**COSC 1200 – Object Oriented Programming I (Winter 2024)**

**Assignment 5: Vet Clinic Patient Registration**

**General Requirements**

1. This assignment can be completed EITHER in a group of 2 or individually (your choice).

2. Submit your solution to the appropriate assignment folder on DC Connect by the due date provided.

3. Your instructor will assign a grade and post feedback on your submission to DC Connect.

4. Note: Each class needs comments at the top with at minimum your name(s), the date (date created or date due), a description of what the class does. In addition, it is expected that you provide a sufficient amount of commenting in your code.

**Requirements**

Complete and submit the program source code that satisfies the following requirements:

For this assignment you are to create a form. This form will take in new patient information including patient name, ­, a valid email address, and the vet assigned to that patient. Only once all input has passed validation, that information will be written to a file. The form must look professional, but you can have fun with colour (as long as it looks professional). The input elements on the form must flow initiatively, it must include tool tips on all necessary elements (text field, buttons, etc), and you must include keyboard shortcuts (key bindings).

**The Form and Form Elements:­­­**

* Make the form an appropriate size, not too big but also not too small that everything is squished together.
* The form should not be resizable.
* The form must load center of your screen.
* The form must have a custom title like “New Patient Registry” for example.
* The form must also display a custom icon in the top left corner that replaces the standard Java icon.
* The exit button in the top right corner must be functional.
* There must be 3 text field elements to take input for patient name, owner name, and email address. Each of these text field must have a corresponding label to provide the user with some instructions as to what they need to enter. For example, “Patient Name:” for the patient text field, etc.
* There must also be 2 radio buttons for the 2 veterinarians that are taking new patients. You should use your first name and last name to make up their names. Set the first radio button to be checked automatically when the form loads. Also, to avoid multiple selections, these radio buttons must be grouped (Hint: use the ButtonGroup class)
* There must be 3 buttons to allow the user to register a new patient, clear the form, and exit the form.
* There must also be a label that will display that the file was successfully written to or not. This label can also be used to display error messages.
* Assign appropriate descriptive tooltips to appropriate form elements (text field, buttons, etc).
* Assign keyboard shortcuts (key bindings) to the 3 regular buttons.

**Input Validation:**

No processing should occur (writing to the file) unless all input is valid.

Patient name cannot be empty, if empty then display an error message in the label or in a message box.

Owner name cannot be empty, if empty then display an error message in the label or in a message box.

Email address cannot be empty. Email address must be a valid email address. If empty or invalid, then display an error message in the label or in a message box. Valid email info: Create a method that takes a string (the email address) and processes it to ensure it is valid. A Boolean method may be a good idea but I’ll leave that up to you. Unless you find something you like better, you can use the following:

String regexPattern = "^[\\w!#$%&'\*+/=?`{|}~^-]+(?:\\.[\\w!#$%&'\*+/=?`{|}~^-

]+)\*"

+ "@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,6}$";

Pattern pattern = Pattern.compile(regexPattern);

**For the above you will need to import:**

import java.util.Date;

import java.util.regex.Pattern;

If you use the label to output error message, ensure you make it large enough. You can display them individually or concatenate them and display them all at once.

Once all data has passed validation, check which radio button has been selected and assign that vet’s name to a string variable.

Grab the current date and save that to a variable with an appropriate data type (Hint: Might be good to create a new date object for this? I’ll leave that up to you to Google)

**Writing to the File:**

Once all data has passed validation, write the title “\*\*Patient Registration Document\*\*”, the patient’s name, owner name, email address, selected vet, and current date to a file. Name the file newpatientfile.txt or output.txt or something like that.

Remember this needs to be in a try catch to ensure possible exceptions are handled.

If the write is successful, display a message to the label that the write was successful. If an exception is throws, display an error message to the label that the write was not successful.

**Additional Notes:**

When the registration button has been clicked, clear the message label (don’t want error messages hanging around when new patients are registered).

The form should be rest back to its original state when the clear or cancel button are clicked. Clear the message label, clear the textboxes, reset the radio buttons, and don’t forget to reset the focus of the curser back to the first text field.

The exit button should just close the form. No additional message or anything should occur once this is clicked.

Note: For this assignment you must submit all project files to DC Connect (Java code) and output.

Style Guide Notes

To be eligible for full marks on this assignment your application must conform to the requirements as outlined above as well as the Google style guide as well as appropriate and complete program documentation. Include Java documentation for all the methods.