**Design Tools - SAT - Pseudocode**

**Form 1 - Membership**

**Begin**

**If** NewBtn is clicked **Then**

**Open** form 2 - SignUp

**Elseif** ExistBtn is clicked **Then**

**Open** form 3 - Login

**EndIf**

**End**

**--------------------------------------------------------------------**

**Form 2 - SignUp**

**Begin**

**Hide** HidePass

**Hide** HideConPass

**End**

**--------------------------------------------------------------**

**SignUpBtn\_Click()**

**Begin**

**Run** SignUp()

**Open** Form3 - Login

**End ---------------------------------------------------------------**

**SignUp()**

**Begin**

**These three variable should be public to work with the other methods**

vUserName <- user entered a username

vPassword <- user entered a password

vConPass <- user entered a confirmation password

**Run** SignUpValidation()

**Run** XmlCheck()

**End**

**--------------------------------------------------------------**

**SignUpValidation()**

**Begin**

**//Split validation into 3 methods instead of writing the three in the same method**

**Run** UsernameValidatio()

**End**

**---------------------------------------------------------------**

**UsernameValidation()**

**Begin**

**bool** Found <- false

**Bool** UNSpaces <- vUsername contains spaces

**Load** the Xml document

**For** (each node in the XML)check

If vLoginUN = vUsername in the XML Then

Found <- true

EndIf

**Endfor**

**If** UNTextBox is empty **Then**

**Display** please enter a username

**Elseif** the text in UNTextBox has spaces **Then**

**Display** spaces aren’t accepted, please retype your username without any spaces

**Elseif** the text in the UNTextBox = username in the XML **Then**

**Display** Username already taken, please choose a different one

**Else**

**Run** PasswordValidation()

**Endif**

**End**

**--------------------------------------------------------------**

**PasswordValidation()**

**Begin**

**If** PassTextBox is empty **Then**

**Display** please enter a password

**Elseif** the password is less than 6 or more than 8 chars long **Then**

**Display** Password should be 6 to 8 characters long

**Elseif** the text in PassTextBox has spaces **Then**

**Display** spaces aren’t accepted, please retype your password without any spaces

**Else**

**Run** ConPassValidation()

**EndIf**

**End**

**--------------------------------------------------------------**

**ConPassValidation()**

**If** ConPassTextBox is empty **Then**

**Display** please enter the confirmation password

**Elseif vPassword <> vConPass**

**Display** Passwords do NOT match

**Else**

**Open** Form 3 - Login

**EndIf**

**End**

**---------------------------------------------------------------**

**XmlCheck()**

**Begin**

**If** the xml file exists then

**Run** NewXML

**Else**

**Run** SaveData

**EndIf**

**End**

**--------------------------------------------------------------**

**NewXML()**

**Begin**

Create the new XML file

Format the XML

Create the XML structure

Close access to the XML

Load the XML

Create the declaration

Insert the XML

Save the XML

**End**

**--------------------------------------------------------------**

**SaveData()**

**Begin**

Load the XML file

Create the new xml elements

Get the entered value

Save the XML

**End**

**--------------------------------------------------------------**

**ShowPass\_Click**

**Begin**

PassTextBox.UseSystemPasswordChar <- false

**Show** HidePass

**Hide** ShowPass

**End**

**--------------------------------------------------------------**

**HidePass\_Click**

**Begin**

PassTextBox.UseSystemPasswordChar <- true

**Show** ShowPass

**Hide** HidePass

**End**

**--------------------------------------------------------------**

**ShowConPass\_Click**

**Begin**

PassConTextBox.UseSystemPasswordChar <- false

**Show** HideConPass

**Hide** ShowConPass

**End**

**--------------------------------------------------------------**

**ShowConPass\_Click**

**Begin**

PassConTextBox.UseSystemPasswordChar <- false

**Show** ShowConPass

**Hide** HideConPass

**End**

**-------------------------------------------------------------------**

**Form 3 - Login**

**Begin**

**Hide** HidePass

**End**

**--------------------------------------------------------------**

**LoginBtn\_Click()**

**Begin**

Login()

**End**

**---------------------------------------------------------------**

**Login()**

**Begin**

**These two variables should be public to work with the other methods**

vLoginUN <- username entered by the user

vLoginPass <- password entered by the user

LoginValidation()

CheckDetails()

**Open** Form4 - Selection

**End**

**---------------------------------------------------------------**

**LoginValidation()**

**Begin**

**//Split validation into 2 methods instead of writing the three in the same method**

**Run** UsernameValidatio()

**End**

**---------------------------------------------------------------**

**UsernameValidation()**

**Begin**

**If** UNTextBox is empty **Then**

**Display** please enter a username

**Elseif** the text in UNTextBox has spaces **Then**

**Display** spaces aren’t accepted, please retype your username without any spaces

**Else**

**Run** PasswordValidation()

**Endif**

**End**

**--------------------------------------------------------------**

**PasswordValidation()**

**Begin**

**If** PassTextBox is empty **Then**

**Display** please enter a password

**Elseif** the password is less than 6 or more than 8 chars long **Then**

**Display** Password should be 6 to 8 characters long

**Elseif** the text in PassTextBox has spaces **Then**

**Display** spaces aren’t accepted, please retype your password without any spaces

**Endif**

**End**

**---------------------------------------------------------------**

**CheckDetails()**

**Begin**

vUserNameFound <- false

vPasswordFound <- false

**Load** the XML file

**For** (each node in the XML)check

**If** vLoginUN = vUsername in the XML **Then**

vUserNameFound <- true

**Else**

**Display** Username not found

**EndIf**

**If** vLoginPass = vPassword **Then**

vPasswordFound <- true

**Else**

**Display** Password does not match

**EndIf**

**EndFor**

**Open** Form 4 - Start/Finish selection

**End**

**--------------------------------------------------------------**

**ShowPass\_Click**

**Begin**

PassTextBox.UseSystemPasswordChar <- false

**Show** HidePass

**Hide** ShowPass

**End**

**--------------------------------------------------------------**

**HidePass\_Click**

**Begin**

PassTextBox.UseSystemPasswordChar <- true

**Show** ShowPass

**Hide** HidePass

**End**

**--------------------------------------------------------------------**

**Form 4 - Start/Finish selection**

**Begin**

**If** StartBtn is clicked **Then**

**Open** form 5 - Start time

**Elseif** FinishBtn is clicked **Then**

**Open** form 6 - Finish time

**EndIf**

**End**

**--------------------------------------------------------------------**

**Form 5 - Start time**

**Display** current date in DateLabel

**SaveBtn\_Click()**

**Begin**

**Run** CollectData()

**End**

**--------------------------------------------------------------CCollectData()**

**Begin**

Create a variable called vStartChoice

Create a variable called vStartAuto

Create a variable called vStartManual

**If** AutoRB is checked **Then**

vStartChoice <- Auto

**Elseif** ManRB is checked **Then**

vStartChoice <- Manual

**EndIf**

**If** vStartChoice = Auto **Then**

vStartAuto <- current time

**Elseif** vStartChoice = Manual **Then**

vStartManual <- time chosen by the user

**EndIf**

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**SaveData()**

**Begin**

Load the XML file

Create the new xml elements

Get the entered value

Save the XML

**End**

**--------------------------------------------------------------**

**NextBtn\_Click()**

**Begin**

**Open** Form 6 - Finish time

**End**

**--------------------------------------------------------------------**

**Form 6 - Finish time**

**Display** current date in DateLabel

**SaveBtn\_Click()**

**Begin**

**Run** CollectData()

**End**

**--------------------------------------------------------------**

**CollectData()**

**Begin**

Create a variable called vFinishChoice

Create a variable called vFinishAuto

Create a variable called vFinishManual

**If** AutoRB is checked **Then**

vFinishChoice <- Auto

**Elseif** ManRB is checked **Then**

vFinishChoice <- Manual

**EndIf**

**If** vFinishChoice = Auto **Then**

vFinishAuto <- current time

**Elseif** vFinishChoice = Manual **Then**

vFinishManual <- time chosen by the user

**EndIf**

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**SaveData()**

**Begin**

Load the XML file

Create the new xml elements

Get the entered value

Save the XML

**End**

**--------------------------------------------------------------**

**NextBtn\_Click()**

**Begin**

**Open** Form 7 - Job Salary Calculator

**End**

**--------------------------------------------------------------------**

**Form 7 - Job Salary Calculator**

**CalcBtn\_Click()**

**Begin**

**Run** Controller()

**End**

**--------------------------------------------------------------**

**Controller()**

**Begin**

**Run** HoursValidation()

**End**

--------------------------------------------------------------

**HoursValidation()**

**Begin**

**If** HrsTextBox is empty **Then**

**Display** please enter your pay per hour before pressing calculate

**ElseIf** Text entered in HrsTextBox is not double **Then**

**Display** Please enter a numerical pay value

**ElseIf** vPPH < 0 **Then**

**Display** Please enter a valid Weight - no negative numbers

**Else**

**Run** ExistanceCheck()

**EndIF**

**End**

**--------------------------------------------------------------**

**ExistanceCheck()**

**Begin**

**If** the xml that stores the start time exists **AND** the xml file that stores the finish time exists **Then**

**Run** StartTimeExistanceValidation()

**ElseIf** the xml that stores the start time doesn’t exist **OR** the xml file that stores the finish time doesn’t exists **Then**

