Programming II

Lab Assignment #4 – Implementing a Student Application

Due Date: **Thursday 29th Nov. 2018 before 6:00pm**

Purpose: The purpose of this Lab assignment is to:

- Practice the use of Files and Streams to save and retrieve data

References: Read the course's text book chapter 14 – Files and Streams and the lecture notes/ppts/examples. This material provides the necessary information that you need to complete the exercises.

Instructions: Be sure to read the following general instructions carefully: This lab should be completed individually by all the students. You will have to demonstrate your solution in a scheduled lab session and submitting the project **through drop box link on e-Centennial**. You must name your Visual Studio solution according to the following rule:
FirstName-lastName_SectionNumber_COMP123_LabAssignmentNumber
For Example: **John-Smith_Sec002_COMP123_Lab01**

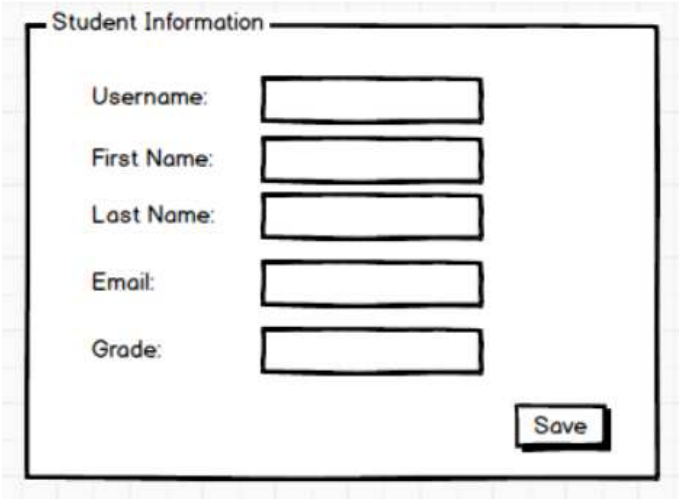
Each exercise should be added to the solution as separate project. Your IDE is Visual Studio 2017 and C#.

Submit your assignment in a **zip file** that is named according to the following rule:
FirstName-lastName_SectionNumber_COMP123_LabAssignmentNumber.zip
Example: **John-Smith_Sec002_COMP123_Lab01.zip**

Apply the naming conventions for variables, methods, classes, and namespaces:
- *variable names* start with a lowercase character for the first word and uppercase for every other word
- *classes* start with an uppercase character of every word
- *namespaces* use only lowercase characters
- *methods* start with a uppercase character for the first word and uppercase for every other word

Exercise 1:

Write an app that will allow the user to enter the following student information:

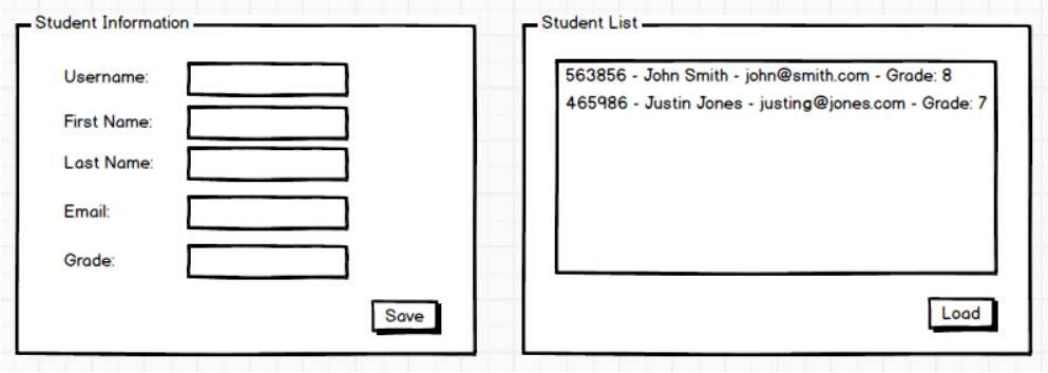


- Username (int)
- First Name (string)
- Last Name (string)
- Email (string)
- Grade (double)

1. Each field will have its own `TextBox` control.
2. At the bottom of the fields there should be a "Save" `Button`. When clicked it should save the student information into a text file called "StudentList.txt" as a CSV file.
3. If the file doesn't exist, make sure you create it.
4. If the file exists, make sure you add the new student record to the bottom of the file.
5. Create a Student Class to be used in the code.

