1.0 Application Plan

1.1 The Goal

To develop an application that users can use to calculate the Net Present Value (NPV) and Internal Rate of Return (IRR).

1.2 Results Outcome

- Net Present Value
- Internal Rate of Return

1.3 Information Needed

- All basic assumptions entered by users

1.4 Calculation Performed

- Costs = Materials + Overhead + Labor
- Return = Sales - Costs
- Investments = Land + Building
- NPV and IRR = Please refer to the coding

2.0 Application Development

2.1 Introduction

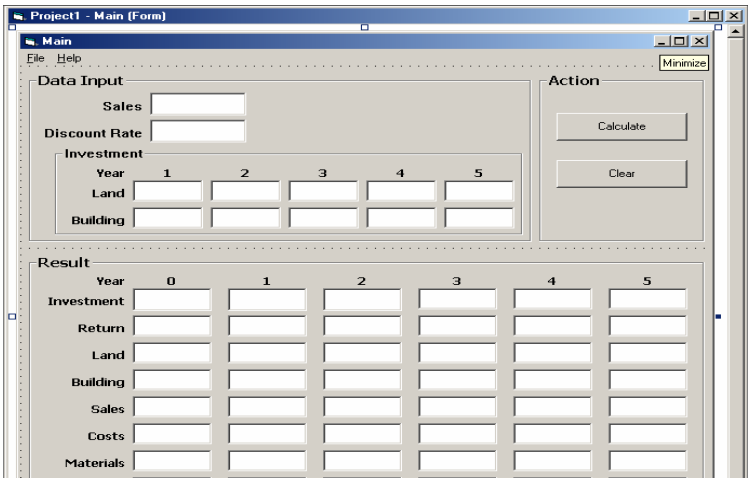
This application has been fully developed using Microsoft Visual Basic 6.0. Myself have written the coding and some was generated automatically by the program. The coding should be easy to understand, as guide has been included along with the coding.

2.2 Coding

There are three forms and one module inside this application.

Forms

Three forms have been created in this application are as followed. The first form is the main



Visual Basic Coding

```
'*****************************************************************
**************
'
' Module        : Main
' FileName      : Decision Support Systems Assignment 2 Part B
' Author        : Alvin Han Jiunn Kwang
' ID Number     : 300-0305-16
' Programme     : OBUAF 5
' Date Created  : 7/22/2001 10:12:22 PM
'
```

