CST2335 – Toolbars & Dialogs

Lab 8

Introduction:

The goal of this lab is become familiar with Toolbars and Dialogs in Android. Toolbars are areas that appear at the top of the screen that are similar to menus. You will learn how to customize the messages and appearance of both the toolbar items, and dialog boxes.

References:

Chapter 16 in Android Application Development in 24 Hours

1. <https://developer.android.com/training/appbar/index.html>
2. <https://developer.android.com/guide/topics/ui/dialogs.html>
3. <http://www.vogella.com/tutorials/AndroidActionBar/article.html>

Steps:

1. Create a branch of your software from Lab 7, and call it “Lab 8”.
2. Click on the File menu, and select new -> Activity -> Basic Activity. For the Activity name, type “TestToolbar”. Specify the lab1 package as the package name and click “Finish”.
3. Look at the newly generated activity\_test\_toolbar.xml layout. It has an AppBarLayout, a Toolbar, an include layout tag, and a floating action button. Delete the include tag, and the file “content\_test\_toolbar.xml”.

1. In your project view, right-click on the “res” folder and select “New” -> “Android Resource File”. In the resource type, select “Menu”. In File name, type “toolbar\_menu” and click Ok. Add 3 menu items using the following template:

<item android:id="@+id/action\_one"

android:title="One"

android:icon="@drawable/add"

android:orderInCategory="101"

app:showAsAction="always" />

The id should be unique for each menu item. Put unique titles for each menu item as well (Choice 1, Choice 2, etc). For the icon, download some icon PNG graphics from: <https://icons8.com/android-icons/>, and copy them into the “res/drawable” folder in Android Studio. In the XML, you don’t need to add the “.png” file ending.

The orderInCategory is a number for sorting the icons from smallest to largest. If you want your icon to be on the left side, give it the lowest number. Repeat this for at least 3 toolbar items.

1. In the TestToolbar class, write the *public boolean onCreateOptionsMenu (Menu m)* function. The purpose of this function is to create your toolbar by inflating it from your xml file:

*getMenuInflater().inflate(R.menu.toolbarmenu, m );*

*return true;*

1. The last part is to respond to one of the items being selected. Write the method *public boolean onOptionsItemSelected(MenuItem mi)*.

The MenuItem object passed in is the object that was selected by the user. The object mi has an int id which you can get by calling *mi.getItemId();*

In the function, write a switch( id ) statement to test which of the ids from your XML file ToolbarMenu match the id that was selected, and you can then run code:

*case R.id.option1:*

*Log.d(“Toolbar”, “Option 1 selected”);*

*break;*

*case R.id.option2:*

*//Start an activity…*

*break;*

1. Add a button “Test Toolbar” to the StartActivity class and layout so that when the user taps it, it starts the TestToolbar activity. Run your lab and click the button to test that the TestToolbar activity starts and you see your toolbar with the 3 icons. Notice that there will be a button on the right side with three dots. It is the settings icon. Add an “About” menu item to the settings menu by adding another <Item> tag to the XML menu. Do not specify an icon, and set the showAsAction=“never”. This gives more menu options, but only in the settings menu. You still have to add a case statement for the item ID in the onOptionItemSelected function, and make it show a Toast saying “Version 1.0, by **Your name**”, with your actual name in the string.
2. Look at the onCreate() function in the TestToolbar class. There is a floating action button object that is created and it is given an onClickListener() callback function. By default, this starts a Snackbar notification. Change the string message to something that you have written. Click on the Letter icon at the bottom right of the screen and you will see the Snackbar message appear from the bottom of the screen. It is similar to a toast, but appears on the edge of your screen instead of the centre. Set the case statements for your first menu item to display a Snackbar saying “You selected item 1”.
3. Look at Resource #2 listed at the top of the lab, showing how to use Dialogs. The basic pattern is:

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

builder.setTitle(R.string.pick\_color)

// Add the buttons

builder.setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// User clicked OK button

}

});

builder.setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// User cancelled the dialog

}

});

// Create the AlertDialog

AlertDialog dialog = builder.create();

dialog.show();

However, you must add your own String resources for the Title, the Positive Button text, and the Negative button text. The title should be “Do you want to go back?” and if the user selects the positive button, then finish the current activity. If they select the negative button then do nothing. Set this to be the code for selecting on the second toolbar item.

1. Looking at the Resource #2, find the section talking about creating a custom layout. That shows how you can make your own custom dialog box. Use the custom layout given in the example, but download another free icon from the website mentioned in step 4. Save this image in the “res/drawable” folder and use this image as the source of the imageview. Also, remove one of the EditText widgets from the example. Instead of Username and Password, just have an EditText for “New Message”. If the user selects Ok, then the string entered will be the new message that is displayed by the Snackbar object when selecting menu item 1. This step (step 10) should be what happens when the user selects your menu item 3.
2. Demonstrate your work to the lab professor showing the following parts of your lab work:  **(6 marks total)**
3. Launch the application and navigate to the TestToolbar. Show that your Toolbar appears with 3 icons, and a fourth About item in the settings menu. **+1**
4. Select Item 1 in the toolbar, and show that it makes a Snackbar notification appear at the bottom of the screen. **+1**
5. Select Item 2 in the toolbar, and select the negative button “No”, or “cancel”. This should not end the current activity and navigate back to the StartActivity. **+1**
6. Select Item 3 in the toolbar, and set a new message for the Snackbar notification. **+1**
7. Select Item 1 in the toolbar and show the Snackbar has a new notification message. **+1**
8. Select Item 2 in the toolbar, but now select the positive button so that it ends the current activity. **+1**
9. Commit your work with Git, using the commit message “Finished lab 8”, and push the code to the repository (either local directory on your computer or Github account).

Note: This lab uses the following patterns:

* 1. Builder pattern for the Dialog building.