**Display** Opps, it seems like you have forgetten to enter a Start/Finish time

**Open** Selection Form

**EndIf**

**End**

--------------------------------------------------------------

**StartTimeExistanceValidation()**

**Begin**

**Load** the user’s start XML file

**Foreach** (Node in the XML file)

**If** date node == vDate **Then**

StartTime <- true

**EndIf**

**EndFor**

**If** StartFound == true **Then**

**Run** FinishTimeExistanceValidation()

**Else**

**Display** Opps, it seems like you have forgetten to enter a Start time

**Open** Start Form

**EndIf**

**End**

**--------------------------------------------------------------**

**CalcTimeDay()**

**Begin**

**Load** the XML file

**Get** StartTime & FinishTime

vWorkTimeDay <- FinishTime - StartTime

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**CalcTimeWeek()**

**Begin**

**Load** the XML file

**Get** the Calculated time

ArrWorkTimeWeek

i <- 0

vWorkTimeWeek <- 0

**Foreach** (Node in the XML file)

ArrWorkTimeWeek[i] ← Calculated time from XML file

i++

**EndFor**

**For** j = 0 to len(ArrWorkTimeWeek)

**If** ArrWorkTimeWeek[j] > 0 **then**

vWorkTimeWeek ← vWorkTimeWeek + ArrWorkTimeWeek[j]

**Endif**

**EndFor**

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**CalcTimeMonth()**

**Begin**

**Load** the XML file

**Get** the Calculated time

ArrWorkTimeMonth

k <- 0

vWorkTimeMonth <- 0

**Foreach** (Node in the XML file)

ArrWorkTimeMonth[k] ← work time per week from XML file

k++

**EndFor**

**For** l = 0 to len(ArrWorkTimeMonth)

**If** ArrWorkTimeMonth[l] > 0 **then**

vWorkTimeMonth ← vWorkTimeMonth + ArrWorkTimeMonth[l]

**Endif**

**EndFor**

**End**

**--------------------------------------------------------------**

**CalcPayDay()**

**Begin**

vPPH <- pay entered by user

**Load** the XML file

**Get** Work time per day

vPayDay <- WorkTime \* vPPH

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**CalcPayWeek()**

**Begin**

**Load** the XML file

**Get** the Calculated pay per week

ArrPayWeek

m <- 0

vPayWeek <- 0

**Foreach** (Node in the XML file)

ArrPayWeek[m] ← Calculated pay from XML file

m++

**EndFor**

**For** n = 0 to len(ArrPayWeek)

**If** ArrPayWeek[n] > 0 **then**

vPayWeek ← vPayWeek + ArrPayWeek[n]

**Endif**

**EndFor**

**Run** SelectionSort(ArrPayWeek, 7)

**Run** SaveData()

**End**

**--------------------------------------------------------------**

**CalcPayMonth()**

**Begin**

**Load** the XML file

**Get** the Calculated pay per week

ArrPayMonth

o <- 0

vPayMonth <- 0

**Foreach** (Node in the XML file)

ArrPayMonth[o] ← pay per week from XML file

o++

**EndFor**

**For** p = 0 to len(ArrPayMonth)

**If** ArrPayMonth[p] > 0 **then**

vPayMonth ← vPayWeek + ArrPayMonth[p]

**Endif**

**EndFor**

**End**

**--------------------------------------------------------------**

**DisplayData()**

**Begin**

Create a variable called vPayChoice

**If** DayRB is checked **Then**

vPayChoice <- Day

**Elseif** WeekRB is checked **Then**

vPayChoice <- Week

**Elseif** MonthRB is checked **Then**

vPayChoice <- Month

**EndIf**

**Get** vWorkTimeDay

**Get** vWorkTimeWeek

**Get** vWorkTimeMonth

**Get** vPayDay

**Get** vPayWeek

**Get** vPayMonth

**If** vPayChoice = Day **Then**

**Display** *you have worked* vWorkTimeDay *hours this day and your salary is* vPayDay

**ElseIf** vPayChoice = Week **Then**

**Display** *you have worked* vWorkTimeWeek *hours this week and your salary is* vPayWeek

**ElseIf** vPayChoice = Month **Then**

**Display** *you have worked* vWorkTimeMonth *hours this month and your salary is* vPayMonth

**EndIF**

**End**

**---------------------------------------------------------------**

**SelectionSort(**Arr, number or indix**)**

**Begin**

**Get** pay per day values from XML

min <- 0

temp <- 0

**For** i = 0 **To** len(Arr) - 1, inc i by 1

min <- i

**For** j = i + 1 **To** len(Arr) - 1, inc j by 1

**If** Arr(j) < Arr(min) **Then**

min ← j

**EndIf**

**EndFor**

**If** min != i **Then**

temp ← Arr(i)

Arr(i) ← Arr(min)

Arr(min) ← temp

**EndIf**

**EndFor**

**Display** the sorted pays with the hours worked and the date of the each day in the listbox

**End()**

**---------------------------------------------------------------**

**SaveData()**

**Begin**

Load the XML file

Create the new xml elements

Get the entered value

Save the XML

**End**

**--------------------------------------------------------------------**