CS6326 Human-Computer Interactions

Fall 2016

Assignment 2

Since this is a course in user interface design, your assignment is to write a simple program to maintain a file of data. This is a very common thing to write in the "real world," but it's not as easy as it looks. Consider an operation that processes rebates*. The buyer sends in a form with various information and a clerk enters it into a program to be stored in a database. Your job is to write that program. Specifications are as follows:

1. The file (and therefore the screen) will contain the following fields:
    a. First name (20)
    b. Last name (20)
    c. Middle initial (1) (may be blank; entry not required)
    d. Address line 1 (35)
    e. Address line 2 (35) (may be blank; entry not required)
    f. City (25)
    g. State (2)
    h. Zip code (9)
    i. Phone number (21)
    j. E-mail address (60)
    k. Proof of purchase attached (Yes/No)
    l. Date received (default to today but changeable)
2. Your program must provide three functions:
    a. **Add** a new record to the end of the file.
    b. **Modify** an existing record and write it back to the file. This means that you can modify **all** fields of the record, including the name.
    c. **Delete** a record from the file.
3. Your program must not be able to add a record that contains the same first name, last name, and middle initial as a record already in the file. In general, your program must not be able to enter bad data into the file. That is, dates must be checked for validity, and so on.
4. **Data file handling:** Below are the requirements (not suggestions) for the data file.
    a. Since this is not a database class, you will use a flat text file with one record per line. One way to do this is to read the entire file into memory when your program starts and write it back out, with changes.
    b. If the file does not exist, your program must create it.
    c. The name of the file is **CS6326Asg2.txt**. You will lose five (5) points for using any other file name.
    d. **Use the tab character** as a field separator rather than comma, for obvious reasons.
    e. The fields must be written in the order given above, with two additional fields given below that are not entered by the user.

    f. When the program starts, read the entire file. When a user saves data, write the entire file. Make sure that you **don't hard-code the path of the file**, just the name, or we will not be able to run it. You will lose 20 points for this.
5. "But C# has database functionality. Can we use that?" No.
6. You may not read everything into a datagrid or something like it and treat it as a spreadsheet. You must have separate fields for the various data elements.
7. You should show the full name and phone number **only** (not every field) in a multi-column scrollable list. Clicking on a list item should put all of its data elements into the fields for modification. That is, if a name is selected from the list, the clerk should be able to save the changes or discard them.
8. All three functions must be done from the same main screen. There should be no menu and no multiple screens.
9. Your program must be written in C# using only packages that are installed as part of Visual Studio. Do not use any third-party controls that we must load in order to compile and run your program.
10. Your program must apply good object-oriented methodology. That is, the user interface, application logic (what little there is in this program) and the technical services layer must be separate classes.
11. This is a user interface assignment, so pay particular attention to how your program looks and how easy it is to use. Minimize the number of keystrokes necessary. Apply principles you have learned from the reading and lectures.
12. This is an individual assignment. All code other than standard libraries and APIs must be your own.
13. Two of your "database" fields must be the time the user started entering a record (adding new records only) and the time the save button was pressed. The start time is recorded when the first key is pressed on the first field. These must be recorded as hours, minutes, and seconds using a 24-hour clock. For example, a possible start time would be "13:04:01"
14. Program naming conventions: Your project must have the name Asg2-<netid>. That is, if I were to hand it in I would name it Asg2-jxc064000. The executable will then also have that name.

I strongly suggest that you ask yourself many "what if?" kinds of questions about how this program may be used and the kinds of errors a clerk might make. Write it and test it. If you finish the code on the day it is due you will almost certainly not get full credit.

This is a complex assignment. Read it carefully before you start, then refer back to it as you design and write, and finally, once the program is complete, make sure you have met all of the requirements.

To hand in through eLearning: Your entire Visual Studio project and a text file with at least two names and addresses you have entered, created by the program. This should contain the executable in the \bin\Release directory.

Grading (total points: 100):

Meets technical requirements and OO design	30%

Does not crash on invalid input	20%
Clean user interface	40%
Program comments	10%

*Rebates are money sent back to someone who purchases something. Let's say you go to Electron Hut and buy a $25.00 flash drive that gives you a $5.00 rebate. You fill out the form, similar to the one above, and send it in along with proof that you actually bought the flash drive. Many weeks later, you receive a check for $5.00 in the mail.