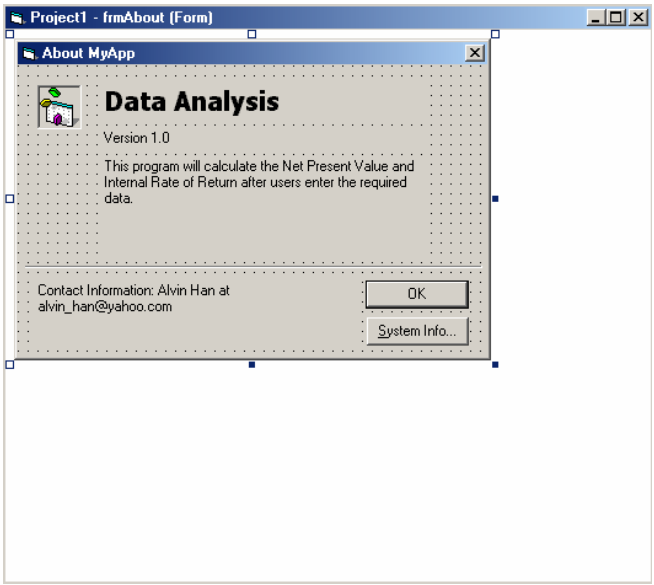
```
'
' Change History :
' 1.0       Thursday, August 02, 2001
'           Alvin Han Jiunn Kwang
'
'
'*****************************************************************
**************
Dim cf As Double

Private Sub Calculate_Click()

   On Error GoTo vbErrorHandler

'Sales
   Sales(1) = Text1
   Sales(2) = Sales(1) + 100000
   Sales(3) = Sales(2) + 100000
   Sales(4) = Sales(3) + 100000
   Sales(5) = Sales(4) + 100000

'Materials
   Materials(1) = 10000 + 0.2 * Sales(1)
   Materials(2) = 10000 + 0.2 * Sales(2)
   Materials(3) = 10000 + 0.2 * Sales(3)
   Materials(4) = 10000 + 0.2 * Sales(4)
   Materials(5) = 10000 + 0.2 * Sales(5)


'Overhead
   Overhead(1) = 0.1 * Sales(1)
   Overhead(2) = 0.1 * Sales(2)
   Overhead(3) = 0.1 * Sales(3)
   Overhead(4) = 0.1 * Sales(4)
   Overhead(5) = 0.1 * Sales(5)

'Labour
   Labour(1) = 20000 + 0.4 * Sales(1)
   Labour(2) = 20000 + 0.4 * Sales(2)
   Labour(3) = 20000 + 0.4 * Sales(3)
   Labour(4) = 20000 + 0.4 * Sales(4)
   Labour(5) = 20000 + 0.4 * Sales(5)

'Costs
   Costs(1) = Val(Materials(1)) + Val(Overhead(1)) + Val(Labour(1))
   Costs(2) = Val(Materials(2)) + Val(Overhead(2)) + Val(Labour(2))
```

```
    Costs(3) = Val(Materials(3)) + Val(Overhead(3)) + Val(Labour(3))
    Costs(4) = Val(Materials(4)) + Val(Overhead(4)) + Val(Labour(4))
    Costs(5) = Val(Materials(5)) + Val(Overhead(5)) + Val(Labour(5))

'Return
    R(0) = Val(Sales(0)) - Val(Costs(0))
    R(1) = Val(Sales(1)) - Val(Costs(1))
    R(2) = Val(Sales(2)) - Val(Costs(2))
    R(3) = Val(Sales(3)) - Val(Costs(3))
    R(4) = Val(Sales(4)) - Val(Costs(4))
    R(5) = Val(Sales(5)) - Val(Costs(5))

'Land
    Land(0) = Text3(0)
    Land(1) = Text3(1)
    Land(2) = Text3(2)
    Land(3) = Text3(3)
    Land(4) = Text3(4)

'Building
    Building(0) = Text5(0)
    Building(1) = Text5(1)
    Building(2) = Text5(2)
    Building(3) = Text5(3)
    Building(4) = Text5(4)

'Investment
    Investment(0) = Val(Land(0)) + Val(Building(0))
    Investment(1) = Val(Land(1)) + Val(Building(1))
    Investment(2) = Val(Land(2)) + Val(Building(2))
    Investment(3) = Val(Land(3)) + Val(Building(3))
    Investment(4) = Val(Land(4)) + Val(Building(4))


'NPV & IRR
    cf0 = Val(R(0)) - Val(Investment(0))
    cf1 = Val(R(1)) - Val(Investment(1))
    cf2 = Val(R(2)) - Val(Investment(2))
    cf3 = Val(R(3)) - Val(Investment(3))
    cf4 = Val(R(4)) - Val(Investment(4))
    cf5 = Val(R(5))

    ReDim valuearray(5) As Double

    valuearray(0) = cf0
    valuearray(1) = cf1
```

```
    valuearray(2) = cf2
    valuearray(3) = cf3
    valuearray(4) = cf4
    valuearray(5) = cf5

    Text2 = Text2 / 100
    discountrate = Text2
    guess = 0.01

    Net = NPV(discountrate, valuearray())
    ir = IRR(valuearray(), guess)

QuitRoutine:
    '
    ' Here is where we leave the routine
    '
    GoSub CleanUpRoutine
    Exit Sub

vbErrorHandler:
    '
    ' Here is where we should handle any errors
    '
    GoSub CleanUpRoutine
    MsgBox Err.Description & vbCrLf & vbCrLf & "Data Input Error, line : " &
Erl, vbCritical, App.ProductName
    Resume QuitRoutine
    Resume

CleanUpRoutine:
    '
    ' Here is where we should clean up any resources
    '
    Return

End Sub


Private Sub Clear_Click()
Dim ctr As Control
For Each ctrl In Me.Controls
    If TypeOf ctrl Is TextBox Then
        ctrl.Text = ""
    End If
Next
```

```
End Sub

Private Sub StatusBar1_PanelClick(ByVal Panel As MSComctlLib.Panel)

End Sub

Private Sub mnuFileExit_Click()
   Beep

   If MsgBox("Thank you for using this application!" & vbCrLf & vbCrLf & "Are
you sure you want to quit!", vbInformation + vbYesNo, "Exit") = vbYes Then
      Unload Me
   End If
End Sub

Private Sub mnuHelpAbout_Click()
   frmAbout.Show
End Sub
```

The second form is the About form.



