CS478: Software Development for Mobile Platforms

Project #3

Due time: 11:59 pm on 3/19/2023

Total points: 100

Instructor: Ugo Buy

TAs: Sasanka Veleti, Anupreet Paulkar and Harshil Bhatt

Copyright © Ugo Buy, 2023. All rights reserved.

The text below cannot be copied, distributed or reposted without the copyright owner's written consent.

Design and code two new Android apps meant to work together on a phone or tablet running Android 13. The first app, A_1, helps vacation planner choose between a visit to Orlando, Florida and New York City, New York. The second app, A_2, has specific information about tourist attactions in these two cities.

1. Application A_1 defines an activity containing two read-only text views and two buttons, one each for New York City and Orlando. The buttons, when selected, will first show a short toast message indicating the selected city, then broadcast an intent. This intent will contain an extra depending on the button pressed, e.g., "New York City" or "Orlando:". This extra will indicate the city a user intends to visit. The text views describe the meaning of the buttons to the device user.

   A_1 must acquire a dangerous level permission denoted by the string *"edu.uic.cs478.spring23.mp3"* before broadcasting the intent. This permission is defined by application A_2.

2. Application A_2 defines dangerous level permission *"edu.uic.cs478.spring23.mp3"*. Application A_2 does not acquire the permission. However, application A_2 defines a broadcast receiver that receives A_1's broadcasts but responds to the broadcasts only if the sender (i.e., application A_1) owns permission *"edu.uic.cs478.spring23.mp3"*.

   Application A_2 starts off by launching a splash activity that just shows the message "Welcome to Application 2!". Upon receipt of A_1's broadcast, A_2 will launch one of two activities. The first activity (Orlando) displays information about 5 points of interest in Orlando. The second activity shows the same information about New York City. Both activities contain two fragments, whose behavior is described below. In addition, application A_2 maintains an *options menu* and an *action bar*. The action bar shows the name of the application (your choice) and the overflow area. The options menu allows a device user to switch between attractions and hotels. This menu should be clearly accessible from the overflow area.

Each of the two tourist-attraction activities in A_2 contains two fragments. The first fragment displays a list of tourist attractions for either Orlando or New York City; this list must be scrollable, as needed. The device user may select any item from either list; the currently selected item will stay highlighted until another item is selected. The second fragment shows the official web site of the highlighted item using a *WebView* widget.

When the device is in portrait mode the two fragments are displayed on different screens. First, the device will show only the first fragment. When the user selects an item, the first fragment disappears and the second fragment is shown. Pressing the "back" soft button on the device will return the device to the original configuration (first fragment only), thereby allowing the user to select a different point of interest. When the device is in landscape mode, application A_2 initially shows only the first fragment across the entire width of the screen. As soon as a user selects an item, the first fragment is "shrunk" to about 1/3 of the screen's width. This fragment will appear in the left-hand side of the screen, with the second fragment taking up the remaining 2/3 of the display on the right. Again, pressing the "back" button will return the application to its initial configuration. The action bar should be displayed at all times in all 3 activities regardless of whether the device is in portrait or landscape mode.

Finally, the state of application A_2 should be retained across device reconfigurations, e.g., when the device is switched from landscape to portrait mode and vice versa. This means that the selected list item (in the first fragment) and the page displayed in the second fragment will be kept during configuration changes.

Hints. This is a challenging project. Start off easy by coding A_1 and a mockup of A_2 containing just a broadcast receiver, the splash activity and two empty activities. Launch the appropriate activity from the broadcast receiver. Note that, since Oreo, A_2's receiver must be registered programmatically in order to respond to A_1's broadcast. This means that A_2 should be installed and running (in the stopped state), when A_1 broadcasts the intents. Make sure that it all works before moving to the next step.

Next, define appropriate layout files for one of the activities in A_2 and define the two fragments for that activity. You'll need to define the fragments programmatically. Make sure that it all works when the device is in landscape mode.

Next, handle the activity's fragments when the device is in portrait mode and include fragment retention across configuration changes.

Finally, code the second A_2 activity in a way similar to the first activity.

Implementation notes. For this project use a Pixel 5 device running the usual Android platform (API 33—V13). You are not required to provide backward compatibility with previous Android versions. Use method *setRetainInstance(true)* to prevent fragments from getting deleted when a configuration change occurs, even though the activity containing the fragments will be destroyed and recreated. Check out the app *Fragments Static Config Layout* from Adam Porter's Coursera course, available on our Blackboard site, to see how to work fragment retention in A_2.

You must work alone on this project. Submit the two Studio projects as a single zip archive using the submission link in the assignment's page on Blackboard. No late submissions will be accepted.