

CMSC 438 Lab 8

Playing with the AndroidManifest.xml File

Overview

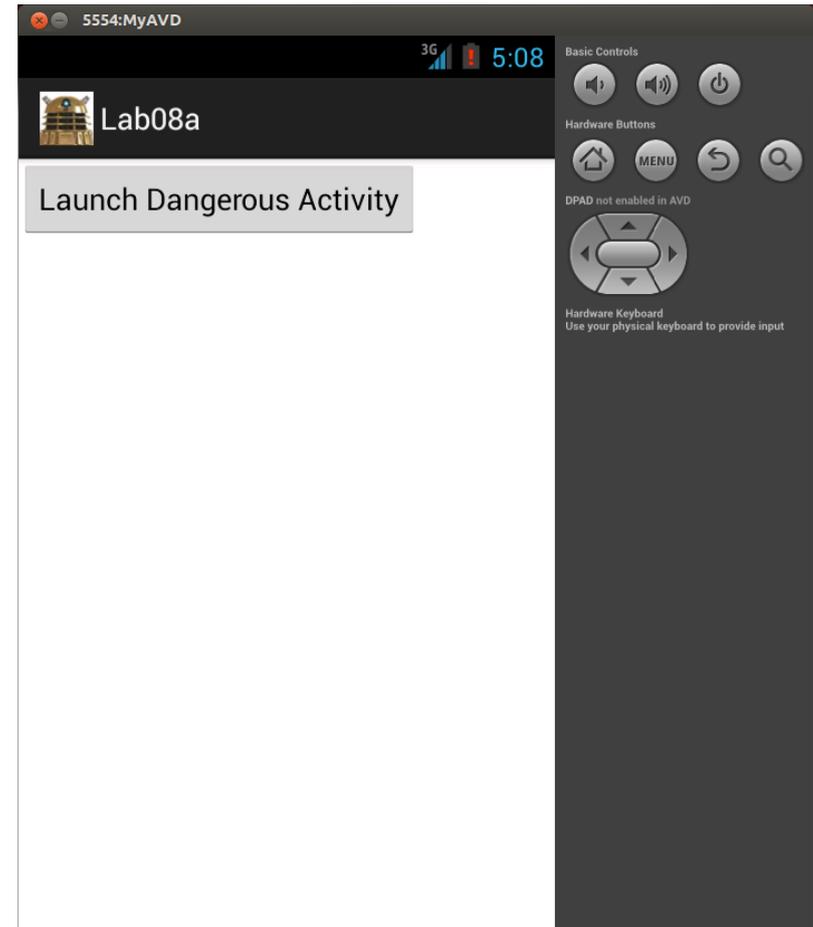
- For this lab you will practice using various features of the AndroidManifest.xml file, including
 - Custom and default permissions
 - Intent filters
 - Icons
- The information needed to do this lab can be found on the Android developer site at
 - <http://developer.android.com/guide/topics/security/permissions.html>
 - <http://developer.android.com/guide/topics/manifest/manifest-intro.html>

Overview

- Unlike previous labs in which everything was done in a single app, in this lab you will create two apps, Lab08a and Lab08b
- To submit the lab, put both Eclipse project folders into a single zip file, named using the usual format lab08-<directory_id>.zip

