Faculty of Engineering and Information Technology
School of Software

31927 - Applications Development with .NET
32998 - .NET Applications Development

SPRING 2018
ASSIGNMENT-2 SPECIFICATION

DUE DATE – Thursday, 18th October 2018 (Extended)
DEMONSTRATIONS – Labs of Week 12

This assignment is worth 35% of the total marks for this subject.

Summary

This assessment requires you to develop a simple text editor with a basic offline user login. It is a windows application that will allow new uses to be added, existing users to login, and to create/save/edit text documents. The text documents can be either viewed or modified/created based on the user type. This will be a prototype of a text editor which will provide an overview of the basic underlying programming techniques required to create windows applications and text manipulation.
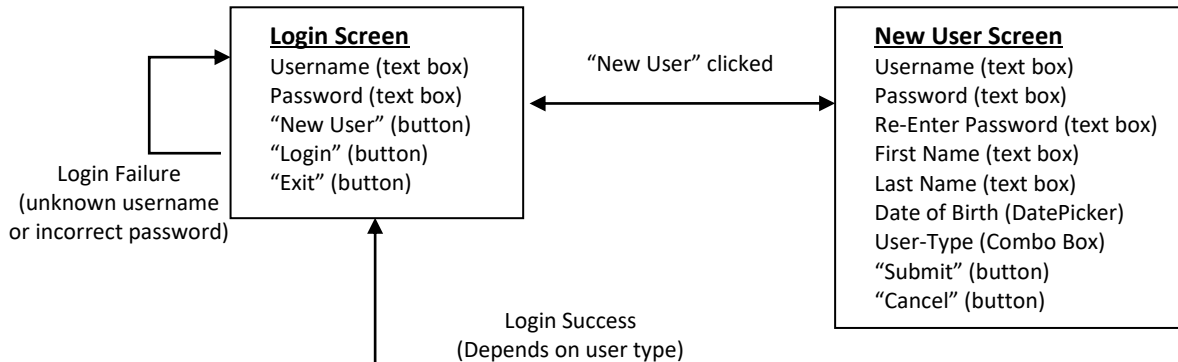
Assignment Objectives

The purpose of this assignment is to demonstrate competence in the following skills.

- ❑ Windows form and controls
- ❑ Communication between multiple windows forms
- ❑ Using collections/generics
- ❑ Enumerators and properties.
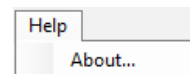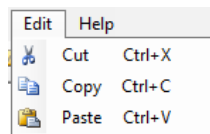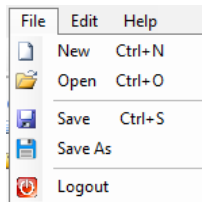- ❑ Text file reading and writing

Windows/Screens:

Each box is a separate screen which shows the visual elements that should be present. The direction of arrows specify the direction of flow between screens. The text in each box describes some of the components that should be in the screen, though you are free to add more if you feel they are needed / appropriate.

Login Screen
Username (text box)
Password (text box)
"New User" (button)
"Login" (button)
"Exit" (button)

"New User" clicked

New User Screen
Username (text box)
Password (text box)
Re-Enter Password (text box)
First Name (text box)
Last Name (text box)
Date of Birth (DatePicker)
User-Type (Combo Box)
"Submit" (button)
"Cancel" (button)

Login Failure
(unknown username
or incorrect password)

Login Success
(Depends on user type)

Text Editor Window

A.   Menu items required (MenuStrip): (Example shown below):

1.   File:
   a.   New
   b.   Open
   c.   Save
   d.   Save As
   e.   Exit

2.   Edit
   a.   Cut
   b.   Copy
   c.   Paster

3.   Help:
   a.   About

The menu item should show the icons along with text.

B.   Top Tool Strip (ToolStrip): (Example shown below):

The top tool strip should be placed below the menu. The first group of 4 buttons are for new, open, save, and save-as operations, along with a separator. The second group of 3 buttons are for text formatting, bold, italics, and underline. The ComboBox will be used to select the font size, to resize the selected text. The buttons should show the respective icons. Appropriate tool tip text should be added to each button. The tool strip Label shows the name of the currently logged user.

C.   Left Tool Strip (ToolStrip): (Example Shown below):

The tool strip should be placed in the left side on the window, below the top tool strip. It has 3 buttons, to edit text, such as, copy, paste and cut functionality, which can be applied to selected text.

D.   The Text editting area (RichTextbox):
A RichTextBox control should be added to the form and should be used for text editing. Use the RichTextBox properties as requried to achieve the functionalities.

Requirements:

In order to test core C# principles and windows programming skills, the following are required. Any additional functionality should be designed and added as needed.

