CST2335 – Graphical Interface Programming

Lab 7

Introduction:

The goal of this lab is become familiar with Fragments. We will modify a layout so that it uses an activity with fragments instead.

References:

1. <https://developer.android.com/training/basics/fragments/index.html>

Steps:

1. From the command line, create a branch of your software from Lab 6. If you have a windows computer, open the “Git Bash” program. If you have an Apple or Linux computer, just open a terminal. On the command line, navigate to your AndroidStudioProjects folder, and then to your Labs folder. Once there, type “git branch” to list the branches on your computer. Now type “git branch Lab7” to create a new branch. Then type “git checkout Lab7” to move to that branch. Now type “git branch” to list all the branches, and there should be a star (\*) next to Lab7 to show it is your current branch.
2. In the Android project view on the left side of Android Studio, right-click on your layout file for your chat activity from Lab 4 and select “copy”. Make sure that AndroidStudio is using the “Project Files” view of your project so that you can see if there is a “layout-sw720dp” folder under the “res” folder. If there is not one, then create the folder. Paste your copy of the layout of the chat activity into that folder. In the layout that you pasted, add a “FrameLayout” to the RelativeLayout so that it is to the right of the existing ListView. You should change the ListView width to be 300dp instead of wrap\_content. Align the right side of the Send button to the right side of the ListView, using the layout\_alignRight parameter. Then add the FrameLayout to the right of the ListView and SendButton:

A screenshot of a computer

Description automatically generated

Figure 1 The tablet layout with a FrameLayout to the right of the chat messages list

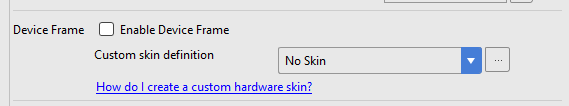
1. In the onCreate() function of your ChatRoom activity, use findViewById() to look for the id of the FrameLayout. If it returns null then you are on a phone, otherwise it’s on a tablet. Store this in result in a Boolean variable.
2. Right-click on the “App” folder in your project window. On the menu, select “New” -> “Fragment” -> “Fragment (Blank)”. On the next window, give the name “DetailsFragment”, and unckeck the option for “Include fragment factory methods”. This will generate a java class that extends Fragment, and an xml file that is the layout. The onCreateView function will already be generated that just inflates a layout and returns it.
3. Modify the generated fragment layout so that it looks like this:

A screenshot of a cell phone

Description automatically generated

Figure 3 The layout for your fragment..

1. In your ChatRoomActivity, you added an onItemLongClick function that shows a dialog box with the details of your chat message. In this lab, ***call setOnItemClick( (list, view, position, id) -> { } );*** to the list view. It should look like line 35 of the FragmentExample.java file that was shown in class.
2. In the callback braces { } , if you’re running on a tablet, create a fragment transaction that calls replace( ) so that it loads a Fragment object that you created in step 4 of this lab, and load it into the FrameLayout id that you created in step 2. If you’re running on a phone, call startActivity() to go to a new Activity that will then load your fragment.
3. In AndroidStudio, create a new Activity called EmptyActivity by using the menus “File” -> “New” -> “Activity” -> “Empty Activity”. It should generate the java and xml files for your new activity. Open the newly created layout file and select all of the xml code and delete it. Copy and paste the <FrameLayout> tags that you created from step 2 and paste it into the file. It is important that FrameLayout is now the root tag. AndroidStudio will complain that there is no namespace declared for android: but you can type alt+enter to import the xmlns= declaration. Lastly, look at the startActivity() call from step 7 and change the Intent object so that it will transition to EmptyActivity.class.
4. In the onCreate() function of EmptyActivity.java that you created in step 8, copy and paste the FragmentTransaction code that you wrote in step 7. Now test your lab on a Tablet emulator and your fragment should be loading in the FrameLayout that is on the right side of the screen. Test your code also on a Phone emulator and when you select a message, you should be running the startActivity() code that starts EmptyActivity, and then loads the Fragment as well. The message, id and isSend parameters won’t be set yet but the fragment should at least be loading.
5. To load the data, look at your DetailsFragment class, at the onCreateView function. Change the function so that you first store the View that is inflated in a variable. Just like you did in Lab4 when you were inflating the row layouts, you have to call findViewById() to get the two TextViews and CheckBox. You can then call setText, and setChecked on those objects. The remaining task is to send the data from the setOnItemClick handler to the Fragment onCreateView function. Use a Bundle object to load the message text, item id, and Boolean isSend into a table. Then call setArguments() on your Fragment object to pass the data. Lastly, call getArguments() to retrieve the Bundle in your onCreateView() function. Look at the FragmentExample, and DetailFragment classes presented in class on how to do it correctly.
6. Lastly, in your DetailsFragment class, when you hit the “Hide” button, it should create a fragment transaction that removes the fragment. Look at lines 40 – 45 of DetailFragment.java that was presented in class on how to write a remove() transaction.
7. Your tablet and phone should now load the DetailsFragment which then shows the details of the selected item from the list view. Remember that doing a short click on the listview should show the details of the selected message. Doing a long click of the listView should still show the AlertDialog asking if you want to delete the message. You should modify the Positive button handler so that it not only deletes the item from the list, but then also removes the fragment from the FrameLayout if you are running on a tablet.
8. There is a strange bug with the emulators where they don’t use the layout-w900dp if the device frame is enabled, and if there is a custom skin definition. You might have to create a new emulator with “No Skin”. Open the virtual device manager and select “Create Virtual Device”. Select Category: Tablet and then 10.1” WXGA Tablet. Next select Lollipop or newer Android version. On the next screen, click Show Advanced Settings and make sure “No Skin” is selected and then Finish.



Demonstrate your work to the lab professor showing the following parts of your lab work: **5 marks**

* 1. Launch the application and navigate to the Chat Activity. Show that your messages saved in the database still display like in the previous lab. **+1**
  2. Launch your application on a Tablet with at least 720 pixels width. When you select a message, you should see it text on the right of the ListView. **+1**
  3. Deleting an item from the listView should remove the fragment from the FrameLayout. **+1**
  4. Launch your activity on a phone emulator. It should launch a new activity that loads the DetailsFragment to take up the whole screen. **+1**
  5. Clicking on the “Hide” button removes the fragment from the FrameLayout **+1**

1. From the command line, commit your work with Git. From git bash, or your Apple/Linux terminal, type **git commit –m “Finished lab 7”**. This should commit your work to git. Next type **git push** to upload your work to your Github account.