Visual Basic Coding

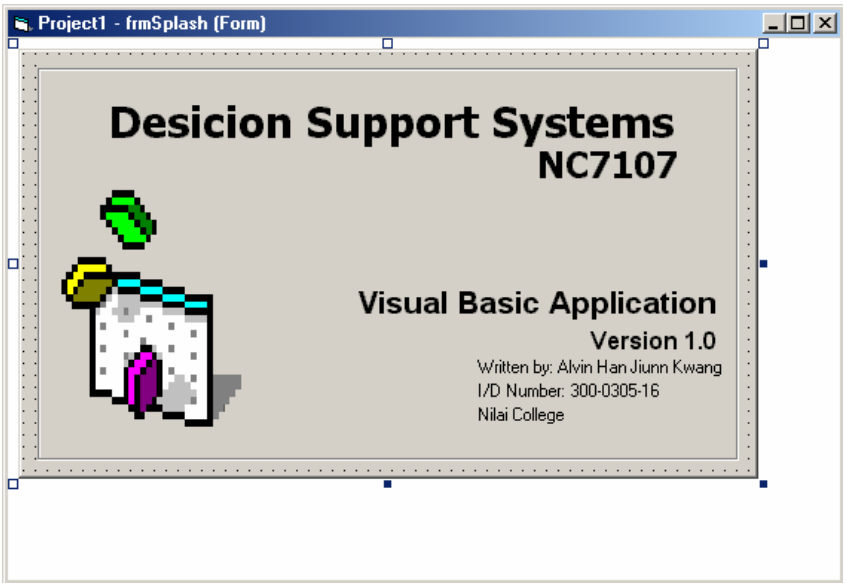
```
Option Explicit

' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE +
_
                KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS
+ _
                KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                    ' Unicode nul terminated string
Const REG_DWORD = 4                 ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools
Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
```

```
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As
Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String,
ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String,
ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long)
As Long


Private Sub cmdSysInfo_Click()
  Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
  Unload Me
End Sub

Private Sub Form_Load()
   Me.Caption = "About " & App.Title
   lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
   lblTitle.Caption = App.Title
End Sub

Public Sub StartSysInfo()
   On Error GoTo SysInfoErr

   Dim rc As Long
   Dim SysInfoPath As String

   ' Try To Get System Info Program Path\Name From Registry...
   If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
   ' Try To Get System Info Program Path Only From Registry...
   ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
      ' Validate Existance Of Known 32 Bit File Version
      If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
         SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

      ' Error - File Can Not Be Found...
```

```
      Else
         GoTo SysInfoErr
      End If
   ' Error - Registry Entry Can Not Be Found...
   Else
      GoTo SysInfoErr
   End If

   Call Shell(SysInfoPath, vbNormalFocus)

   Exit Sub
SysInfoErr:
   MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String,
SubKeyRef As String, ByRef KeyVal As String) As Boolean
   Dim i As Long                                ' Loop Counter
   Dim rc As Long                               ' Return Code
   Dim hKey As Long                             ' Handle To An Open Registry
Key
   Dim hDepth As Long                           '
   Dim KeyValType As Long                       ' Data Type Of A Registry Key
   Dim tmpVal As String                         ' Tempory Storage For A Registry
Key Value
   Dim KeyValSize As Long                       ' Size Of Registry Key Variable
   '------------------------------------------------------------
   ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
   '------------------------------------------------------------
   rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) '
Open Registry Key

   If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Error...

   tmpVal = String$(1024, 0)                    ' Allocate Variable Space
   KeyValSize = 1024                            ' Mark Variable Size

   '------------------------------------------------------------
   ' Retrieve Registry Key Value...
   '------------------------------------------------------------
   rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                  KeyValType, tmpVal, KeyValSize)    ' Get/Create Key Value

   If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Errors
```

```
   If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then          ' Win95 Adds Null
Terminated String...
      tmpVal = Left(tmpVal, KeyValSize - 1)              ' Null Found, Extract From
String
   Else                                                  ' WinNT Does NOT Null Terminate
String...
      tmpVal = Left(tmpVal, KeyValSize)                  ' Null Not Found, Extract
String Only
   End If
   '------------------------------------------------------------
   ' Determine Key Value Type For Conversion...
   '------------------------------------------------------------
   Select Case KeyValType                                ' Search Data Types...
   Case REG_SZ                                           ' String Registry Key Data Type
      KeyVal = tmpVal                                    ' Copy String Value
   Case REG_DWORD                                        ' Double Word Registry Key
Data Type
      For i = Len(tmpVal) To 1 Step -1                   ' Convert Each Bit
         KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))   ' Build Value Char. By
Char.
      Next
      KeyVal = Format$("&h" + KeyVal)                    ' Convert Double Word To
String
   End Select

   GetKeyValue = True                                    ' Return Success
   rc = RegCloseKey(hKey)                                ' Close Registry Key
   Exit Function                                         ' Exit

GetKeyError:      ' Cleanup After An Error Has Occured...
   KeyVal = ""                                           ' Set Return Val To Empty String
   GetKeyValue = False                                   ' Return Failure
   rc = RegCloseKey(hKey)                                ' Close Registry Key
End Function
```

