Program Cover Sheet

|  |
| --- |
| Name: Nathan VanSnepson |
| Assignment: Assignment 1 |
| List any parts of the assignment that do not work/were not completed: (None) |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Imports System.IO

Public Class frmName

'------------------------------------------------------------

'- File Name : frmMain.frm -

'- Part of Project: PayrollSystem -

'------------------------------------------------------------

' Written By: Nathan VanSnepson -

' Written On: January 16, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- -

'- This file contains the main form for the entire applic- -

'- ation. All user input is gathered on this form. The cal- -

'- culations for the program is done on this form. The out- -

'- is contained in this file. The file outputs to the user -

'- and saves to a text file. -

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- The purpose of this program is to calculate wages of em- -

'- ployees for a company. It allows the user to add new em- -

'- ployees. The program will output the calculated wage. -

'- Employee information will be saved to a text file and -

'- and will load in when the program is loaded. -

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- intEmployeeAccessed - The Position in the udtEmployeeList-

'- where the currently displayed employee is located. -

'- udtEmployeeList - ArrayList containing Employee informa- -

'- tion

'------------------------------------------------------------

'---------------------------------------------------------------------------------------

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'---------------------------------------------------------------------------------------

Const MI\_OVERTIME = 1.5 'State of Michigan overtime rate

'-------------------------------------------------------------------------------------------

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'-------------------------------------------------------------------------------------------

Structure udtEmployee

Dim strName As String 'The name of the Employee

Dim strID As String 'The ID of the Employee

Dim intHours As Integer 'The Number of hours the employee worked

Dim dblWage As Double 'The hourly wage of the employee

Dim blnSalaried As Boolean 'True if the Employee is salaried, false if they are hourly

End Structure

'---------------------------------------------------------------------------------------

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'---------------------------------------------------------------------------------------

Dim intEmployeeAccessed As Integer = 0 'The employees position in the array

Dim udtEmployeeList As ArrayList 'Array List of Employees

'-----------------------------------------------------------------------------------

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'-----------------------------------------------------------------------------------

Private Sub frmMain\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'------------------------------------------------------------

'- Subprogram Name: frmMain\_Load -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 17, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to read from an emplo- -

'- yee text file and display existing Employees. If no emp- -

'- oyees exist then it displays for the user to add a new -

'- employee. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Creates a new instance of the ArrayList

udtEmployeeList = New ArrayList

'Reads employees from file

readEmployeeFile()

'If there are no employees in the system

If udtEmployeeList.Count = 0 Then

'Set configuration of the application to add an employee

addEmployeeBtnChange()

Else

'Set the application to display the first employee

intEmployeeAccessed = 1

'Displays the employee

displayEmployee()

'Sets the application configuration to display employes

showEmployeeBtnChange()

End If

End Sub

Private Sub btnAddNewEmployee\_Click(sender As Object, e As EventArgs) Handles btnAddNewEmployee.Click

'------------------------------------------------------------

'- Subprogram Name: btnAddNewEmployee\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to setup the applicat- -

'- ion to allow the user to add an employee. It sets the -

'- text header of the foem. It also sets alltexts boxes to -

'- a blank string, and defaults the salaried checkbox to -

'- false. It then calls addEmployeeBtnChange() to hide the -

'- arrow buttons and the add employee button. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Sets text header of Form

Text = "Payroll System - Add New Employee"

'Sets textboxes to blank strings

txtName.Text = ""

txtID.Text = ""

txtHours.Text = ""

txtWage.Text = ""

'Sets salaried position to false

chkSalariedPosition.Checked = False

'Sets the application to the add employee configuration

addEmployeeBtnChange()

End Sub

Private Sub btnCancel\_Click(sender As Object, e As EventArgs) Handles btnCancel.Click

'------------------------------------------------------------

'- Subprogram Name: btnCancel\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to cancel out of -

'- creating a new employee if there are employees already -

'- entered -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'If there are no employees already in the system

If udtEmployeeList.Count = 0 Then

'Display message telling user they must add an employee

MessageBox.Show("No employees exist. Add an employee!")

Else

'Disply to user that they cancelled adding an employee

MessageBox.Show("Canceled adding an Employee")

'Displays employee information into the textboxes and checkbox

displayEmployee()

'Updates the application configuration

showEmployeeBtnChange()

End If

End Sub

Private Sub btnLeft\_Click(sender As Object, e As EventArgs) Handles btnLeft.Click

'------------------------------------------------------------

'- Subprogram Name: btnLeft\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to move up the list of -

'- employees when the user hits the corresponding button. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'If the currently accessed employee is the first in the ArrayList

If intEmployeeAccessed = 1 Then

'Display to user they cannot go further up the list

MessageBox.Show("You cannot move past the first record!")

Else

'Sets the currently displayed user to be the next one up the list

intEmployeeAccessed -= 1

'Dislpays the current employee information

displayEmployee()

End If

End Sub

Private Sub btnRight\_Click(sender As Object, e As EventArgs) Handles btnRight.Click

'------------------------------------------------------------

'- Subprogram Name: btnRight\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to down up the list of -

'- employees when the user hits the corresponding button. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'If the currently accessed employee is the last in the ArrayList

If intEmployeeAccessed = udtEmployeeList.Count Then

'Display to user they cannot move past the last record

MessageBox.Show("You cannot move past the last record!")

