Homework #3

Complete By: Tuesday, July 7th @ 11:59pm
Assignment: completion of following exercise
Policy: Individual work only, late work *is* accepted
Submission: electronic via Blackboard

Overview

The exercise is to build a Windows Forms Application in C# that allows the user to buy songs and albums from your uiTunes database. You'll need to insert enough data in your database to test your application: songs, albums, users, and reviews.

Application Functionality

You have reasonable freedom in how you design your user interface (which is why a screenshot of the application is not provided), as long as you provide the following functionality:

1. The user enters their first name, last name, and password. If their name and password matches, display their account balance.
2. All songs can be displayed, e.g. in a listbox.
3. All albums can be displayed, e.g. in a listbox.
4. When a song or album is selected, the following information is display:
    a. Artist(s)
    b. Release date
    c. Price
    d. Total number of reviews (0 if none)
    e. Average review (as a real number, N/A if none)

5. Ability to purchase selected song or album:
    a. Has song or album already been purchased? If so, inform the user
    b. Does the user has enough credit to purchase song or album? If not, inform the user
    c. Otherwise make the purchase and update the database

Keep the interface simple — list boxes, text boxes, and buttons — and focus on the DB interaction. As time permits, improve your interface as desired.

Assignment Details

You must build a Windows Forms Application, using C# and ADO.NET. You must also dynamically execute raw SQL queries; avoid other options provided by C# and Visual Studio (such as LINQ and data-binding, which we'll discuss later).

I would recommend displaying all the songs / albums using a list box. For an example of list box programming, see the ***DoctorsApp*** discussed in class on Thursday, July 2nd. This app is available for download from the course web page: see Lectures\07-02\Solution\DoctorsOnCall.zip. Here's a summary of how to work with a list box object:

- Fill the list box when the app starts by coding the *Form_Load* event handler. Double-click on the form's background to stub out this event handler; add items (which must be strings) to the list box by calling listBox1.Items.Add().

- When the user selects an item in a list box, this triggers the list box's *SelectedIndexChanged* event. Double-click on the list box to stub out an event handler. To retrieve the text denoting the value selected by the user, use the *Text* property (e.g. *listBox1.Text*).

A few other tips:

- Text boxes used for display only should be read-only; set the *ReadOnly* property to True. Text boxes used for passwords can be set to hide the password chars; see the PasswordChar property.

- Disable buttons in the Form_Load event handler, and then enable at the appropriate time (e.g. disable Buy button until a song or album is selected?). Use the *Enable* property.

That should be enough to create a reasonably-nice user interface.

Electronic Submission

When you are ready to submit, first make sure there's a header comment at the top of your main form's C# code file:

```
//
// uiTunes music purchase application
//
// <<YOUR NAME HERE>>
// U. of Illinois, Chicago
// CS480, Summer 2015
// Homework 3
//
```

Then create a .zip of your entire Visual Studio project folder; we want it all, including your database. Finally, submit using Blackboard (http://uic.blackboard.com/) using the link for "**HW3**". You may submit as many times as you want before the due date, but we grade the last version submitted. This implies that if you

submit a version before the due date and then another after the due date, we will grade the version submitted after the due date — we will *not* grade both and then give you the better grade. We grade the last one submitted. In general, do not submit after the due date unless you had a non-working app before the due date.

Policy

Late work *is* accepted. You may submit as late as 24 hours after the deadline for a penalty of 25%. After 24 hours, no submissions will be accepted.

Unless stated otherwise, all work submitted for grading *must* be done individually. While I encourage you to talk to your peers and learn from them, you cannot submit their work — whether partial or complete — as your own. The University's policy is described here: http://www.uic.edu/depts/dos/docs/Student%20Disciplinary%20Policy.pdf . In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of 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 failure to expulsion from the university; cases may be submitted to the official student conduct process: http://www.uic.edu/depts/dos/studentconductprocess.shtml