Fragments, Permissions & Broadcast Receivers

Due Date: Sunday November 7th 2021, by 11:59pm

Project Details:

Design and code two new Android apps meant to work together on an Android phone or tablet running Android 11, API level 30, . The first app, A1, helps visitors in Chicago decide on points of interest in the Windy City. The second app, A2, has specific information about the points of interest.

1. Application A1 defines a dangerous level permission "edu.uic.cs478.fall2021.project3". In addition, A1 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 A2 receives the intents; however, this app will respond to the intents only if the sender owns permission "edu.uic.cs478.fall2021.project3". Depending on the kind of intent that was received, A2 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 Industry, 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 A2 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 A2 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 A2 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 A2 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.

Implementation Details:

For this project use the usual Pixel 3 XL device running the usual Android platform (API 30, Android 11). You are not required to provide backward compatibility with previous Android versions. Your layout for the fragments should use FragmentContainerView

and all fragment communication should occur through the use of a ViewModel. See the sample app “Fragment with ViewModel” on BB for an example of using ViewModel objects.

Submission Details:

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.

Academic Integrity:

Unless stated otherwise, all work submitted for grading *must* be done individually. While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you *cannot* work in teams, you cannot work side-by-side, you cannot submit someone else’s work (partial or complete) as your own. The University’s policy is available here:

https://dos.uic.edu/conductforstudents.shtml.

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing

your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. It is also considered academic dishonesty if you click someone else’s iClicker with the intent of answering for that student, whether for a quiz, exam, or class participation. Academic dishonesty is unacceptable, and penalties range from a letter grade drop to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml.