

Chapter 5 System Implementation

System implementation is the process to build and test a system that meets user and business requirements and the design specification. The analyzed knowledge from functional and non-functional requirements in the chapter 3 will be transformed to program codes that will be tangible to the users. This chapter will discuss on the actual development environment of the system in terms of hardware and software being used, the structure of the system, coding and integration of subsystem to form the real system and example some of the most difficult portion of the system development.

5.1 Development Environment

Development environment of CTutorial4u is divided into hardware configuration and software configuration.

5.1.1 Hardware Configurations

Ctutorial4u would interface with the following hardware requirements as discussed in chapter 3:

- Pentium II 100 MHz processor or higher
- At least 2 GB of hard disk
- At least 32MB of RAM
- Other standard computer peripherals – monitor, keyboard, mouse printer and CD-ROM.

5.1.2 Software Configurations

The system remains to use the following software types to develop the system and there are some minor changes in the software being used to develop this system.

Table 5.1: CTutorial4u Software Configuration

Software Type	Required Software	Remarks
Operating System	Windows 98	no changes to the OS.
Database	MS Access 2000	- MS Access 2000 remains to be the database tool but after saving the data it needs to be converted to prior version which is Access 97 to be able to communicate with VB 6.0 interface.
Software Development Tool	Visual Basic 6.0 (VB6) Adobe Photoshop 7.0 MS Image Composer MS Office 2000 Ms Project MS Visual C++ 6.0	No-changes to the software development tool. Added Ms Project 2000 as a tool to develop the project schedule.

5.2 Project Development

Project development started with literature review, requirement analysis, system design and output from this phases made the developer to proceed to next stage of development phase which is the implementation using the tools analyzed in the previous phases. The right tool is the most essential part of the development phase. There was some problem when accessing and reading the data from Microsoft Access 2000 from Visual Basic 6.0 interface. So, it is needed to be changed from Ms Access 2000 to Ms Access 97. After changing to prior version, the interface works perfectly to call the data from the database.

5.2.1 Data Preparation

Data required to develop this system is collected through questionnaire from the students and the requirements is listed in chapter 3 (System Analysis). The data collected from questionnaire is only the functionalities of the system. The most

important part of this system is the tutorial session, so the essential requirement is to get the if-then rules right. Apart from this data, suitable still images and graphics to be included in the system were collected by browsing and surfing the internet. Data required is important to be in hand before starting to build a system to avoid delay in the deliverables.

5.2.1.1 Still Images and animated graphics

Most of the images are related to education and computer since CTutorial4u is a educational package.

5.2.1.2 Database preparation

Based on the data collected from questionnaire and similar projects, essential data are listed and Microsoft Access 2000 is used to create the database tables which will be called in the VB codes to display the data. (Table 2.2, 2.3 to 2.8) in chapter 2 were converted into database tables in MS Access 2000 as if-then rules tables for arithmetic, control structures, functions, arrays, pointers and file processing chapters in tutorial session. Since, it is a client-based system and standalone so Microsoft Access 2000 is sufficient enough to cater the system.

5.2.1.3 Input form design

Prototypes with minimum coding of the interface were designed to get the feedbacks from the users. The prototypes then shown to selected students and colleagues to get comments on the design. Their comments is considered when implementing the real system and started with coding phases.

5.2.1.4 User interface design

User interface developed based on the comments and tested with the same students to verify their satisfaction. The 'Ten Usability Heuristics' by Jacob Nielsen [8] was taken as the guideline to design the user interfaces. Based on the functional and non-functional requirement stated in chapter 3, the system interfaces were developed. All the screens are clearly depicted in Appendix A, user manual.

5.2.2 Coding

Coding is the process of transforming the program logic into written codes without any errors. Coding phase is where design phase is translated into computer programming codes. Firstly, the pseudo code will be written in paper to check the logic and if its logic sounds workable then it will be coded using Visual Basic 6.0. Microsoft Visual Basic 6.0 It is easy to be learned and deployed by novice users. The coding phase is tedious and need to be done carefully because it plays a very important role in providing effective input to the information system using techniques of good screen design. The example of coding in tutorial session for the topic of arithmetic is shown in Figure 5.1, control structure in Figure 5.2 and function in Figure 5.3.

```

Sub Arithmetic()
Dim Response As String
Dim s As New Stream
Dim str
Dim fso As New FileSystemObject, txtfile, _
    fill As File, ts As TextStream
Set fill = fso.GetFile(App.Path & "\" & "Arithmetic.txt")
Set ts = fill.OpenAsTextStream(ForReading)
List1.clear
While (Not ts.AtEndOfStream)
str = ts.ReadLine
If str = "main()" Then
    List1.AddItem "main()"
    If rdoInteger.Value Then
        List1.AddItem "int a, b, ans;"
    ElseIf rdofloating.Value Then
        List1.AddItem "float a, b, ans;"
    ElseIf rdoDouble.Value Then
        List1.AddItem "double a, b, ans;"
    End If
    List1.AddItem "a = " & txtVarA.Text & ";"
    List1.AddItem "b = " & txtVarB.Text & ";"
    ArithType = cboArith.ListIndex
    If ArithType = 0 Then
        List1.AddItem "ans = " & "a" & "*" & "b;"
    ElseIf ArithType = 1 Then
        List1.AddItem "ans = " & "a" & "/" & "b;"
    ElseIf ArithType = 2 Then
        List1.AddItem "ans = " & "a" & "+" & "b;"
    ElseIf ArithType = 3 Then
        List1.AddItem "ans = " & "a" & "-" & "b;"
    ElseIf ArithType = 4 Then
        List1.AddItem "ans = " & "(a" & "+" & "b)" & "/2;"
    End If
    List1.AddItem "printf(""Here is the answer...\n" & """)"
    If rdoInteger.Value Then
        List1.AddItem "printf(""%d\n",ans);"
    ElseIf rdofloating.Value Then
        List1.AddItem "printf(""%f\n",ans);"
    ElseIf rdoDouble.Value Then
        List1.AddItem "printf(""%lf\n",ans);"
    End If
    Else
- List1.AddItem str
End If
Wend

```

Figure 5.1: Abstract of 'Arithmetic' Algorithm in the Tutorial session

```

Sub CtrlStruct()
Dim s As New Stream
Dim str
Dim str1
Dim str2
Dim fso As New FileSystemObject, txtfile, _
    fil1 As File, fil2 As File, fil3 As File, ts As TextStream, ts2 As TextStream, ts3 As
TextStream
    List4.clear
    Set fil1 = fso.GetFile(App.Path & "\" & "if.txt")
    Set fil2 = fso.GetFile(App.Path & "\" & "switchstruc.txt")
    Set fil3 = fso.GetFile(App.Path & "\" & "ifelse.txt")

    Set ts = fil1.OpenAsTextStream(ForReading)
    Set ts2 = fil2.OpenAsTextStream(ForReading)
    Set ts3 = fil3.OpenAsTextStream(ForReading)
    List4.clear

    If cboselection.ListIndex = 0 Then
        While (Not ts.AtEndOfStream)
            str = ts.ReadLine
            List4.AddItem str
        Wend

    ElseIf cboselection.ListIndex = 1 Then
        While (Not ts3.AtEndOfStream)
            str2 = ts3.ReadLine
            List4.AddItem str2
        Wend

    ElseIf cboselection.ListIndex = 2 Then

        While (Not ts2.AtEndOfStream)
            str1 = ts2.ReadLine
            List4.AddItem str1

        Wend

    End If

    ts.Close
    ts2.Close
    ts3.Close
End Sub

```

Figure 5.2: Abstract of ‘Control structure’ Algorithm in the Tutorial session

```

Sub Function()
Dim s As New Stream
Dim str
Dim fso As New FileSystemObject, txtfile, _
    fill As File, ts As TextStream
Set fill = fso.GetFile("C:\Tutorial4u\function.txt")
Set ts = fill.OpenAsTextStream(ForReading)
List2.clear
While (Not ts.AtEndOfStream)
    str = ts.ReadLine

    If str = "int main()" Then

        List2.AddItem "int " + cbomaxmin.Text + "(int , int, int)"
        List2.AddItem "int main (){"
        List2.AddItem "int a, b, c;"
        List2.AddItem "printf( " + """"Enter integers: """)"
        List2.AddItem "scanf( " + """"%d" + "%d" + "%d" + """" , &a, &b &c);"
        List2.AddItem "printf( """" + cbomaxmin.Text + "is: " + "%d\n"""" + ", " +
cbomaxmin.Text + "( a, b, c) );"
        List2.AddItem "int(int x, int y, int z){ "
        List2.AddItem "int ans = w;"
        List2.AddItem "if ( x" + txtFuncSym.Text + "ans )"
        List2.AddItem "ans = x"
        List2.AddItem "if ( y" + txtFuncSym.Text + "ans )"
        List2.AddItem "ans = y"
        List2.AddItem "if ( z" + txtFuncSym.Text + "ans )"
        List2.AddItem "ans = z"

    Else
        List2.AddItem str
    End If

Wend

ts.Close

End Sub

```

Figure 5.3: Abstract of 'Function' Algorithm in the Tutorial session

5.2.2.1 Database connection

The database to be connected is globally defined in the module as depicted in the Figure 5.4. The database is defined in the Data Sources (Open Database Connectivity ODBC) at the System DSN as CTutorial which the location is at C:\CTutorial4u\C1.mdb. So, every time to open the connection to the database just need to call the function conn_open and conn_close to close the connection. This will ease the developer's programming task which does not require the developer to define in every form.

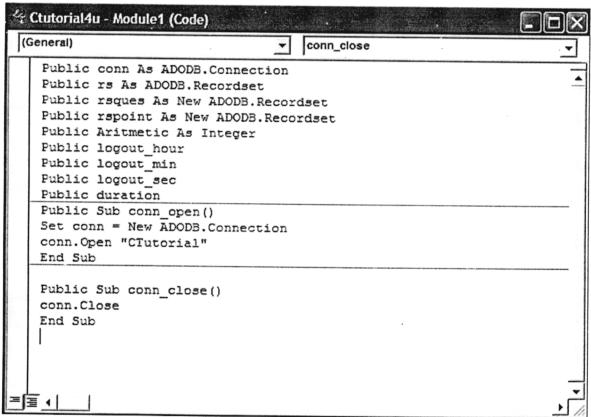


Figure 5.4: Module page

5.2.2.2 Authenticate member

Figure 4.10 shows the login form in the page 121 where this is the page which authenticates the users to access the system. If either the username or password supplied by the user does not match with those records in the database, the system will display a warning message that incorrect logon and as the remedy or help to recover

from the problem system the message will require the user to enter correct username and password. The users who are not authorized need to contact the administrator to add them as the authorized user.

5.2.2.3 Process with Database

The Figure 5.5 is the abstract of command add question in frmQuestion form where the administrator add the question to arithmetic table after choosing arithmetic chapter in combo box cboQuizchap. Firstly, conn_open is called to connect the C1 database and select statement is used to connect the arithmetic table and reques is the recordset defined globally in the module.

Example of VB6.0 code:

```
rsques.Open "select * from Arithmetic where QID = " & txtQID & "'", conn,  
adOpenDynamic, adLockOptimistic
```

```

Private Sub CmdAdd_Click()
conn_open

If cboQuizchap.ListIndex = 0 Then

rsques.Open "select * from Arithmetic where QID = '" & txtQID & "'", conn,
adOpenDynamic, adLockOptimistic
'rsAdmin.Open "select * from User where Username = '" & Username1 & "'", conn,
adOpenDynamic, adLockOptimistic
    If rsques.EOF = True Then
        rsques.AddNew
        rsques!QID = txtQID.Text
        rsques!Que = txtQuestion.Text
        rsques!A = txtA.Text
        rsques!B = txtB.Text
        rsques!C = txtC.Text
        rsques!D = txtD.Text
        rsques!ans = txtanswer.Text
        rsques.Update
        'rsAdmin!UserName = Username1.Text
        'rsAdmin!Password = Password1.Text
        'rsAdmin.Update
    End If

    rsques.MoveNext
    'rsAdmin.MoveNext

MsgBox "Question Successfully Added..."
txtQID.Text = ""
txtQuestion.Text = ""
txtA.Text = ""
txtB.Text = ""
txtC.Text = ""
txtD.Text = ""
txtanswer.Text = ""
rsques.Close
End If
    conn_close
End Sub

```

Figure 5.5: Abstract of 'Add New Question' in Administration session

```

(General) rndmz
frmgrid.Show vbModal 'Display the frmgrid only
End Sub

Public Sub rndmz()
Dim rndnum As Integer 'To store the random number value
Dim cho As Integer 'To store the choice value from the obtained total number records
Visible 'Call the function visible
    Data1.Recordset.MoveFirst 'Move the record pointer to first record
    cho = Data1.Recordset.RecordCount 'Count the total number of records in the
    cho = cho - 1 'Subtract 1 from the total number of records
    Randomize 'This is a library function which is used to randomize
    rndnum = Int((cho * Rnd) + 1) 'Obtain the random number and store it in the rnd
    Data1.Recordset.Move rndnum 'Move the record pointer to the random numbers
    If cnt = Data1.Recordset.RecordCount Then
MsgBox "Sorry Only " & frmQuizArith.Data1.Recordset.RecordCount & " Questions Available
Unload Me
Load frmgrid
frmgrid.Show vbModal
End.If
End Sub

Public Sub random()
cnt = cnt + 1 'Count the number of questions
lblQueno.Caption = cnt + 1 'To display the number of current question
If cnt < Val(lbl5.Caption) Then 'To check whether the question is less than user cho:
rndmz 'Call the rndmz sub program
Else
Visible 'call the visible sub program
lbl5.Visible = False 'Make the lbl5

```

Figure 5.6: Abstract of Rndmz function in the Quiz session

The function rndmz shown in the figure 5.6 uses data control which defines the database name for Data1 to be C:\CTutorial4u\C1.mdb and record source to be Arithmetic table. Once connected to the database, it will move to the first record and count total number of records. Then using the random function (Rnd) the function randomize and store the random number in memory and display the next question as the question of random number. This is how the data control is being used as database connection.

5.2.3 System Integration

System integration is the process of combining the all the subsystems together to form a complete system. Once all the pages are designed then connection of one page to another is done by using command buttons and navigational buttons.