

Faculty of Engineering and Information Technology

School of Computer Science

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

SPRING 2021

ASSIGNMENT-2 SPECIFICATION

Due date	Monday 11:00am, 24 October 2022
Demonstrations	Required in the lab/tutorial session
Marks	35% of the total marks for this subject
Submission	Complete project folder zip (Code, solution files, etc.), any instructions to run the program in a text file, all in 1 single zip file.
Submit to	Canvas assignment submission
Note: This assignment is individual work.	

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 users 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.

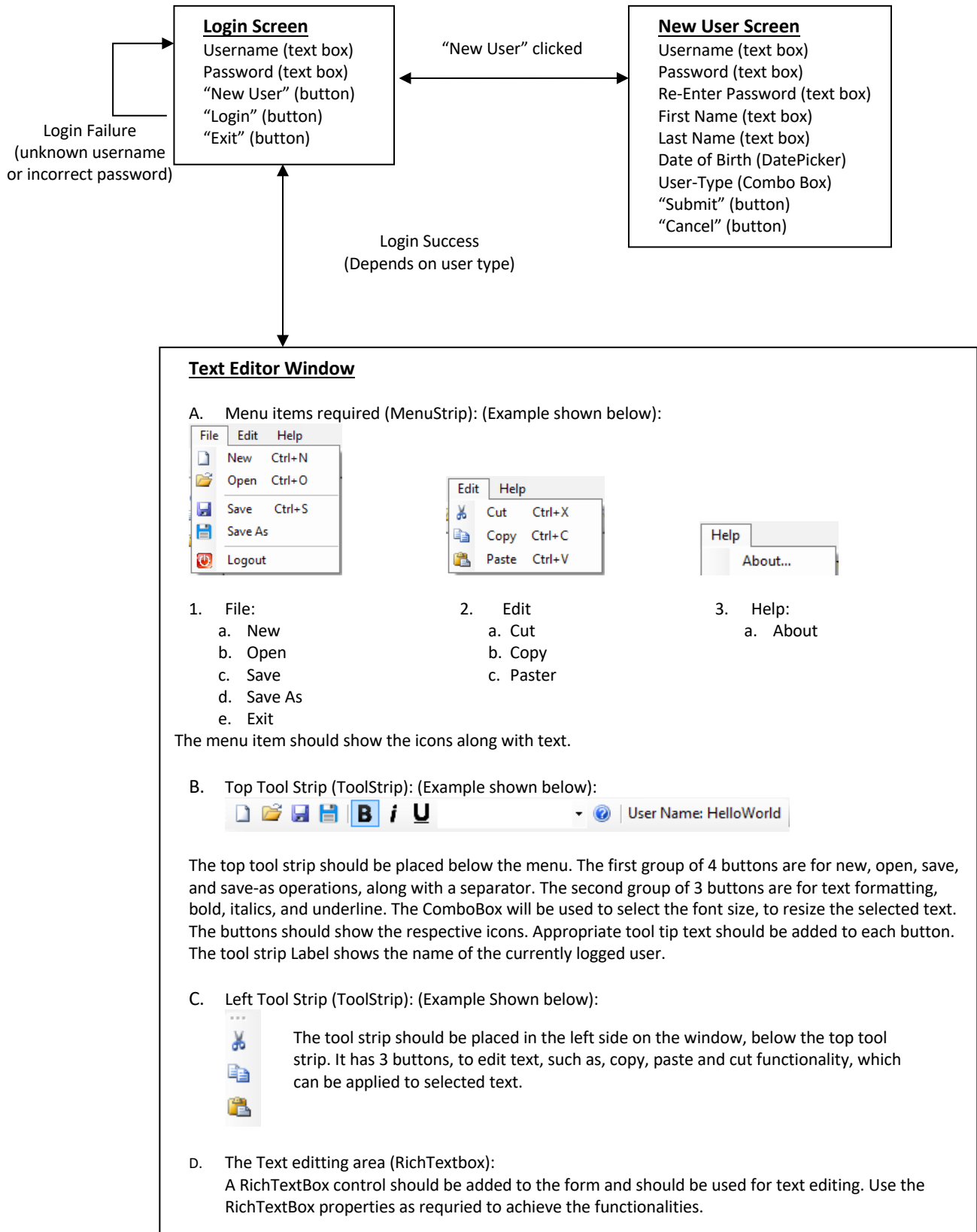
Assignment Objectives

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.



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 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 RichTextBox 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.

In the 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: <ul style="list-style-type: none"> - 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)	<i>Text Editor Window</i> - 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. <ul style="list-style-type: none"> - 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%
	<i>Login screen</i> working correctly by reading from and writing to <i>login.txt</i> file instead of a hardcoded User list	3	89%
	<i>Login Screen</i> – successful login and failed login with sample User details are hard coded and appropriate data structure used.	2	94%
	<i>New User Screen</i> – successfully add new user and allow that user to then login	2	100%
	Total	35	

Additional Information:

Assignment Submission:

You must upload a zip file of the C# solution to Canvas. This must be done by the Due Date. You may submit as many times as you like until the due date. The final submission you make is the one that will be marked. If you have not uploaded your zip file within 7 days of the Due Date, or it cannot be compiled and run in the lab, then your assignment will receive a zero mark

NOTE 1: It is your responsibility to make sure you have thoroughly tested your program to make sure it is working correctly.

NOTE 2: Your final submission to Canvas is the one that is marked. It does not matter if earlier submissions were working; they will be ignored. Download your submission from Canvas and test it thoroughly in your assigned laboratory.

Queries

If you have a problem such as illness which will affect your assignment submission contact the subject coordinator as soon as possible.

Dr. Avinash Singh
Room: CB11.07.108
Phone: 9514 4426
Email: avinash.singh@uts.edu.au

If you have a question about the assignment, please post it to the Canvas discussion board for this subject so that everyone can see the response.

If serious problems are discovered in assignment specification the class will be informed via an announcement on Canvas. It is your responsibility to make sure you frequently check Canvas.

PLEASE NOTE : If the answer to your questions can be found directly in any of the following

- ☐ Subject outline
- ☐ Assignment specification
- ☐ Canvas FAQ and addendum
- ☐ Canvas discussion board

You will be directed to these locations rather than given a direct answer.

Extensions and Special Consideration

Please refer to subject outline.

Academic Standards and Late Penalties

Please refer to subject outline.