Exercise 2:

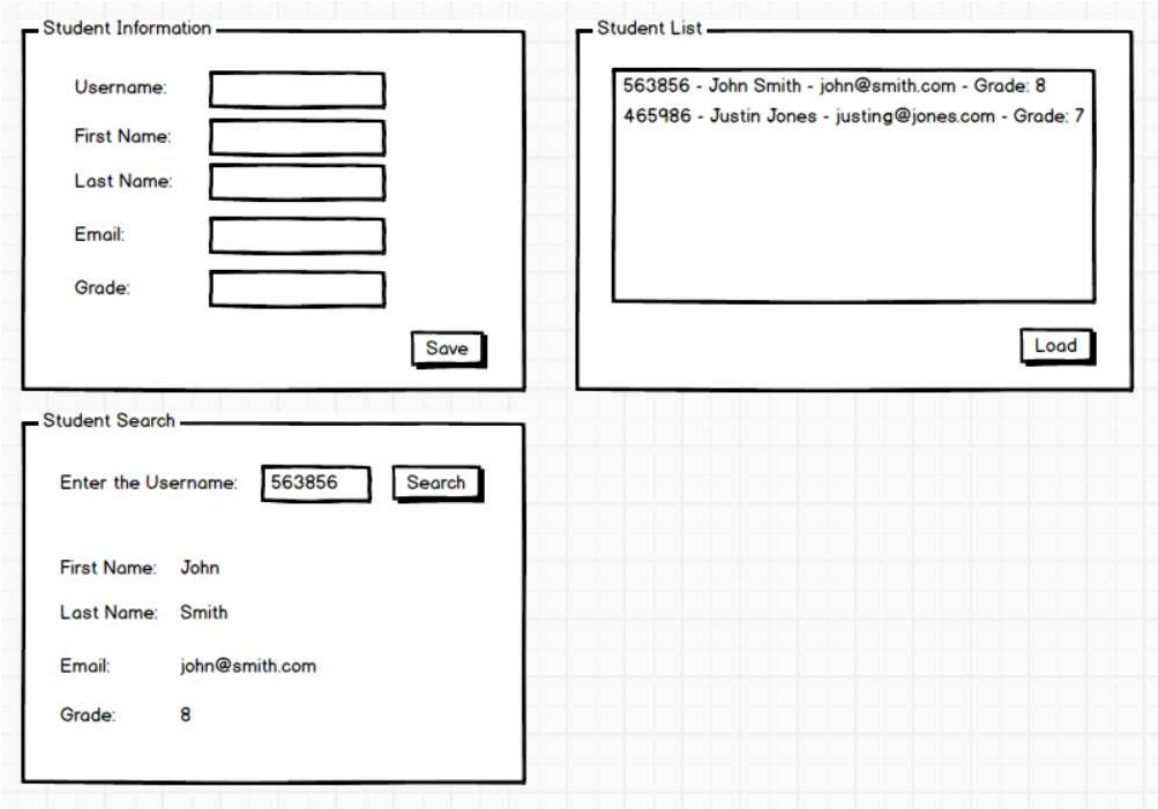
Enhance the app you have created on Exercise 1 and include a Student list section:



1. When the user clicks the "Load" `Button` the list of students stored in the file "StudentList.txt" should be read and displayed in a `ListBox` above the Button.
2. Go back to the "Save" button and when the use clicks on it, it should also add the student to the Student List control.
3. It is expected that you use the Student class when getting the information from the "StudentList.txt" file.

Exercise 3:

Enhance the app you have created on Exercise 2 and include a Student Search section:



1. When the user enters the Username on the TextBox and clicks the "Search" `Button`, go through the list of students in the file to find the Username that was entered;
2. If you find the student, display the student details on the labels below the "Search" `Button`
3. If you don't find the student, provide a message that "The student was not found"

Evaluation:

Functionality	
Correct implementation of classes (instance variable declarations, constructors, getter and setter methods etc.)	65%
Correct implementation of student class (declaring and creating objects, calling their methods, interacting with user, displaying results)	20%
Comments, correct naming of variables, methods, classes, etc.	5%
Correct implementation of the student search	5%
Friendly input/output	5%
Total	100%