CST2335 – Graphical Interface Programming

Lab 4

Introduction:

The goal of this lab is become familiar with the ListView widget, and creating a custom data adapter object to populate the ListView with data.

References:

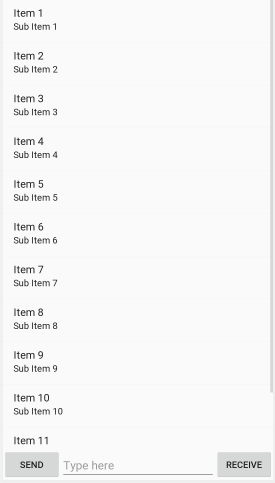
1. <http://www.vogella.com/tutorials/AndroidListView/article.html>
2. <https://www.javacodegeeks.com/2013/09/android-listview-with-adapter-example.html>
3. <https://www.tutlane.com/tutorial/android/android-alertdialog-with-examples>

Steps:

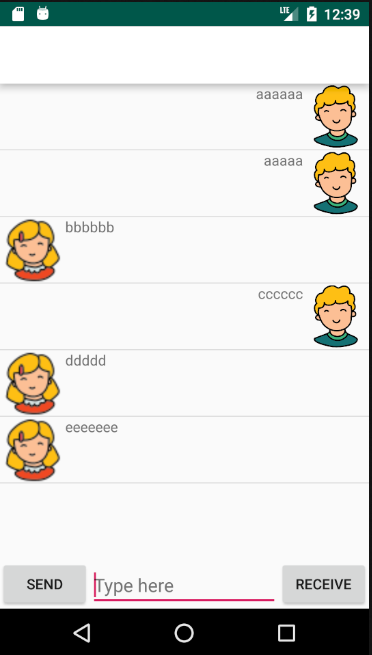
1. Make sure that you Create a branch of your software from Lab 3. Select the VCS menu in Android Studio, and click “Git” -> “branches”. From the dialog box, select “New Branch” and call it “Lab 4”.
2. Add a button to the to your ProfileActivity from Lab 3. Set the text of the button to be: “Go to chat”. Set an onClick listener of the button to go to a new activity, called “ChatRoomActivity”. You can create the activity in AndroidStudio by selecting “File” -> “New” -> “Activity” -> “Empty Activity”, and give it the name ChatRoomActivity.
3. You will need two drawable resources for the image view. Go to the website:

<https://icons8.com/icon/set/avatar/all> and find two png files that you want to have display in your list. Try and get the small version of the images, and rename them row\_receive.png, and row\_send.png. Drag them both into your drawable folder.

1. Design a layout that looks like this:



1. There is a ListView at the top, a Send button which simulates you sending a message, and a Receive button which simulates you receiving a message. Between the buttons is an EditText with a hint “type here”. Depending on which button you press, a new message should show in the list containing what you typed in the EditText. If you pressed the Receive button, then one of the avatars you downloaded should show on the left side of the string. If you pressed the Send button, the text should be to the right side of the screen, and the avatar should show to the right of the text.



1. A hint: You should create a Message class that stores each message text, and whether it is a Send or Receive message. Then when you display a row, the layout you create will either be a layout for Send with the avatar and text on the left, or a layout for Receive with the avatar and text on the right side.
2. In AndroidStudio, type ctrl + O (the letter ‘O’, not number zero). From the list of inherited functions, implement these ones:
   1. **int getCount()** - This returns the number of rows that will be in your listView. In your case, it should be the number of strings in the array list object ( *return list.size()* ).
   2. **Object getItem(int position)** – This returns the item to show in the list at the specified position: ( *return list.get(position)* )
   3. **View getView(int position, View convertView, ViewGroup parent)** – this returns the layout that will be positioned at the specified row in the list. Do this in step 9.
   4. **long getItemId(int position)** – This is the database id of the item at position. For now, we aren’t using SQL, so just return the number: position.
3. When you click Send or Receive, the EditText where you typed should be emptied, and you have to call notifyDatasetChanged() on the ListView’s adapter object. This will cause the functions in step 7 to be called to populate the list.
4. Set an onItemLongClick() listener to the listView. Whenever you click on the list at a certain row, show an AlertDialog with the title: “Do you want to delete this?”, the message should display the following information:
   1. “The selected row is: “ followed by the index that was clicked.
   2. “The database id id:” followed by the database id that is given.

And have a positive and negative button. The positive button should delete the item at the specified row and update the list with the item removed.

1. Commit your work with Git, using the commit message “Finished lab 4”, and push the code to your Github account.
2. Demonstrate your work to the lab professor showing the following parts of your lab work:
   1. There is a button on the ProfileActivity page that goes to the ChatRoomActivity.
   2. Typing in various messages populates the chat window, like the picture above.
   3. Typing in various messages clears the EditText control.
   4. You can select items in the list and view an AlertDialog window showing the index and database id of the selected row. The user can then delete the row by selecting the Positive button.
   5. Clicking the Android Back button goes back to ProfileActivity.
   6. Update your strings.xml file to include translated messages for the Send, Receive, and Start Chat buttons. Show that changing Android’s language to your second language displays the translated messages.

Note: This lab uses the following patterns:

* 1. Adapter pattern for the ListView data adapter.
  2. Iterator pattern for letting the ListView iterate over all the data in your data adapter.
  3. Delegate (Delegation) pattern for using an inner class (ChatAdapter) to provide data for the ListView.

Marks:

* The ProfileActivity has a button to launch a chat window +1
* Typing chat messages populates the chat window (send/receive) +1
* The EditText window gets cleared after submitting a message +1
* User can exit the chat back to the Profile Activity +1
* Lab 3 functionality still works +1
* You can select a chat row to view an AlertDialog with the index and database id +1
* You can select the Positive button if the AlertDialog to delete the item +1
* The Send, Receive, and Start Chat buttons are translated into 2nd language +1