SCT221-c002-0030/2023
FELIX MUINDE
Applications Programming CAT

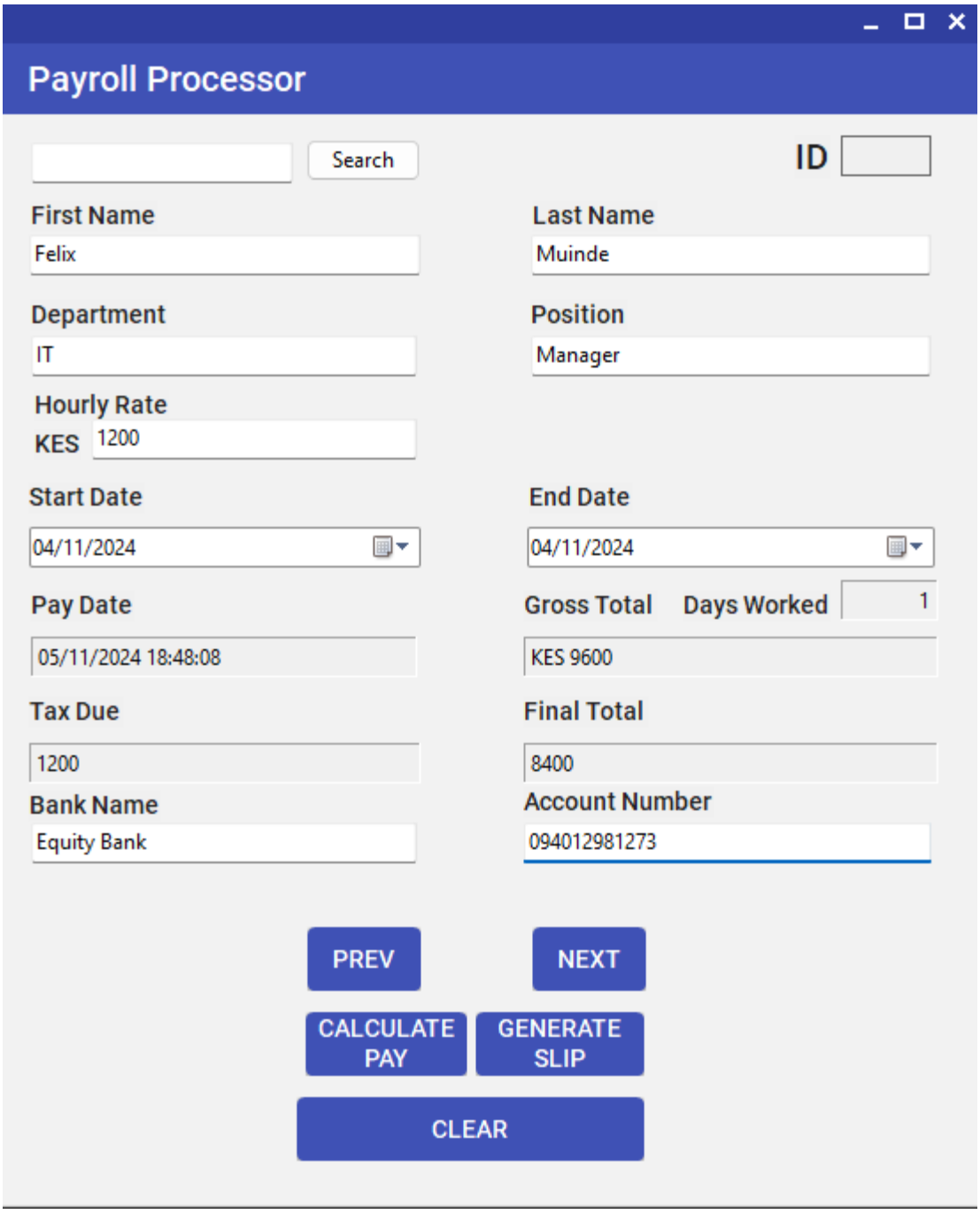

Figure 1 Screenshot of Application

The application is meant to simulate a payroll processing system and generate a payslip for employees stored in an Access Database file. While also allowing users to input custom records.