The third form is the Splash Screen.



Visual Basic Coding

```
Option Explicit

Private Sub Form_KeyPress(KeyAscii As Integer)
   Unload Me
End Sub

Private Sub Form_Load()
   lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision

End Sub

Private Sub Frame1_Click()
   Unload Me
End Sub
```

Modules

```
Public fMainForm As Main


Sub Main()
   frmSplash.Show
   frmSplash.Refresh
   Set fMainForm = New Main
   Load fMainForm
   Unload frmSplash


   fMainForm.Show
End Sub
```

3.0 Template Illustrations

3.1 Introduction

This Visual Basic application is done for the purpose to calculate the Net Present Value (NPV) and Internal Rate of Return (IRR). With the required data being input, it also able to calculate the sales, costs and return in the upcoming five years.

3.2 Starting the Application

Whenever this application is executed, users will be first greeted by a Splash Screen.

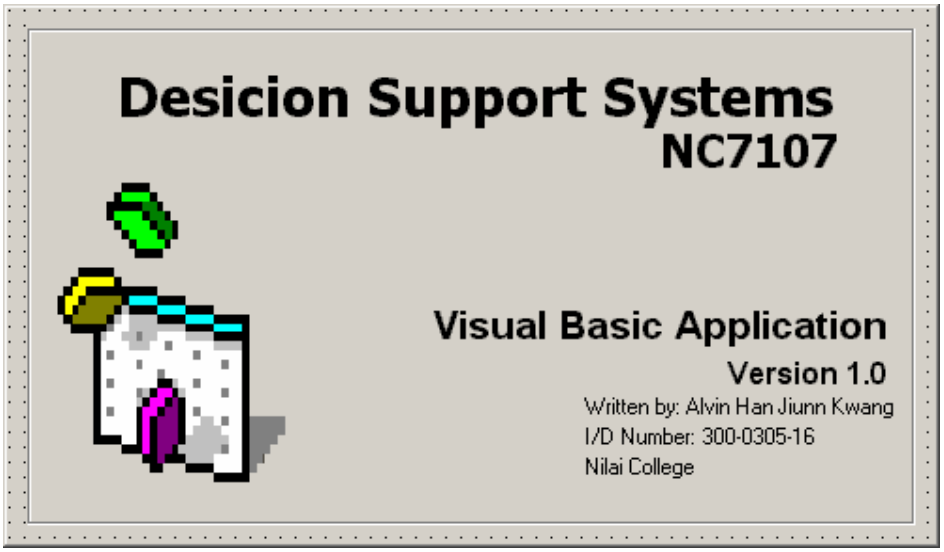


Figure 3.1: Splash Screen

This splash screen contains some general information about the application, such as the name of the application, version of the application, the author and so on. This splash screen will only appear for a few seconds being unloaded automatically.

