using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using CrystalDecisions.CrystalReports.Engine;

using CrystalDecisions.ReportSource;

using CrystalDecisions.Shared;

namespace PensionVerificationSystem

{

public partial class FrmGeneralPensionEnrolReport : Form

{

public FrmGeneralPensionEnrolReport()

{

InitializeComponent();

}

private void FrmGeneralPensionEnrolReport\_Load(object sender, EventArgs e)

{

try

{

OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Application.StartupPath + "\\dbPension.accdb;Persist Security Info=True");

//OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Project 2016\PensionVerificationSystem\PensionVerificationSystem\dbPension.accdb;Persist Security Info=True");

OleDbDataAdapter adapter = new OleDbDataAdapter("select UniquePensionerId, PensionerFullName, DateOfBirth, PensionerGender, SalaryStructure, CurrentMonthlyPension, Ministry, NextOfKinName, NextOfKinPhoneNumber from tblPensionerEnrollment", cn);

DataSet ds = new DataSet();

adapter.Fill(ds, "tblPensionerEnrolment");

CrystalReportPensionEnrolment rpt = new CrystalReportPensionEnrolment();

rpt.SetDataSource(ds);

crystalReportViewer1.ReportSource = rpt;

}

catch (Exception er)

{

MessageBox.Show(er.Message);

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace PensionVerificationSystem

{

public partial class FrmChangePassword : Form

{

private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Application.StartupPath + "\\dbPension.accdb;Persist Security Info=True");

//private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Project 2016\PensionVerificationSystem\PensionVerificationSystem\dbPension.accdb;Persist Security Info=True");

private OleDbCommand cmd = new OleDbCommand();

private OleDbDataReader dr;

public FrmChangePassword()

{

InitializeComponent();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnOk\_Click(object sender, EventArgs e)

{

try

{

if ((isAllEntryFilled()) && (isRegistered(txtUsername.Text)))

{

string Username = txtUsername.Text;

string Password = txtPassword.Text;

/\*cmd.CommandType = CommandType.Text;

cmd.CommandText = "UPDATE tblUser set Password = @Password WHERE [Username] = @Username and [ID] = @Id";

cmd.Parameters.AddWithValue("@Password", txtPassword.Text);

cmd.Parameters.AddWithValue("@Username", txtUsername.Text);

cmd.Parameters.AddWithValue("@Id", id);

UPDATE tblUser SET [Password] = 'admin' WHERE (Username = 'Administrator') AND (ID = 1)

\*/

cmd.CommandText = "UPDATE tblUser SET [Password] = '" + Password+ "' WHERE (Username = '"+Username+ "')";

cmd.Connection = cn;

cn.Open();

cmd.ExecuteNonQuery();

MessageBox.Show("Authorized User's Password has been successfully Changed!!!", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private bool isAllEntryFilled()

{

bool isEntryFilled = false;

if ((txtUsername.Text == ""))

{

MessageBox.Show("Please enter the username", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtUsername.Focus();

return isEntryFilled;

}

else if ((txtPassword.Text == ""))

{

MessageBox.Show("Please enter the User Password", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtPassword.Focus();

return isEntryFilled;

}

else if ((txtConfirmPassword.Text == ""))

{

MessageBox.Show("Please enter the user confirm password", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtConfirmPassword.Focus();

return isEntryFilled;

}

else if ((txtPassword.Text != txtConfirmPassword.Text))

{

MessageBox.Show("Password and Confirm Password mismatch. Please retype either password or confirm password", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtPassword.Focus();

return isEntryFilled;

}

else

return true;

}

private bool isRegistered(string user)

{

bool isUserRegistered = true;

try

{

cn.Open();

cmd.CommandText = "select \* from tblUser where Username ='" + user + "'";

dr = cmd.ExecuteReader();

if (!dr.HasRows)

{

MessageBox.Show("Fail to change password!!! Username not exist. No user found ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

isUserRegistered = false;

}

/\* else

{

dr.Read();

id = Convert.ToInt16(dr["ID"].ToString());

}\*/

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Pension Verification App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return isUserRegistered;

}

private void FrmChangePassword\_Load(object sender, EventArgs e)

{

cmd.Connection = cn;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace PensionVerificationSystem

{

public partial class FrmPensionEnrollment : Form

{

private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Application.StartupPath + "\\dbPension.accdb;Persist Security Info=True");

//private OleDbConnection cn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Project 2016\PensionVerificationSystem\PensionVerificationSystem\dbPension.accdb;Persist Security Info=True");

private OleDbCommand cmd = new OleDbCommand();

private OleDbDataReader dr;

string picture = "";

public FrmPensionEnrollment()

