420-N33-LA .NET Development

Term Project – Contact Manager

Date due: December 20, 2022.

Champlain COLLEGE SAINT-LAMBERT

Description and Purpose

This term project will allow us the opportunity to integrate collections, database access, and WPF in a complete and relevant desktop application that uses object-oriented principles and design best practices.

When creating new projects, always select the type: .NET Framework, not .NET Core!

Learning Objective(s)

1. C# Syntax
2. Collections
3. ADO
4. WPF
5. Object-oriented Principles
6. Object-oriented design best practices

Pre-requisites

SQL Server Database installed with database tables created.

Requirements

- ☐ You must work in your assigned teams of 2-4 people.
- ☐ You must use Git to provide version control. Send an invite as a collaborator to bwood-crc for verification, failure to do so will result in a 0 for the project.
- ☐ The goal of this project is to create a contact manager. What is a contact manager? It is an application that allows a user to add contacts, edit contacts, view contacts and delete contacts. We call these CRUD (create, read, update, delete) operations.
- ☐ For viewing contacts, we will see a list of the existing contacts (contact names). When a user clicks on a contact in the list, a new window will be displayed that shows the details of that particular contact.
    - The contact data to be displayed and how it is displayed is up to you. However, it should be relevant and logical.
- ☐ The wireframes and screen layouts are NOT provided, I want to see how far creativity and usability takes you.

Your users will

1. Your contact manager must present a summarized list of all existing contacts.
2. Your contact manager must allow a user to add a new contact.
3. Your contact manager must allow a user to view an existing contact.
4. Selection of a contact in the list of existing contacts will open a new window displaying details of that contact.
5. You should be able to update a contact, preferably the screen should not be updatable until you click “edit”, then you can make changes and save the record.
6. You should also be able to update the email, phone and address for your contact.

7. The contact should show a list of addresses and phone numbers on their detail page, to edit this address, a third window should pop up showing full address details.
8. Your database functions must be all within a common database class.
9. You should be able to export a CSV file of the records in the database.

Resources, Reference, Please See

STARTER CODE

Code was started in class.

DATABASE INFO

1. There should be no redundancies in the database. (3rd normal form at least).
2. The delete function should update the active flag on a record (or set it to today's date if it's a date).
3. The last update date should be updated for all changes to a table.
4. The create date should show the date created (research how triggers work to accomplish this automatically).
5. To export your contact info, you might have to write a large "join" sql statement. Also you might research how "stored procedures" work to go the extra mile.
6. Create new tables as necessary.

Submission Procedure

The project is to be submitted via Moodle. The submission is to include:

1. The complete project in a zipped file.
2. Git link. Make sure your teacher (me) has access.
3. The database script file, including data.
4. A demonstration per group by video, please supply a YOUTUBE link only.

Grading Scheme / Rubric

Note: your code must run! If it does not, you will get 0.

0 points = Not submitted.

1 - 6 points – All requirements attempted.

7, 8, 9, 10 – level to which the requirements are met and to the quality of the code and design.