3.3 Main Page

After the splash screen disappear by it self, the main page of this application will be loaded. This is the only important page in the whole application. This page divided into three main parts. There are the Data Input, Action and Result.

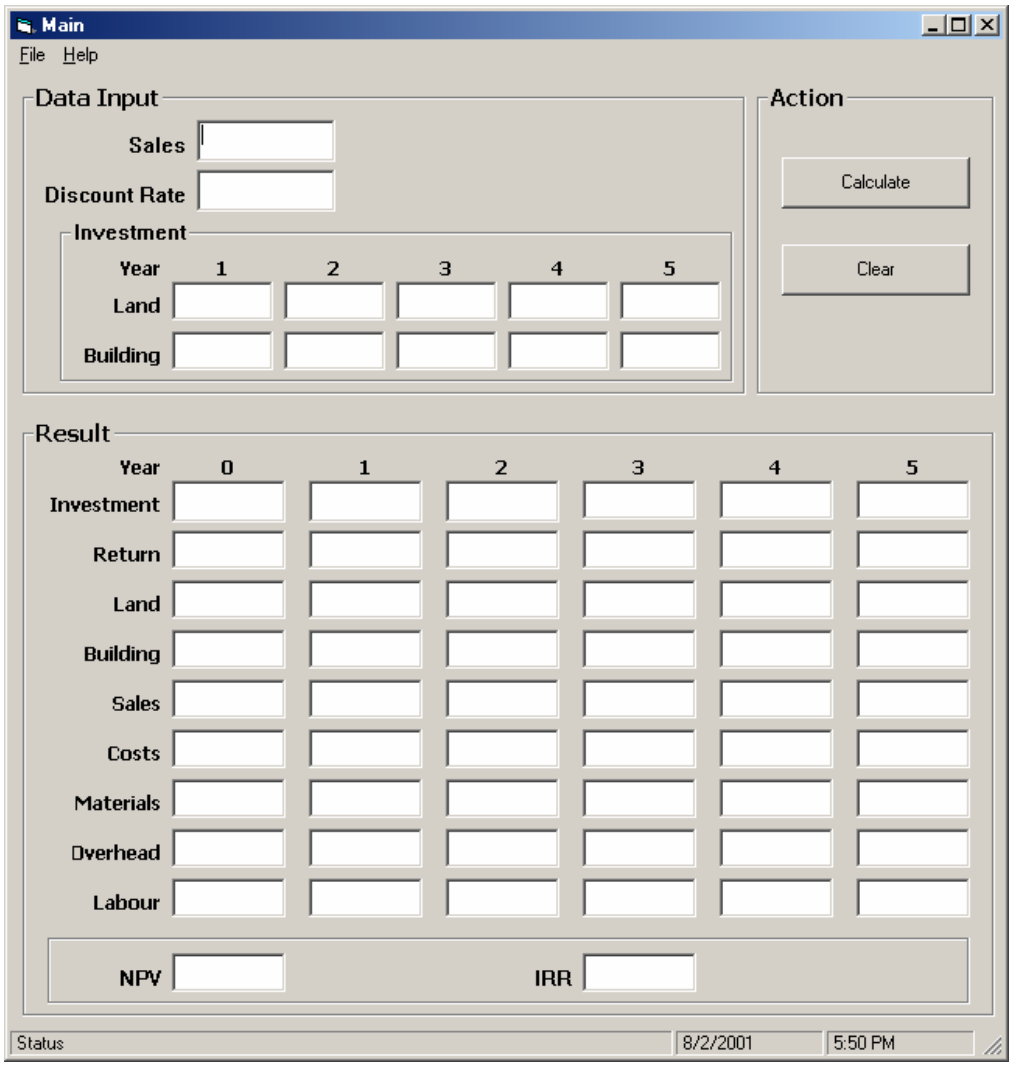


Figure 3.2: Main Page

Data Input

The data input part is for the users to input their data. There are only three main data users need to enter, which are Sales, Discount Rate and Investments.

