CS478: Software Development for Mobile Platforms

Project #3

Due time: 9:00 pm on 4/3/2020
Total points: 100
Instructor: Ugo Buy
TAs: Aishwarya Chavali and Ajith Nair

Design and code two new Android apps meant to work together on an Android phone or tablet running version 9, Pie. The first app, A_1, helps visitors in Chicago decide on points of interest in the nation's capital. The second app, A_2, has specific information about the points of interest.

1. Application A_1 defines a dangerous level permission *"edu.uic.cs478.sp2020.project3"*. In addition, A_1 defines an activity containing two read-only text views and two buttons. The buttons, when selected, will first show a short toast message, then broadcast two different intents (e.g., *attractions* and *restaurants*) depending on the button pressed. The text views describe the meaning of the buttons to the device user.

2. Application A_2 receives the intents; however, this app will respond to the intents only if the sender owns permission *"edu.uic.cs478.sp2020.project3"*. Depending on the kind of intent that was received, A_2 will launch one of two activities. The first activity (attractions) displays information about 8 points of interest in the city of Chicago, Illinois (e.g., the Lincoln Park Zoo, Navy Pier, the Museum of Science and Indutry, the Art Institute, the TILT!, etc.) The second activity shows at least 6 restaurants located within Chicago's city limits. 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 restaurants. This menu should be clearly accessible from the overflow area.

Each of the two activities in A_2 contains two fragments. The first fragment displays a list (either the attractions or the restaurants, depending on the activity). 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 browser stored on the device. This browser could be Firefox, Chrome or any other, depending on installed applications and the preferences of the device's user.

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 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 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 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. 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 2.3 device running the usual Android platform (API 28—V9, Pie). You are not required to provide backward compatibility with previous Android versions. Use method *setRetainInstance()* to prevent fragments from getting deleted when a configuration change occurs, resulting in the destruction of the containing activity. 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 zip archive using the submission link in the assignment's page on Blackboard. No late submissions will be accepted.