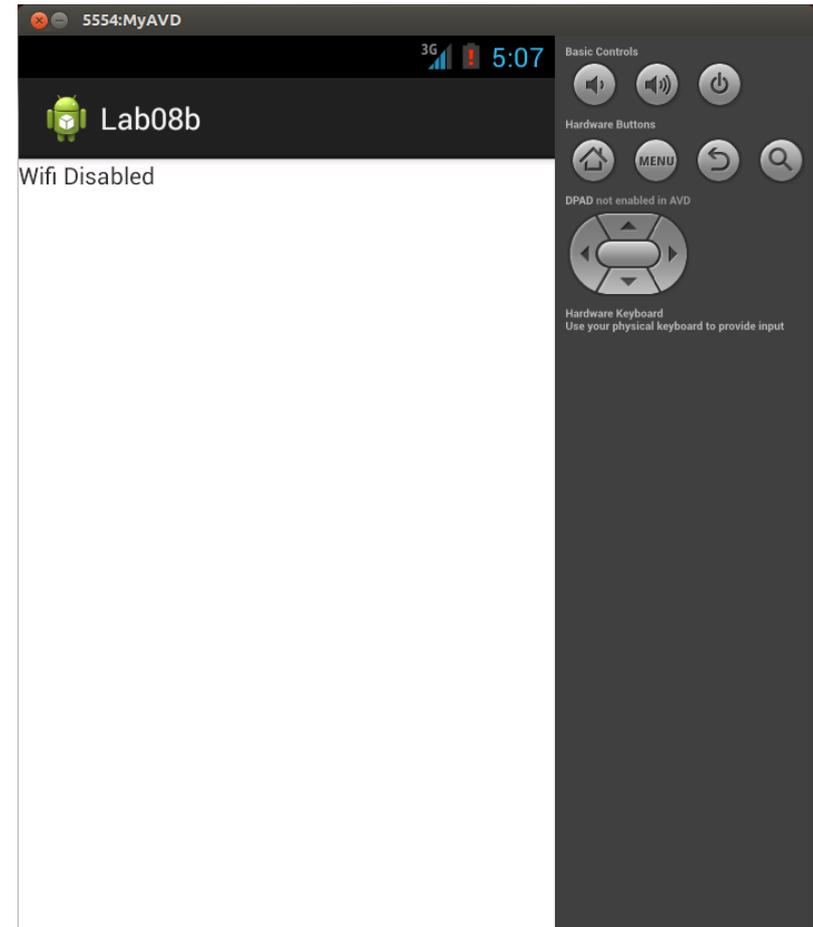
Overview

- The main activity of the first app should contain a single “Launch Dangerous Activity” button
- Clicking this button should launch the main activity of the second app
- Note the custom icon in the top left



Overview

- The main activity of the second app should contain a `TextView` that displays whether Wifi is enabled or disabled



Check Wifi State

- Your first task should be to implement the Lab08b activity that checks wifi state
- You can get the Wifi state by calling `isWifiEnabled()` on the `WifiManager`

```
WifiManager wifiManager =  
(WifiManager) getSystemService(Context.WIFI_SERVICE);
```

- Note that if you are using the emulator you will probably not be able to test the case where Wifi is enabled

Check Wifi State

- If you try to check the Wifi state you will notice that your app throws a `SecurityException`- this is because checking the Wifi state requires a specific permission
- Add a `<uses-permission>` tag with name `android.permission.ACCESS_WIFI_STATE` to the app's manifest file and try running the app again

Launching Another App

- Now implement the Lab08a activity that will launch the main activity in Lab08b
- Clicking the button should launch the activity using an implicit intent- you already used implicit intents in Lab06, but this time the intent will use a custom action string
- In the onClick callback create an Intent with the action string
“com.example.DANGEROUS_ACTION” and start an activity with it

Launching Another App

- To make the activity in Lab08b start in response to this intent, you will need to add an `<intent-filter>` tag to the `<activity>` in the manifest file
- In the `<intent-filter>`, set the name of the `<action>` tag to the same name you used for the Intent and set the name of the `<category>` tag to “`android.intent.category.DEFAULT`”

Using a Custom Permission

- Now you will create a custom permission to protect the dangerous activity that checks the Wifi state
- In Lab08b set the permission attribute in the `<activity>` tag to
“com.example.DANGEROUS_ACTION”
- Now try launching the activity from Lab08a again- the launch should fail since Lab08a doesn't have the required permission

Using a Custom Permission

- To give Lab08a permission to launch the activity you will need to add a `<uses-permission>` tag to its manifest
- Since the permission is custom, you will also need to define it with a `<permission>` tag- set the name attribute to the name you used before, set the label attribute to an appropriate string, and set the protectionLevel attribute to “dangerous”

Custom Icon

- Finally you will add a custom icon to Lab08a
- Android stores icons of various resolutions in `res/drawable-???`, but if you just want to use a single icon of a single resolution you can just put it in `res/drawable`
- Find an icon with Google image search or create one of your own, and put it in this directory
- Modify the icon attribute of the `<application>` tag in the manifest file to use your new icon