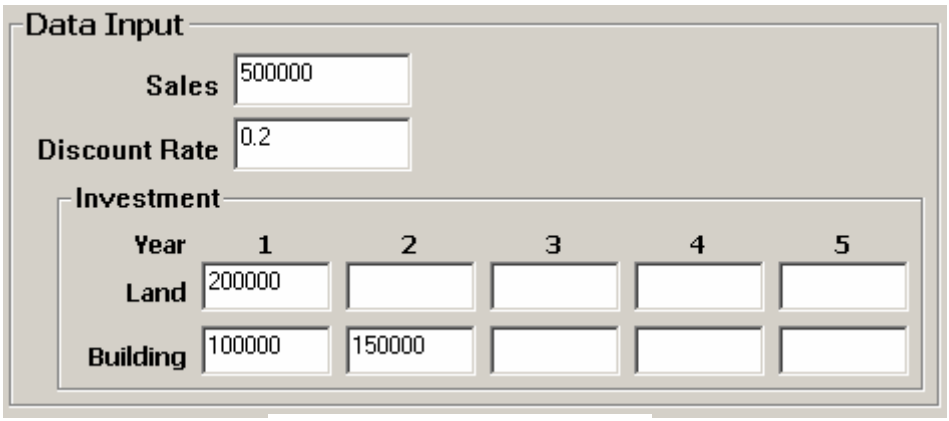


Figure 3.3: Data Input

Action

After the users have entered the appropriate data, the next things the users need to do is to click the Calculate button found in the Action area. Or else, users can click Clear button to clear all the data being input.

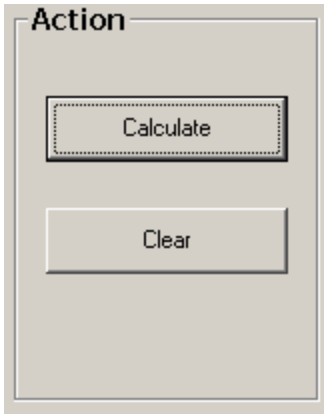


Figure 3.4: Action

Result

All of the analysed information is being displayed in the Result area. These include Investment, Return, Land, Building, Sales, Costs, Materials, Overhead and Labour for five years. Not to leave behind will be the Net Present Value

(NPV) and Internal Rate of Return (IRR), the most important output for this application.