Else

'Sets the currently displayed user to be the next one down the list

intEmployeeAccessed += 1

'Dislpays the current employee information

displayEmployee()

End If

End Sub

Private Sub btnCalculateWage\_Click(sender As Object, e As EventArgs) Handles btnCalculateWage.Click

'------------------------------------------------------------

'- Subprogram Name: btnCalculateWage\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is displayed the calcula- -

'- ted wage of the employee based on their wage, hours, and -

'- if they are salaried. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- dblTotal - the total amount the employee is owed -

'- dblWage - the wage of the employee -

'- intHours - the number of hours the employee worked -

'------------------------------------------------------------

Dim intHours As Integer 'the wage of the employee

Dim dblWage As Double 'the number of hours the employee worked

'If there is no text in the hours input box or it is not an integer

If txtHours.Text.Length = 0 Or Not Integer.TryParse(txtHours.Text, intHours) Then

'Display to user that they have invalid input

MessageBox.Show("Enter the number of hours as an integer! (e.g. 40)")

'If there is no text in the wage input box or it is not an integer

ElseIf txtWage.Text.Length = 0 Or Not Double.TryParse(txtWage.Text, dblWage) Then

'Display to user that they have invalid input

MessageBox.Show("Enter the wage as a Double! (e.g. 9.75)")

Else

Dim dblTotal As Double 'the total amount the employee is owed

'If the employee is salary

If chkSalariedPosition.Checked Then

'Total is wage \* 40 hours

dblTotal = dblWage \* 40

Else

'If the employee has overtime

If intHours > 40 Then

'Calculation for overtime

dblTotal = dblWage \* 40 + dblWage \* MI\_OVERTIME \* (intHours - 40)

Else

'Calculation for straight time 40 and under

dblTotal = dblWage \* intHours

End If

End If

'Display in application the employees wage

lblWage.Text = "Employee is due " & FormatCurrency(dblTotal)

End If

End Sub

Private Sub btnSave\_Click(sender As Object, e As EventArgs) Handles btnSave.Click

'------------------------------------------------------------

'- Subprogram Name: btnSave\_Click -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to save the employee -

'- information to an ArrayList and then save the ArrayList -

'- to a file. -

'- -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- dblWage - the wage of the employee -

'- intHours - the number of hours the employee worked -

'- udtTempEmployee - temporary instance of udtEmployee -

'- structure to add to the ArrayList -

'------------------------------------------------------------

Dim intHours As Integer

Dim dblWage As Double

'If the name textbox is empty

If txtName.Text.Length = 0 Then

'Display to user to enter a Name

MessageBox.Show("Enter the employees Name!")

'If the ID textbox is empty

ElseIf txtID.Text.Length = 0 Then

'Display to user to enter an ID

MessageBox.Show("Enter the employees ID!")

'If there is no text in the hours input box or it is not an integer

ElseIf txtHours.Text.Length = 0 Or Not Integer.TryParse(txtHours.Text, intHours) Then

'Display to user that they have invalid input

MessageBox.Show("Enter the number of hours as an integer! (e.g. 40)")

'If there is no text in the wage input box or it is not an integer

ElseIf txtWage.Text.Length = 0 Or Not Double.TryParse(txtWage.Text, dblWage) Then

'Display to user that they have invalid input

MessageBox.Show("Enter the wage as a Double! (e.g. 9.75)")

Else

'Temporary instance of udtEmployee structure

Dim udtTempEmployee As udtEmployee

'Sets the values in udtTempEmployee to their corresponding text or checkbox values

udtTempEmployee.strName = txtName.Text

udtTempEmployee.strID = txtID.Text

udtTempEmployee.intHours = txtHours.Text

udtTempEmployee.dblWage = txtWage.Text

udtTempEmployee.blnSalaried = chkSalariedPosition.Checked

'Adds udtTempEmployee to the employee ArrayList

udtEmployeeList.Add(udtTempEmployee)

'Writes the Employee ArrayList to file

writeEmployeeFile()

'Sets the current displayed Employee to be the added employee

intEmployeeAccessed = udtEmployeeList.Count

'Displays to user the employee was successfully added

MessageBox.Show("Employee was successfully added!")

