Ser423/Cse494 Mobile Computing

Lab 8 iOS Movie Collection with Core Data

Due by midnight Thursday April 7, 2016. No late submissions will be accepted.

Changes since distribution:

- No changes.

Description

Recast your Movie Library app to work with **iOS Core Data**. Your solution to this problem should be an **Xcode** project that includes the schema for your movie persistent storage. And, your **iOS** app should use **Core Data** to persist movies. Your app should provide the ability to add new Movie Descriptions, remove existing Movie Descriptions, and to view/browse the movies that are stored in the app's core data. Your app should also provide the ability to search Open Movie Database API. Use the **Json** representation returned by a successful search, providing the user the ability to add new descriptions into the apps database. For example with the search http://www.omdbapi.com/?t=Frozen&y=&plot=short&r=json returns the Json of an entry, whose details should be displayed by your app. Include at least the following fields: **Title, Year, Rated, Released, Runtime, Genre, Actors, Poster,** and **Plot**. You may limit the Genre you save to the first in the list returned by the **API**.

What To Hand-In

The same packaging, documentation and solution formatting constraints as with the prior assignments apply to this assignment as well.

1. Place all of your application source code into the package named: **edu.asu.bsse.asuriteid.appname** where **bsse** designates your academic program: **bscse**, **msse**, **bsse**, **bscs**, which stand for **BS Computer Systems Engineering**, **MS Software Engineering**, **BS Software Engineering**, or **BS Computer Science** respectively.
   asuriteid is your asurite id.
2. As part of all class header comments, that you create and turn-in this semester, include a copyright notice, such as: **Copyright 2016 Your Name**.
3. As part of the class header comments, include a **right to use** statement. The examples presented in class use the Apache License Version 2, but you should put whatever rights you prefer. At the very least, you must provide the instuctor and the University with the right to build and evaluate the software package for the purpose of determining your grade and program assessment.
4. As part of the class header comments, include a reference to the software's author, such as: **@author FirstName LastName mailto:FirstName.LastName@asu.edu**.
5. As part of the class header comments, include an indication of the software version, such as: **@version March 2, 2016**

These comments are required on all code that you generate this semester. If you hand in an assigment that

does not include them, it will not be graded.

Structure your project as a sub-directory of the folder named **Assign8MyASURITEID**. Use the same folder structure as used in the sample app: studentCourseCoreData.jar

You will submit this project, by first cleaning its iOS app to remove all generated files. Then create a **jar** or **zip** archive of the project (Assign8*) directory. You can create a **Java Archive (jar)** by executing the following command from a terminal in the directory which is parent to the project directory:

jar -cvf Assign8MyASURITEID.jar Assign8MyASURITEID/

That archive will then be submitted via **Blackboard**. See the Content section.

Grading Criteria

- **5 points**. Your core data schema must be defined to include at least one entity for a movie, and should contain at least the attributes defined above. You may want to define more than one entity with a relationship between, which will account for 2 of the 5 points.
- **5 points**. Your **iOS** app should define and properly use network requests off of the UI/Main Thread to search the open movie database using one of the approaches discussed in class.
- **5 points**. Your iOS app includes comments described above as headers for each class used in the app.
- **5 points**. Your app provides the ability to list all movie titles, display the details of a movie description, search by movie title and display successful results, as well as to add and remove movie descriptions.