1. Create appropriate class and methods for handling user login information.
    a. User information/details are stored in a text file named **"login.txt"** in the following format (each value is separated by a comma):
        Username, Password, User_Type, First_Name, Last_Name, dd-mm-yyyy(Date-of-birth)

    b. Use of appropriate collection/data structure is required for storing the user information from the **"login.txt"** file.

    c. The class should have appropriate variables to:
        i. Track the currently logged user and their user type
        ii. Accessors for each of the variables.

    d. The class should have methods to:
        i. Load all users from the **"login.txt"** file into an appropriate data structure.
        ii. Save all users back to **"login.txt"**, when a user logs out of the text editor.
        iii. Checking if a user name and password provided by the user is valid.
        iv. The type of user describes whether a user is allowed to edit any file. There are two type of users:
            - "View": The user can only view any text file. Text editing is disabled in the text editor window.
            - "Edit": The user can create/edit any text file.

2. **Login Form:**
    a. This is the first screen displayed when the program is launched
    b. All the user information from **"login.txt"** should be loaded into appropriate data structure/collection.
    c. When **login button** is pressed, the username and password in the textbox are checked with all the available user information. If the login is successful, the text editor form/window is displayed. In case of unsuccessful login, appropriate message is shown using MessageBox() and the login windows remains open.
    d. When **Exit button** is pressed, the application is closed.
    e. When **New User button** is pressed, the New user window/form is displayed and hides the login form.

3. **New User Form:**
    a. A user is allowed to enter the required information.
    b. When **submit button** is pressed, the information is added to the data structure/collection, and all information is saved back into the **"login.txt"** file.
    c. When **cancel button** is pressed, the Login form is display and hides the New user form.

4. **Text Editor Window:**
    a. Appropriate common dialog boxes should be used to implement the **open, save and save as** functionality available in the menu strip under **File** menu and also the respective tool strip buttons.
    b. **Open** menu will allow to select a text file and load all the text from the file into the RichTextBox.
    c. Save and Save as, will save all the text line in the RichTextBox into an **.rtf** file.
    d. The **Bold, Italic and UnderLine** buttons on the top ToolStrip are used change the selected text style to bold, Italic or UnderLine.
    e. The ComboBox on the top ToolStrip shows the **font size** from 8 to 20, and is used to change the size of the selected text in the RichTextBox.
    f. The **Cut, Copy and Paste** buttons on the left ToolStrip and **Edit** Menu are used to cut/copy/paste selected text in the RichTextBox.
    g. The About form should be used to implement the **About** menu item under **Help**.

h. On clicking **Logout** option from the **File** Menu, the text editor window should close, and the Login window should be displayed.
i. The RichTectBox should be disabled or enabled for editing based on the user type.

5. **About Window:**
   a. Should display appropriate information about your text editor, such as name, version number, etc.
   b. On clicking OK button, the **About** form should close.

Marking Guide:

Below is the marking guide for this assessment. It is designed to allow you to get a Pass grade with minimal effort while still demonstrating that you understand the core principles of .NET development, to get a Distinction with reasonable effort, and to get a High Distinction with solid effort, and 100% with considerable effort. It is recommended that you pay attention to the grade distribution and work towards your own skill level.

During demos in the lab, your code needs to be compiled in Visual Studio and then the tutor will test for normal functionality as described in the descriptions above. If your code does not compile you will receive zero marks. You need to demonstrate that you understand the functionality of various components in a Windows form and your forms should be easily readable and usable by your tutor.

Task	Items	Max Points	Grade (Assuming full points above this item)
Code Design (Programming style)	Includes high cohesion and low coupling for classes and methods, using properties, using enumerations, collections where appropriate, etc.	**3**	**9%**
Code Quality (Programming style)	Includes proper indenting and white spacing, helpful comments and meaningful class/method/property/field names.	**3**	**18%**
Design	Includes form/window designs, with basic transition between forms/windows on button clicks or as appropriate: - Login: 2 marks - New User: 3 marks - Text Editor: 6 marks - About: 1 mark	**12**	**52%**
Functionality (in order of recommended implementation, e.g. start with items higher on the list)	*Text Editor Window* - Required Classes, Methods, and Member Variables are correctly implemented, in order to achieve the overall functionality of a simple text editor and code compiles without error. - All File menu options implemented: 4 marks - All Edit menu options and buttons: 2 marks - All text editing buttons (bold, italics, etc): 4 marks	**10**	**80%**
	Login screen working correctly by reading from and writing to *login.txt* file instead of a hardcoded User list	**3**	**89%**
	Login Screen – successful login and failed login with sample User details are hard coded and appropriate data structure used.	**2**	**94%**
	New User Screen – successfully add new user and allow that user to then login	**2**	**100%**
	Total	**35**	