'Displays the current Employee

displayEmployee()

'Updates the application Configuration to display correct buttons

showEmployeeBtnChange()

End If

End Sub

Private Sub addEmployeeBtnChange()

'------------------------------------------------------------

'- Subprogram Name: addEmployeeBtnChange -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to configure the appl- -

'- ication to allow user to add a new employee. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Hides and shows buttons based on configuring the appliction

'for user to add a new employee

btnLeft.Hide()

btnRight.Hide()

btnAddNewEmployee.Hide()

lblWage.Text = ""

btnSave.Show()

btnCancel.Show()

txtName.Enabled = True

txtID.Enabled = True

txtHours.Enabled = True

txtWage.Enabled = True

chkSalariedPosition.Enabled = True

End Sub

Private Sub showEmployeeBtnChange()

'------------------------------------------------------------

'- Subprogram Name: showEmployeeBtnChange -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to configure the appl- -

'- ication to show the user the current employee. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Hides and shows buttons based on configuring the appliction

'for user to view the employees

btnLeft.Show()

btnRight.Show()

btnAddNewEmployee.Show()

btnSave.Hide()

btnCancel.Hide()

txtName.Enabled = False

txtID.Enabled = False

txtHours.Enabled = False

txtWage.Enabled = False

chkSalariedPosition.Enabled = False

End Sub

Private Sub readEmployeeFile()

'------------------------------------------------------------

'- Subprogram Name: readEmployeeFile -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is read employee informa- -

'- tion from a file and save it to an ArrayList. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender - reference to the object that raised the event -

'- e - reference to EventArgs class that contains any addi- -

'- tional information about the event. -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- srEmployee - A StreamReader instance to read from file. -

'- strArrEmployeeInfo - Array split from line read in file. -

'- strEmployeeInput - A string of the currently read line -

'- from the file. -

'- udtTempEmployee - temporary instance of udtEmployee -

'- structure to add to the ArrayList -

'------------------------------------------------------------

'If the file exists

If File.Exists("employees.txt") Then

Dim srEmployee As New StreamReader("employees.txt") 'A StreamReader instance to read from file.

Dim strEmployeeInput As String ' A string of the currently read line from the file.

'Do while to read through the file

Do

'Reads the next line of the file

strEmployeeInput = srEmployee.ReadLine()

'If there is not nothing in the text

If strEmployeeInput <> Nothing Then

Dim strArrEmployeeInfo As String() 'Array split from line read in file.

'Splits the line into an array

strArrEmployeeInfo = strEmployeeInput.Split(New Char() {";"c})

'temporary instance of udtEmployee structure to add to the ArrayList

Dim udtTempEmployee As udtEmployee

'Sets the corresponding structure values

udtTempEmployee.strName = strArrEmployeeInfo(0)

udtTempEmployee.strID = strArrEmployeeInfo(1)

udtTempEmployee.intHours = strArrEmployeeInfo(2)

udtTempEmployee.dblWage = strArrEmployeeInfo(3)

udtTempEmployee.blnSalaried = strArrEmployeeInfo(4)

'Adds the employee to the ArrayList

udtEmployeeList.Add(udtTempEmployee)

End If

'End when there is nothing in strEmployeeInput

Loop While strEmployeeInput <> Nothing

'Close the StreamReader

srEmployee.Close()

End If

End Sub

Private Sub writeEmployeeFile()

'------------------------------------------------------------

'- Subprogram Name: writeEmployeeFile -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to write employee inf- -

'- ormation to a file. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- srEmployee - StreamWriter to write employees to file. -

'- strFileOutput - A string that contains an employee to -

'- write at each line. -

'- udtTempEmployee - temporary instance of udtEmployee -

'- structure to add to the ArrayList -

'------------------------------------------------------------

'StreamWriter to write employees to file.

Dim srEmployee As New StreamWriter("employees.txt")

'Loops through the ArrayList of employees

For i As Integer = 0 To udtEmployeeList.Count - 1 Step 1

'temporary instance of udtEmployee structure to add to the ArrayList

Dim udtTempEmployee As udtEmployee = udtEmployeeList(i)

'A string that contains an employee to write at each line

Dim strFileOutput = udtTempEmployee.strName & ";" &

udtTempEmployee.strID & ";" &

udtTempEmployee.intHours & ";" &

udtTempEmployee.dblWage & ";" &

udtTempEmployee.blnSalaried

'Writes line to Stream Writer

srEmployee.WriteLine(strFileOutput)

Next

'Flushed the Stream Writer to the file

srEmployee.Flush()

'Closes the file

srEmployee.Close()

End Sub

Private Sub displayEmployee()

'------------------------------------------------------------

'- Subprogram Name: displayEmployee -

'------------------------------------------------------------

'- Written By: Nathan VanSnepson -

'- Written On: January 18, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- The purpose of this subprogram is to display the current -

'- employee. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- (None) -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- udtTempEmployee - temporary instance of udtEmployee -

'- structure to add to the ArrayList -

'------------------------------------------------------------

'temporary instance of udtEmployee structure to add to the ArrayList

Dim udtTempEmployee As udtEmployee

'Sets the temporary employee to be the current employee from the array

udtTempEmployee = udtEmployeeList(intEmployeeAccessed - 1)

'Sets corresponding controls to the value of the current employee

txtName.Text = udtTempEmployee.strName

txtID.Text = udtTempEmployee.strID

txtHours.Text = udtTempEmployee.intHours

txtWage.Text = udtTempEmployee.dblWage

chkSalariedPosition.Checked = udtTempEmployee.blnSalaried

'Updates the text header at the top of the application

Text = "Payroll System - Employee " & intEmployeeAccessed & "/" & udtEmployeeList.Count

'Sets the calculated employee wage label to be an empty string

lblWage.Text = ""

End Sub

End Class