Source Code

```
Imports iTextSharp.text
Imports iTextSharp.text.pdf
Imports System.IO


Public Class frmPayroll
    Private connectionString = My.Settings.ConnectionString

    ' With these we can store employee data temporarily while reading from the DB
and navigate through records sequentially
    Private currentRecordIndex As Integer = 0
    Private employeeDataTable As DataTable
```

On loading the form, the first employee record is used to fill form and its various labels.

```
    Private Sub frmPayroll_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        LoadEmployeeRecords()
    End Sub
```

Employee Data Loading Functions

```
    Private Sub LoadEmployeeRecords()
        Using connection As New OleDbConnection(connectionString)
            Try
                connection.Open()
                Dim query As String = "SELECT Employees.*, Accounts.AccountNumber,
Accounts.BankName " &
                                      "FROM Employees " &
                                      "INNER JOIN Accounts ON Employees.EmployeeID =
Accounts.EmployeeID " &
                                      "WHERE Employees.Active = True " &
                                      "ORDER BY Employees.EmployeeID"
                Dim adapter As New OleDbDataAdapter(query, connection)
                employeeDataTable = New DataTable()
                adapter.Fill(employeeDataTable)

                If employeeDataTable.Rows.Count > 0 Then
                    currentRecordIndex = 0
                    PopulateEmployeeDetails(currentRecordIndex)
                Else
                    MessageBox.Show("No active records found.")
                End If
            Catch ex As Exception
                MessageBox.Show("Error loading data: " & ex.Message)
            End Try
        End Using
    End Sub

    Private Sub PopulateEmployeeDetails(index As Integer)
        If employeeDataTable IsNot Nothing AndAlso employeeDataTable.Rows.Count >
index AndAlso index >= 0 Then
```

```
            Dim row As DataRow = employeeDataTable.Rows(index)
            IDLabelText.Text = row("EmployeeID").ToString()
            lblFirstName.Text = row("FirstName").ToString()
            lblSecondName.Text = row("LastName").ToString()
            lblDepartment.Text = row("Department").ToString()
            lblPosition.Text = row("Position").ToString()
            lblRate.Text = Convert.ToDecimal(row("HourlyRate")).ToString()
            lblAccNo.Text = row("AccountNumber").ToString()
            lblBank.Text = row("BankName").ToString()
        End If
    End Sub
```

Clear fields functions

```
    Private Sub ClearDates()
        dtpStart.CustomFormat = ""
        dtpEnd.CustomFormat = ""
    End Sub

    Private Sub Clear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
                                                                                    
                                                                                    
                                                                                    
                                                                                    
                                                                                    
                                                                                    
    End Sub
```

Calculate Total Pay

```
    Private Sub CalcPay_Click(sender As Object, e As EventArgs) Handles
btnCalcPay.Click
        Dim startDate As Date = dtpStart.Value
        Dim endDate As Date = dtpEnd.Value
        Dim rate As Decimal = CInt(lblRate.Text)
        Dim difference As TimeSpan = (endDate - startDate)
        Dim totalDays As Integer = CInt(difference.TotalDays) + 1

        ' We assume they are working only 8hrs and not being paid overtime and a
flat Tax Rate of 12.5%
        Dim totalPay As Decimal = rate * totalDays * 8
        Dim tsxDue As Decimal = totalPay * 0.125
        Dim finalTotal As Decimal = totalPay - tsxDue

        ' Prevent time travel and silly values like negative working days
        If totalDays <= 0 Then
            MessageBox.Show($"Unless your employee is a time traveller, I don't see
how they could have achieved working  {totalDays} days. Check your math!")
        Else
            lblDaysWorked.Text = totalDays.ToString()
            lblGross.Text = "KES " + totalPay.ToString()
            lblPayDate.Text = endDate.AddDays(1).ToString()
            lblTax.Text = tsxDue.ToString()
            lblNetTotal.Text = finalTotal.ToString()
        End If
    End Sub
```

Move through records sequentially

```
    Private Sub PreviousRecord_Click(sender As Object, e As EventArgs) Handles
btnPreviousRecord.Click
        If currentRecordIndex > 0 Then
            currentRecordIndex -= 1
            PopulateEmployeeDetails(currentRecordIndex)
        Else
            MessageBox.Show("There are no more employees before this one.")
        End If
    End Sub


‘
Private Sub NextRecord_Click(sender As Object, e As EventArgs) Handles
btnNextRecord.Click
        If currentRecordIndex < employeeDataTable.Rows.Count - 1 Then
            currentRecordIndex += 1
            PopulateEmployeeDetails(currentRecordIndex)
        Else
            MessageBox.Show("Congratulations! You reached the end of the records.")
        End If
    End Sub
```

