.NET Technology

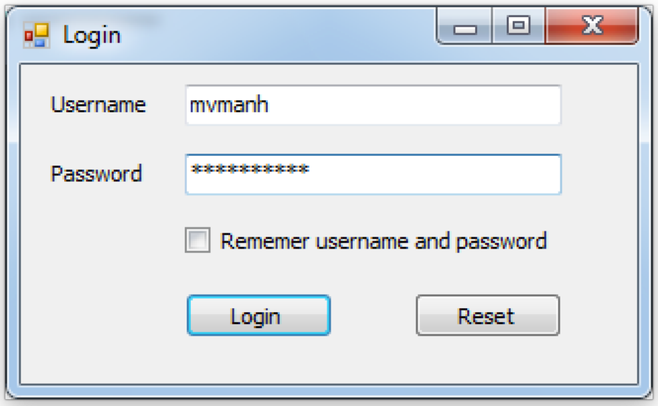
(503112)

By Mai Van Manh

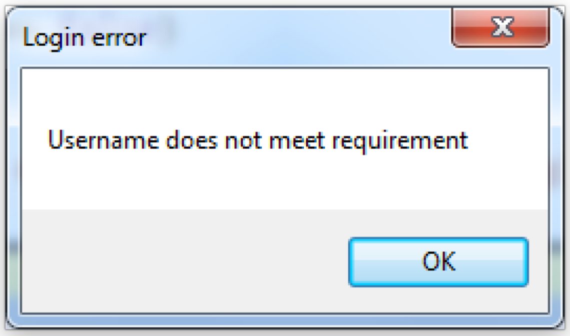
Lab 04: Windows Forms

Exercise 1: Basic Windows Forms

Design the login application as shown below:

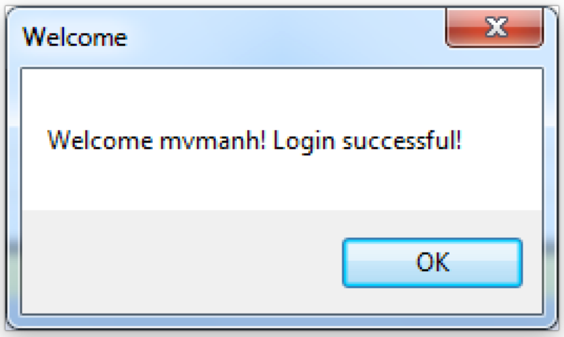


1. When the user clicks the Login button, if the username or password has less than 6 characters, the corresponding message will be displayed.



Login error

Username does not meet requirement



Welcome

Welcome mvmanh! Login successful!

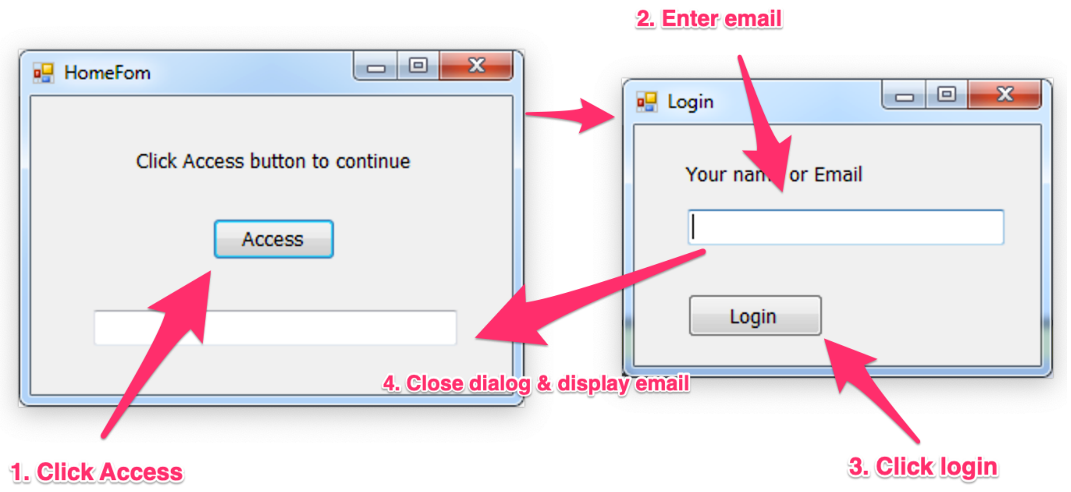
2. When pressing the Login button, if the username and password satisfy, the message "Congratulations username! Successful login".