{

InitializeComponent();

}

private void FrmPensionEnrollment\_Load(object sender, EventArgs e)

{

cmd.Connection = cn;

txtUniquePensionerId.Text = genPensionerId();

txtUniquePensionerId.Enabled = false;

}

private string genPensionerId()

{

string pensionId = "PTAD/OS/";

try

{

int rowCount = 0;

cn.Open();

cmd.CommandText = "select count(\*) as RowCount from tblPensionerEnrollment";

dr = cmd.ExecuteReader();

while(dr.Read())

rowCount = (int)(dr["RowCount"]);

rowCount++;

if(rowCount < 10)

{

pensionId = pensionId + "000" + rowCount;

}

else if(rowCount < 100)

{

pensionId = pensionId + "00" + rowCount;

}

else if(rowCount < 1000)

{

pensionId = pensionId + "0" + rowCount;

}

else if(rowCount < 10000)

{

pensionId = pensionId + rowCount;

}

dr.Close();

cn.Close();

}

catch(Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Pension Verification App", MessageBoxButtons.OK,MessageBoxIcon.Error);

}

return pensionId;

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void btnRefresh\_Click(object sender, EventArgs e)

{

txtUniquePensionerId.Text = genPensionerId();

txtPensionerFullname.ResetText();

dtpDateOfBirth.ResetText();

cboPensionerGender.ResetText();

dtpDateOfFirstAppointment.ResetText();

txtSalaryStructure.ResetText();

cboGradeLevel.ResetText();

txtCurrentMonthlyPension.ResetText();

cboMinistry.ResetText();

cboBankName.ResetText();

txtBranchAddress.ResetText();

txtAccountNumber.ResetText();

txtBranchSortCode.ResetText();

txtNextOfKinName.ResetText();

txtNextOfKinAddress.ResetText();

txtNextOfKinPhoneNumber.ResetText();

txtNextOfKinRelationship.ResetText();

pictureBox1.Image = null;

pictureBox1.Width = 282; pictureBox1.Height = 201;

picture = "";

txtPensionerFullname.Focus();

}

private void btnEnroll\_Click(object sender, EventArgs e)

{

try

{

if (isAllEntryFilled())

{

string UniquePensionerId = txtUniquePensionerId.Text;

string PensionerFullname = txtPensionerFullname.Text;

string DateOfBirth = dtpDateOfBirth.Text;

string PensionerGender = cboPensionerGender.Text;

string DateOfFirstAppointment = dtpDateOfFirstAppointment.Text;

string SalaryStructure = txtSalaryStructure.Text;

string GradeLevel = cboGradeLevel.Text;

double CurrentMonthlyPension = Convert.ToDouble(txtCurrentMonthlyPension.Text);

string Ministry = cboMinistry.Text;

string BankName = cboBankName.Text;

string BranchAddress = txtBranchAddress.Text;

string AccountNumber = txtAccountNumber.Text;

string BranchSortCode = txtBranchSortCode.Text;

string NextOfKinName = txtNextOfKinName.Text;

string NextOfKinAddress = txtNextOfKinAddress.Text;

string NextOfKinPhoneNumber = txtNextOfKinPhoneNumber.Text;

string NextOfKinRelationship = txtNextOfKinRelationship.Text;

//pictureBox1.Text;

if (isSavePicture())

{

cn.Open();

cmd.CommandText = "insert into tblPensionerEnrollment values('" + UniquePensionerId + "','" + PensionerFullname + "','" + DateOfBirth + "','" + PensionerGender + "','" + DateOfFirstAppointment + "','" + SalaryStructure + "','" + GradeLevel + "'," + CurrentMonthlyPension + ",'" + Ministry + "','" + BankName + "','" + BranchAddress + "','" + AccountNumber + "','" + BranchSortCode + "','" + NextOfKinName + "','" + NextOfKinAddress + "','" + NextOfKinPhoneNumber + "','" + NextOfKinRelationship + "','" + picture + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("New Pensioner Enrollment Information successfully saved!!!", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

btnRefresh\_Click(sender, e);

}

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private bool isAllEntryFilled()

{

bool isEntryFilled = false;

if ((txtUniquePensionerId.Text == "PTAD/OS/"))

{

MessageBox.Show("Please click refresh button to regenerate pensioner's id", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtPensionerFullname.Focus();

return isEntryFilled;

}

else if ((txtPensionerFullname.Text == ""))

{

MessageBox.Show("Please enter the Pensioner's FullName", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtPensionerFullname.Focus();

return isEntryFilled;

}

else if ((dtpDateOfBirth.Text == ""))

{

MessageBox.Show("Please set the Pensioner's Date of Birth", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

dtpDateOfBirth.Focus();

return isEntryFilled;

