ECEN489 Mobile Applications with Android MP 8 - Due Date: 04/3/2019

Firebase

Lecturer: Srinivas Shakkottai *Scribe: Self*

1 Introduction

In this machine problem, you will learn how to handle remote databases using Firebase. You will learn how to create a new user, authenticate the user, to push objects to Firebase, and to pull relevant objects using queries.

2 Problem statement

This app will have three activities. MainActivity is an account creation/login screen. PushActivity is used to push objects into Firebase, and the PullActivity is used to query Firebase.

First create a Firebase account for yourself, and upload the simpsons json file to it. Link up your app to Firebase. Follow the steps using the GUI, and afterwards modify the module build.gradle module and app build.gradle as shown in the lecture slides. Don't forget to include Internet permissions in the Manifest.

MainActivity

This activity will allow the user to either create an account, or login to an existing account. After successful account creation or login, the app immediately switches to the PullActivity

XML: The layout of MainActivity will consist of a title (TextView), two editTexts to capture an email ID and a password, and two buttons for "Create Account" and "Login". Remember to use android:inputType="textPassword" as a field in your EditText for capturing the password.

JAVA: When "Create Account" button is pressed, a new account us created on Firebase. When "Login" is pressed, the user is logged into Firebase. The app then switches to PullActivity.

PullActivity

This activity is used to query data from the database. First you must check if the user is authenticated (i.e., not null). You must support two types of queries as follows, with the outputs of the queries being shown either in a ListView or a RecyclerView (preferred).

Query type 1: Upon entering a student ID, each course and grade of that student are displayed in the RecyclerView, eg., if you enter "123" all of Bart's courses and grades should be shown.

Query type 2. Upon entering a subject ID, all courses and grades of each student with ID greater than or equal to the entered ID are pulled along with their names and displayed in the RecyclerView, eg. if you enter "456," the output should be "Milhouse, Math, B+", "Lisa, Physics, A+" etc. Note that the name associated with an ID, and the grades of that ID are in two different children in the json file.

The app should open the PushActivity when the "Push" button is pressed. Finally, a "Sign out" button will enable sign out.

XML: The layout of PullActivity will consist of an editText to capture the student ID, and two buttons for "Query 1" and "Query 2". There should also be a button called "Push" to start the PushActivity, and one called "Sign Out" to sign out. You can place all these buttons in a horizontal LinearLayout. The output of the queries will appear in a RecyclerView (preferred) that appears below the row of buttons.

JAVA: When Query 1 is pressed, obtain the relevant data as objects, and cycle through them using a for loop to add the relevant ones to the Recycler. In the case of Query 2, pull the relevant grade object, find the ID, and then pull the relevant student object using the ID to obtain the name of the student. When the “Push” button is pressed, the PushActivity must be started. Finally, pressing the“Sign out” button will cause sign out.

PushActivity

This activity is used to push new grades into the database. First you must check if the user is authenticated (i.e., not null). The four students will be selected using radio buttons. The fields for a grade object will be entered in three EditTexts (course_id, course_name, grade), and, along with the appropriate student_ID, will be pushed into Firebase using a randomly generated key. Immediately after this, the app returns to the PullActivity.

XML: The layout of the PushActivity will consist of a RadioGroup with four RadioButtons corresponding to Bart, Ralph, Milhouse, and Lisa, and three editTexts to capture (course_id, course_name, grade), which along with the right student_id (corresponding to the selected student in the RadioGroup) will be used to create a grade object that will be pushed into Firebase. A button named “Push” will enable you to push the grade object to Firebase.

JAVA: When “Push” is pressed, the strings from the three editTexts is captured, and the correct student ID is determined by using the state of the RadioButtons. Using these four fields, a grade object is created and pushed into Firebase with a random key. Finally, the app returns to the PullActivity.

Grade Class

You will need to create a class called grade. See code corresponding to Authentication and Queries On Firebase on the course website for relevant code.

3 Evaluation Guidelines

1. Check operation using the interface.
2. Your application must run on either a physical or a virtual device.