CS 646 Android Mobile Application Development
Fall Semester, 2015
Assignment 3

San Diego State University -- This page last updated 2/26/15

Assignment 3 - Rate The Instructor

Points 130
Due March 15 23:59

Objectives

Table Views, Navigation, network connections

App description

You will build an Android app that accesses information about instructors and lets people rate them. Use a list view to present some identifying information about each instructor. When the user selects an instructor they get a view of more detailed information about that instructor. There also is a way to rate an instructor on a scale of 1 to 5 (integer values only) and provide comments about an instructor. The information available about an instructor is:

- First name
- Last name
- Office
- Phone
- Email
- Rating
- Comments

The information is available in JSON format using a REST-like interface via http using the following urls.

GET urls

http://bismarck.sdsu.edu/rateme/list

Returns an json array. Each element of the array is a json object (dictionary). The keys of the object are:

id - value is a number
firstName - value is a string
lastName - value is a string

The id is the identifier to be used when referring to the given instructor. Here is an example of an object:

{"id":2,"firstName":"Dr. Leland","lastName":"Beck"}

http://bismarck.sdsu.edu/rateme/instructor/n (where n is replaced by an instructors id)

Returns an json object containing the data for the instructor with id “n” (replace n with the actual id of an instructor). All the keys are shown in the example below. The rating key has a value that is a json object. A concrete request would be:

http://bismarck.sdsu.edu/rateme/instructor/2

A sample response is:

'{"id":2,"office":"GMCS 407B","phone":"619-594-6191","email":"beck@cs.sdsu.edu","rating":{"average":5.0,"totalRatings":12},"firstName":"Dr. Leland","lastName":"Beck"}'

http://bismarck.sdsu.edu/rateme/comments/n

Returns the comments for the instructor with id "n" (replace n with the actual id of an instructor). The result is a json array of objects. Each object represents one comment. The object contains keys "text" and "date". The date is the date the comment was submitted to the server. The text is the actual comment. Below is a example.

'[{"text":"Good Instructor","date":"10/26/11"},{"text":"do it","date":"10/25/11"},{"text":"cat","date":"10/24/11"},{"text":"He is cold","date":"10/24/11"},{"text":"He is hot","date":"10/24/11"}]'

Post URLs

http://bismarck.sdsu.edu/rateme/rating/n/k

Adds the rating "k" to the instructor with id "n". "k" is one of the values 1, 2, 3, 4, or 5.

http://bismarck.sdsu.edu/rateme/comment/n

Adds a comment to the instructor with id "n". The comment is the body of the post.

Extra Credit

One common issue with apps that access data from the network is accessing the same data repeatedly. For example assume that the user first looked at instructor A, then instructor B, then instructor A again and then instructor B again. The app would then end up fetching instructor A’s data twice, instructor B’s data twice and the list of instructors four times. If any of this data is rather large the app will be slow to show the data each time and will consume the users downloading the same data multiple times. A common solution is to cache the data locally. The first time the user accesses instructor A’s data one puts it into a local cache. So the second time the user wants to view the data we can read the data from the cache instead of downloading it a second time. This does make the app more complex. Adding a cache for the instructor’s data (basic data, comments, rating) is one of the extra credit parts of this assignment.

Another issue is dealing with being off-line. If the app needs to access the network to get the data, then the app does not work when the device is off-line. One solution to this is each time the app accesses data from the network the data is stored in a database on the device. Then when the device is off-line the app uses the data stored in the local database. One needs to do this in a way that does not slow the app down. Sometimes students will load all the data from

the server into the device database the first time the app starts. If you consider the app being used to rate all the instructors at SDSU this loading all at once becomes unpractical.

Grading

The assignment will be graded as follows:

Points	Item
35	Network connections
25	List View for Instructors
25	Instructor detail view(s)
30	Posting rating & Comments
15	GUI look
5	Cache values in memory (Extra Credit)
20	Store copy of Instructor list, ratings and comments in a database on the device (Extra Credit)

What to Turn in

Create a Xcode project for the assignment. Xcode places the project in its own directory. Place the directory (and all its contents) into a zip file. We will use the course portal (http://bismarck.sdsu.edu/CoursePortal).