}

else if ((cboPensionerGender.Text == ""))

{

MessageBox.Show("Please set the Pensioner's Gender", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboPensionerGender.Focus();

return isEntryFilled;

}

else if ((dtpDateOfFirstAppointment.Text == ""))

{

MessageBox.Show("Please set the Pensioner's Date of First Appointment ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

dtpDateOfFirstAppointment.Focus();

return isEntryFilled;

}

else if ((txtSalaryStructure.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Salary Structure ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtSalaryStructure.Focus();

return isEntryFilled;

}

else if ((cboGradeLevel.Text == ""))

{

MessageBox.Show("Please set the Pensioner's Grade Level ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboGradeLevel.Focus();

return isEntryFilled;

}

else if ((txtCurrentMonthlyPension.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Current Monthly Pension ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtCurrentMonthlyPension.Focus();

return isEntryFilled;

}

else if ((cboMinistry.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Ministry ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboMinistry.Focus();

return isEntryFilled;

}

else if ((cboBankName.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Bank Name ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboBankName.Focus();

return isEntryFilled;

}

else if ((txtBranchAddress.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Branch Address ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtBranchAddress.Focus();

return isEntryFilled;

}

else if ((txtAccountNumber.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Account Number ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtAccountNumber.Focus();

return isEntryFilled;

}

else if ((txtBranchSortCode.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Branch Sort Code ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtBranchSortCode.Focus();

return isEntryFilled;

}

else if ((txtNextOfKinName.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Next of Kin Name ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtNextOfKinName.Focus();

return isEntryFilled;

}

else if ((txtNextOfKinAddress.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Next of Kin Address ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtNextOfKinAddress.Focus();

return isEntryFilled;

}

else if ((txtNextOfKinPhoneNumber.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Next of Kin Phone Number ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtNextOfKinPhoneNumber.Focus();

return isEntryFilled;

}

else if ((txtNextOfKinRelationship.Text == ""))

{

MessageBox.Show("Please type the Pensioner's Next of Kin Relationship ", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtNextOfKinRelationship.Focus();

return isEntryFilled;

}

else if ((picture == "") || (pictureBox1.Image == null))

{

MessageBox.Show("Please upload the Pensioner's recent passport", "Pension Verification App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

return isEntryFilled;

}

else

return true;

//pictureBox1.ResetText();

}

private bool isSavePicture()

{

bool isSaved = false;

if ((picture != "") && (pictureBox1.Image != null))

{

if (txtUniquePensionerId.Text != "PTAD/OS/")

{

string pictureFileName = getLastFourString(txtUniquePensionerId.Text);

if (File.Exists(Application.StartupPath + "\\Pictures\\"+ pictureFileName +".jpg"))

{

MessageBox.Show("Picture Cannot be Saved!!!, Duplicate Found", "Filename Already Exists");

}

else

{

pictureBox1.Image.Save(Application.StartupPath + "\\Pictures\\" + pictureFileName + ".jpg");

MessageBox.Show("Picture Saved!!!");

picture = Application.StartupPath + "\\Pictures\\" + pictureFileName + ".jpg";

isSaved = true;

}

}

}

return isSaved;

}

private string getLastFourString(string txtPensionerId)

{

string lastFourString = txtPensionerId.Substring(8,4);

MessageBox.Show(lastFourString);

return lastFourString;

}

private void btnUpload\_Click(object sender, EventArgs e)

{

try

{

MessageBox.Show("Always ensure that upload the recent passport of the present pensioner");

OpenFileDialog openFileDialog = new OpenFileDialog();

//openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

openFileDialog.InitialDirectory = @"C:\";

openFileDialog.Filter = "Image Files (\*.jpg;\*.jpeg;\*.gif;\*.bmp)|\*.jpg;\*.jpeg;\*.gif;\*.bmp|All Files (\*.\*)|\*.\*";

if (openFileDialog.ShowDialog(this) == DialogResult.OK)

{

Bitmap uploadImage = new Bitmap(openFileDialog.FileName);

pictureBox1.Image = resizeImage(uploadImage, 162, 162);

pictureBox1.Size = pictureBox1.Image.Size;

picture = openFileDialog.FileName;

//MessageBox.Show(picture);

}

}

catch (Exception er)

{

MessageBox.Show(er.Message);

}

}

private static Image resizeImage(Image image, int imgWidth, int imgHeight)

{

Bitmap newImage = new Bitmap(imgWidth, imgHeight);

Graphics g = Graphics.FromImage((Image)newImage);

g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.High;

g.DrawImage(image, 0, 0, imgWidth, imgHeight);

return newImage;

}

}

}