Search function the Access Database file

```
    Private Sub Search_Click(sender As Object, e As EventArgs) Handles
btnSearch.Click
        Dim searchID As String = txtSearchBox.Text.Trim()
        Dim employeeID As Integer

        If String.IsNullOrEmpty(searchID) OrElse Not Integer.TryParse(searchID,
employeeID) Then
            MessageBox.Show("Please enter a valid numeric Employee ID.")
            Return
        End If

        Using connection As New OleDbConnection(connectionString)
            Try
                connection.Open()
                Dim query As String = "SELECT Employees.*, Accounts.AccountNumber,
Accounts.BankName " &
                                      "FROM Employees " &
                                      "INNER JOIN Accounts ON Employees.EmployeeID =
Accounts.EmployeeID " &
                                      "WHERE Employees.EmployeeID = @EmployeeID"
                Dim command As New OleDbCommand(query, connection)
                command.Parameters.AddWithValue("@EmployeeID", employeeID)

                Using reader As OleDbDataReader = command.ExecuteReader()
                    If reader.Read() Then
                        IDLabelText.Text = reader("EmployeeID").ToString()
                        lblFirstName.Text = reader("FirstName").ToString()
                        lblSecondName.Text = reader("LastName").ToString()
                        lblDepartment.Text = reader("Department").ToString()
                        lblPosition.Text = reader("Position").ToString()
                        lblRate.Text =
Convert.ToDecimal(reader("HourlyRate")).ToString()
                        lblAccNo.Text = reader("AccountNumber").ToString()
```

```
                    lblBank.Text = reader("BankName").ToString()
                Else
                    MessageBox.Show("Employee ID not found.")
                End If
            End Using
        Catch ex As Exception
            MessageBox.Show("Error searching for employee: " & ex.Message)
        End Try
    End Using
End Sub
```

Function that outputs a pdf using itextsharp library

```
    Private Sub Play_Click(sender As Object, e As EventArgs) Handles btnPlay.Click
        Dim savePath As String =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Desktop),
$"{lblFirstName.Text}{lblSecondName.Text}-{Date.Now.ToString("yyyy-MM-
dd")}{Date.Now.ToString("ss")}.pdf")
        Dim pdfDoc As New Document(PageSize.A4, 40, 40, 40, 20)

        Try
            PdfWriter.GetInstance(pdfDoc, New FileStream(savePath, FileMode.Create))

            pdfDoc.Open()

            Dim titleFont As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 16,
iTextSharp.text.Font.BOLD)
            Dim regularFont As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 12,
iTextSharp.text.Font.NORMAL)
            'Title
            pdfDoc.Add(New Paragraph($"PaySlip: {lblFirstName.Text}
{lblSecondName.Text}", titleFont))
            pdfDoc.Add(New Paragraph(Environment.NewLine))
            'Body
            pdfDoc.Add(New Paragraph("Employee ID: " & IDLabelText.Text, regularFont))
            pdfDoc.Add(New Paragraph("First Name: " & lblFirstName.Text, regularFont))
            pdfDoc.Add(New Paragraph("Last Name: " & lblSecondName.Text, regularFont))
            pdfDoc.Add(New Paragraph("Department: " & lblDepartment.Text, regularFont))
            pdfDoc.Add(New Paragraph("Position: " & lblPosition.Text, regularFont))
            pdfDoc.Add(New Paragraph("Hourly Rate: " & lblRate.Text, regularFont))
            pdfDoc.Add(New Paragraph("Start Date: " & dtpStart.Value.ToString(),
regularFont))
            pdfDoc.Add(New Paragraph("End Date: " & dtpEnd.Value.ToString(),
regularFont))
            pdfDoc.Add(New Paragraph("Total Days Worked: " & lblDaysWorked.Text,
regularFont))
            pdfDoc.Add(New Paragraph("Gross Pay: " & lblGross.Text, regularFont))
            pdfDoc.Add(New Paragraph("Tax Due: " & lblTax.Text, regularFont))
            pdfDoc.Add(New Paragraph("Net Total " & lblNetTotal.Text, regularFont))
            pdfDoc.Add(New Paragraph("Account Number: " & lblAccNo.Text, regularFont))
            pdfDoc.Add(New Paragraph("Bank Name: " & lblBank.Text, regularFont))
            pdfDoc.Add(New Paragraph(Environment.NewLine))

            MessageBox.Show("PDF created successfully at: " & savePath)
        Catch ex As Exception
            MessageBox.Show("Error generating PDF: " & ex.Message)
        Finally
```

```
            pdfDoc.Close()
        End Try
    End Sub
End Class
```

Figure 2 Printout

GitHub Repo Link