Exercise 2: Passing Data Between Forms

Students are provided with a project named '503112-Lab4.2.zip' to do the following exercises.

You must use delegate to complete this exercise. You are provided a Windows Forms Application which contains two forms: home and login.

At the home screen, when the user clicks the **Access** button, the login screen appears asking the user to enter their name or email address. After the name or email address has been entered, the user clicks on the **Login** button, the login screen will close, and the entered name or email address will be displayed in the home sreen textbox.



Hints

- In the main screen, create a method that accepts a string as a parameter and then update the textbox value according to the input string. Create a delegate object which references to the previous created method.
- Send the delegate object to the second screen via constructor or setter.
- At the second screen, when the user clicking on the login button, invoke the delegate object to update the UI of home screen.

Exercise 3: Working with Events

You are provided a windows form application that allows you to download a file from the given url and save it in the local directory. However, this example uses synchronous programming approach that blocks responsiveness while download is in progress.

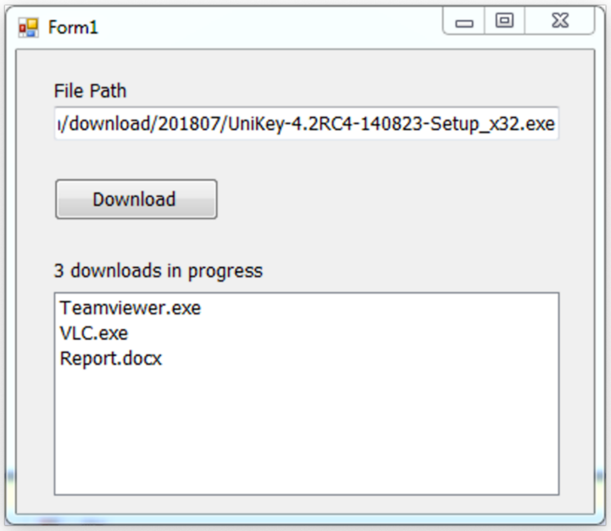
Requirements: change from a synchronous approach to an asynchronous approach. Use the events provided by the WebClient to display the download progress on the progress bar and show the dialog box to notify users after the download process is complete. Use lambda expression to register to those events if possible.

Hints

- Change from DownloadFile() to DownloadFileAsync().
- Use DownloadProgressChanged event to update download progress.
- Use DownloadFileCompleted event to notify user when download complete.

Exercise 4: Multithreading

Given a windows form application as shown below. Please use Thread to complete the application that works as a file downloader.



ı/download/201807/UniKey-4.2RC4-140823-Setup_x32.exe

3 downloads in progress

Teamviewer.exe
VLC.exe
Report.docx

Requirements: When users fill in the link and click the **Download** button, a new thread will be created to execute the download process. Many files can be downloaded at the same time and they are downloaded in separate threads. When the download is in progress, the name of the file will be listed in the listview. After the download is complete, the file name will be removed from the listview.

Because we are using different threads to do the work, we can use WebClient.DownloadFile() to download the files in each thread, it will not block the UI.

Exercise 5. The List<T> collection is a very good choice for storing dynamic array in C#, but it does not support events. Please write a new class which extends from List<T> and provide some events like OnAdd, OnRemove, OnInsert ... Write a windows form application which uses your new class to demonstrate your result.