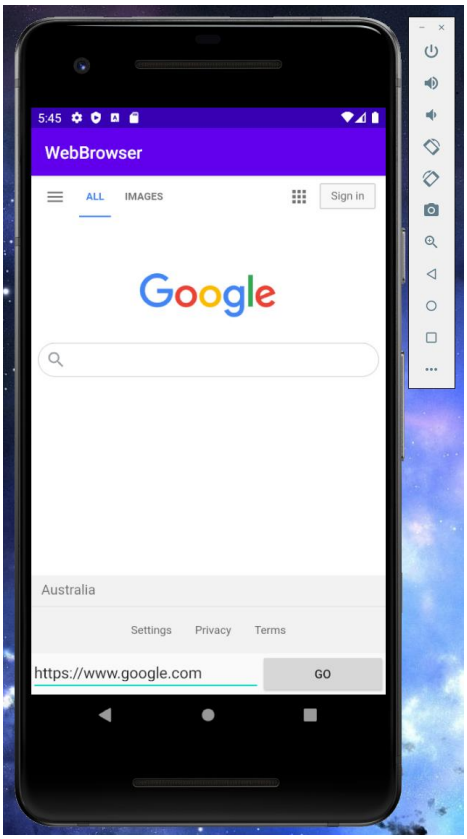
These are the id's you created in the xml file for each component, this links the xml items to the class variables.

g) Now let's register a listener. This is a key feature in the Java UI model.
https://developer.android.com/guide/topics/ui/ui-events

```
//listener for button click
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl(textView.getText().toString());
    }
});
```

When the button fires an onClick event, we have code to respond ("listen for") to it.
In this case we set up the webview with the default client, and load the URL in the textfield.

h) Now you will be able to use your brand new browser!



Notes:

- https:// is required in the URL string