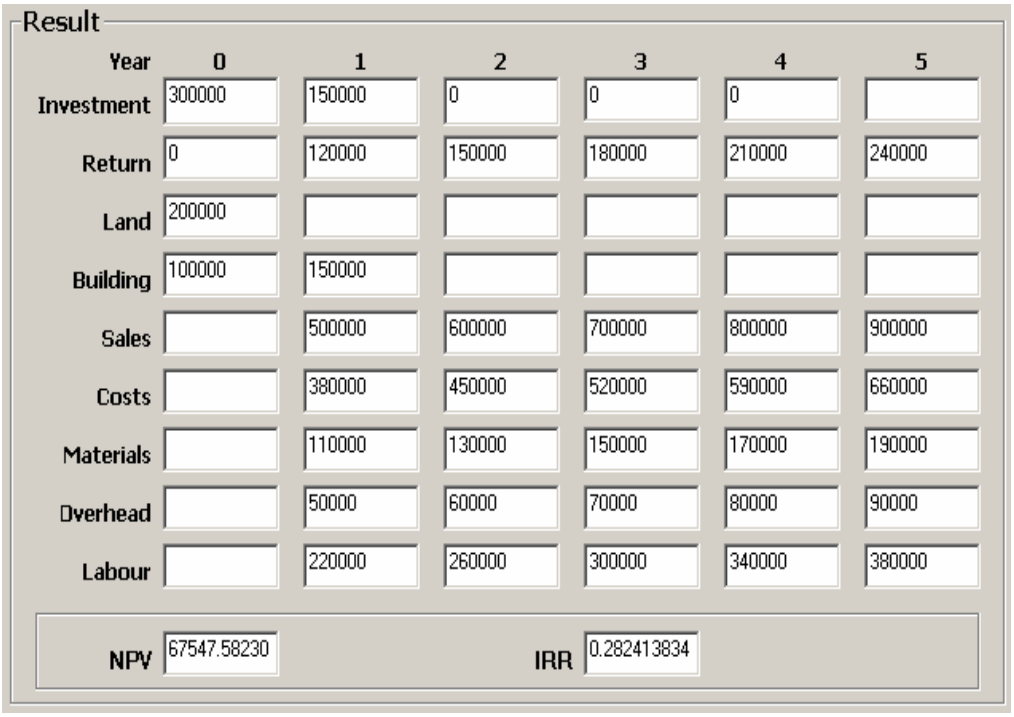


Figure 3.5: Result

Data Validation

For every data input, which is found to be invalid, the further calculation will be terminated and a warning message box will appear to alert the users about the Data Input Error. For example, users are not allowed to input alphabet characters in any of the boxes.

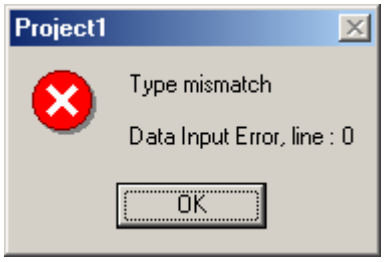


Figure 3.6: Data Input Not Acceptable

3.4 Additional Functions

There are two menu can be found on top of the main page, which is File and Help.

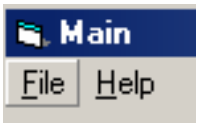


Figure 3. 7: Menu

Under File menu, there is an Exit menu. When clicked, a confirmation message box will appear as followed:

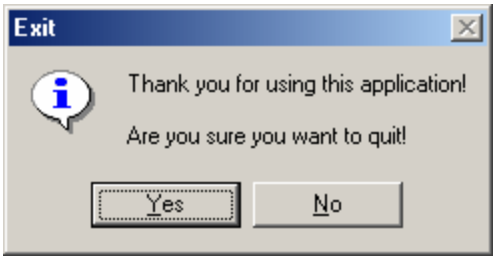


Figure 3.8: Exit

Under Help menu, there is an About menu, once clicked, a box will appear giving some information regarding to this application.

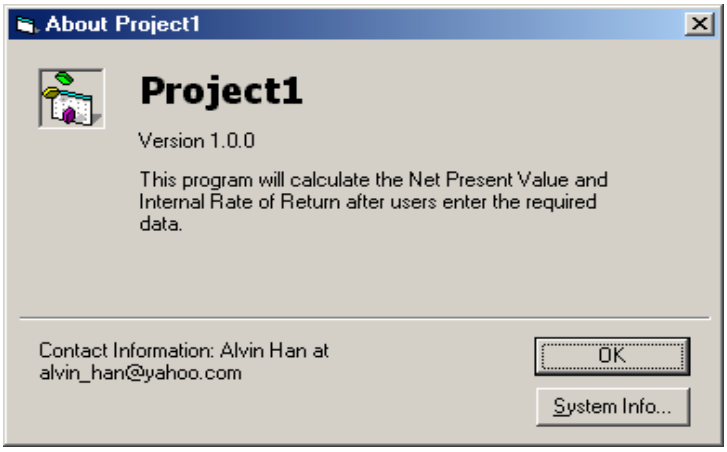


Figure 3.9: About

At the bottom of the main page, the status bar has been added to show the date and the time as shown.

Figure 3.10: Status Bar

Bibliography

Books

Marc Young, Jean Ross, (1998), *Advanced Microsoft Visual Basic 6.0*, Microsoft Press.

Francesco Balena (1999), *Programming Microsoft Visual Basic 6.0*, Microsoft Press.

Web Sites

Aivosto, (2001), *http://www.aivosto.com/vblinks.html* visited on 1 July 2001.

Erlandsendata, (2001), *http://www.erlandsendata.no/english/topics.htm/* visited on 1 July 2001.

JWalk & Associates, (2001), *http://j-walk.com/ss/excel/files/* visited on 1 July 2001.

JWalk & Associates, (2001), *http://www.j-walk.com/ss/excel/files/developer.htm* visited on 1 July 2001.

Micorsoft, (2001), *http://support.microsoft.com/support/kb/articles/Q110/8/88.asp* visited on 1 July 2001.

VB Code, (2001), *http://www.vbcode.com/default.htm* visited on 1 July 2001.