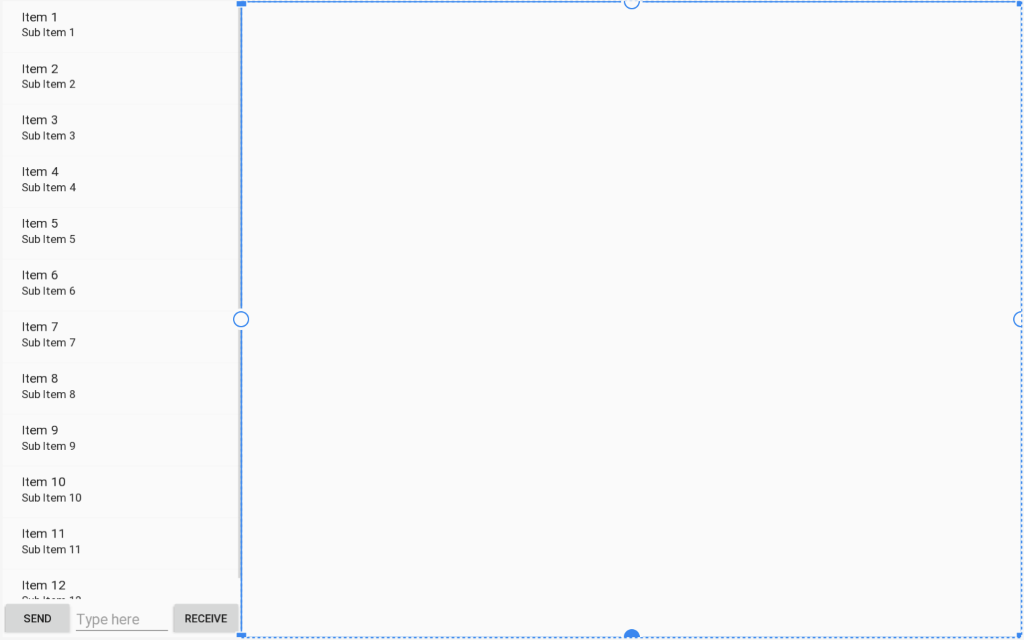
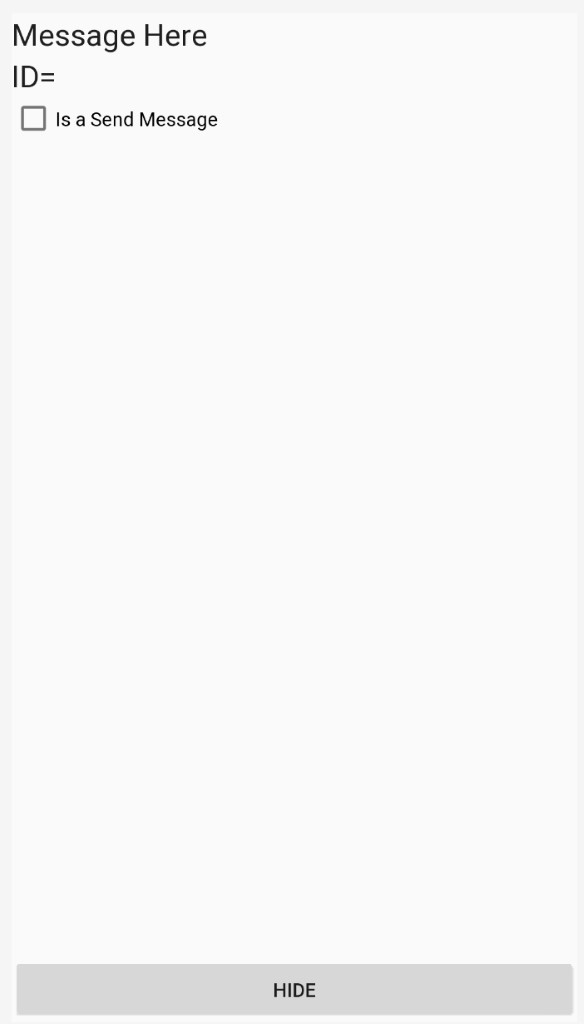
1. .  
   2. Add a “FrameLayout” to the RelativeLayout in activity\_chat\_room.xml so that it is to the right of the existing ListView. You should change the ListView width to be 300dp instead of wrap\_content. Align the right side of the Send button to the right side of the ListView, using the layout\_alignRight parameter. Then add the FrameLayout to the right of the ListView and SendButton:

****

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. Create a new fragment named “DetailsFragment”. 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:

****

1. In your ChatRoomActivity, there is an onItemLongClick function that shows a dialog box with the details of your chat message. Now, ***call setOnItemClick( (list, view, position, id) -> {   } );*** to the list view. It should look similar to:Text

   Description automatically generated

1. 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.

1. Create a new Activity named “Empty Activity”. Open the newly created layout file and select all 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.

1. 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 code 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.

1. 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. You must 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.

1. Lastly, in your DetailsFragment class, when you hit the “Hide” button, it should create a fragment transaction that removes the fragment.

1. 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.