using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Threading;

using AForge.Video;

using AForge.Video.DirectShow;

using MessagingToolkit.Barcode;

using BasselTech\_CamCapture;

using FireSharp.Config;

using FireSharp.Response;

using FireSharp.Interfaces;

namespace GatewayExpress

{

public partial class EToll : Form

{

Thread th; //declaration of thread as th

MyUser c = new MyUser();

string orNumber;

int panelWidth;

int en, ex;

double Balance = 0;

string birthDate, cityAddress, dateAndTime, gender, homeAddress, licenseNumber, middleName, nationality, password, phoneNumber, telephoneNumber, type, status, accountStatus, purpose;

bool Hidden;

void clear()//sub class clear

{

//command for clearing the input

txtId.Text = "";

txtUsername.Text = "";

txtGivenName.Text = "";

txtLastName.Text = "";

txtAmount.Text = "";

Balance = 0;

}

FilterInfoCollection filterInfoCollection;

VideoCaptureDevice captureDevice;

IFirebaseConfig ifc = new FirebaseConfig()

{

AuthSecret = "P2k1udVSBnRvxFKNk1Ve5OVZo8A4x90jWAEFXqYH",

BasePath = "https://lancaster-new-city-default-rtdb.firebaseio.com/"

};

IFirebaseClient client;

public EToll()

{

InitializeComponent();

panelWidth = PanelSlide.Width;

Hidden = false;

timer3.Start();

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (Hidden)

{

PanelSlide.Width = PanelSlide.Width + 10;

if (PanelSlide.Width >= panelWidth)

{

timer1.Stop();

Hidden = false;

this.Refresh();

}

}

else

{

PanelSlide.Width = PanelSlide.Width - 10;

if (PanelSlide.Width <= 0)

{

timer1.Stop();

Hidden = true;

this.Refresh();

}

}

}

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

{

try

{

captureDevice = new VideoCaptureDevice(filterInfoCollection[comboBox1.SelectedIndex].MonikerString);

captureDevice.NewFrame += CaptureDevice\_NewFrame;

captureDevice.Start();

timer2.Start();

}

catch (Exception ex)

{

timer2.Stop();

MessageBox.Show(ex.Message);

}

}

private void CaptureDevice\_NewFrame(object sender, NewFrameEventArgs eventArgs)

{

ptbScanner.Image = (Bitmap)eventArgs.Frame.Clone();

}

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

{

try

{

client = new FireSharp.FirebaseClient(ifc);

}

catch

{

MessageBox.Show("No Internet or Connection Problem");

}

filterInfoCollection = new FilterInfoCollection(FilterCategory.VideoInputDevice);

foreach (FilterInfo filterInfo in filterInfoCollection)

comboBox1.Items.Add(filterInfo.Name);

comboBox1.SelectedIndex = 0;

}

private void EToll\_FormClosing(object sender, FormClosingEventArgs e)

{

try

{

if (captureDevice.IsRunning)

captureDevice.Stop();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void timer2\_Tick(object sender, EventArgs e)

{

try

{

if (ptbScanner.Image != null)

{

BarcodeDecoder barcodeDecoder = new BarcodeDecoder();

Result result = barcodeDecoder.Decode((Bitmap)ptbScanner.Image);

if (result != null)

{

txtId.Text = result.ToString();

timer2.Stop();

if (captureDevice.IsRunning)

captureDevice.Stop();

try

{

if (txtId.Text != "")

{

FirebaseResponse res2 = client.Get(@"USER INFO/" + txtId.Text + "/BALANCE");

string holder1 = res2.ResultAs<String>();

Balance = Double.Parse(holder1);

if (Balance >= Double.Parse(txtAmount.Text))

{

FirebaseResponse res = client.Get(@"COUNTER/COUNTER");

int Counter = int.Parse(res.ResultAs<String>());

Counter++;

MyUser std1 = new MyUser()

{

COUNTER = Counter.ToString()

};

var setter1 = client.Set("COUNTER", std1);

if (Counter < 10) { orNumber = "0000000" + Counter.ToString(); }

else if (Counter < 99) { orNumber = "000000" + Counter.ToString(); }

else if (Counter < 999) { orNumber = "00000" + Counter.ToString(); }

else if (Counter < 9999) { orNumber = "0000" + Counter.ToString(); }

else if (Counter < 99999) { orNumber = "000" + Counter.ToString(); }

else if (Counter < 999999) { orNumber = "00" + Counter.ToString(); }

else if (Counter < 9999999) { orNumber = "0" + Counter.ToString(); }

else { Counter.ToString(); }

MyUser std2 = new MyUser()

{

OR\_NUMBER = orNumber.ToString().Trim()

};

var setter2 = client.Set("SERIAL NUMBER/" + Counter.ToString(), std2);

MyUser std3 = new MyUser()

{

OR\_NUMBER = orNumber.ToString(),

USERNAME = txtUsername.Text,

GIVEN\_NAME = txtGivenName.Text,

LAST\_NAME = txtLastName.Text,

DATE = lblDate.Text,

TIME = lblTime.Text,

AMOUNT = txtAmount.Text,

STATUS = status,

PURPOSE = purpose

};

var setter3 = client.Set("TOLL/" + orNumber.ToString(), std3);

Balance = Balance - Double.Parse(txtAmount.Text);

MyUser std4 = new MyUser()

{

USERNAME = txtUsername.Text,

PASSWORD = password,

GIVEN\_NAME = txtGivenName.Text,

MIDDLE\_NAME = middleName,

LAST\_NAME = txtLastName.Text,

PHONE\_NUMBER = phoneNumber,

TELEPHONE\_NUMBER = telephoneNumber,

BIRTHDATE = birthDate,

GENDER = gender,

NATIONALITY = nationality,

HOME\_ADDRESS = homeAddress,

CITY\_ADDRESS = cityAddress,

DATE\_AND\_TIME = dateAndTime,

STATUS = status,

BALANCE = Balance.ToString(),

ACCOUNT\_STATUS = accountStatus,

PURPOSE = purpose

};

var setter4 = client.Set("USER INFO/" + txtId.Text, std4);

MessageBox.Show("Data Inserted Successfully!");

if (status == "Visitor")

{

th = new Thread(openVisitorsPass); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

}

else

{

MessageBox.Show("Not Enough Balance");

}

}

else

{

MessageBox.Show("Scan A Bar Code First");

}

timer2.Stop();

if (captureDevice.IsRunning)

captureDevice.Stop();

}

catch (Exception ex1)

{

}

clear();

}

else

{

timer2.Stop();

if (captureDevice.IsRunning)

captureDevice.Stop();

//MessageBox.Show("There is no QR code Scanned");

}

}

}

catch(Exception ex)

{

timer2.Stop();

if (captureDevice.IsRunning)

captureDevice.Stop();

//MessageBox.Show("There is no QR code Scanned");

}

}

private void timer3\_Tick(object sender, EventArgs e)

{

DateTime time = DateTime.Now;

this.lblDate.Text = time.ToString("MM/dd/yyyy");

this.lblTime.Text = time.ToString("hh:mm tt");

}

private void txtId\_TextChanged(object sender, EventArgs e)

{

try

{

if (txtId.Text != "")

{

var result2 = client.Get("USER INFO/" + txtId.Text);

MyUser std = result2.ResultAs<MyUser>();

if (std.STATUS == "Home Owner") {

txtAmount.Text = "00.00";

}

else if (std.STATUS == "Visitor") {

txtAmount.Text = "30.00";

}

else {

txtAmount.Text = "50.00";

}

txtUsername.Text = std.USERNAME;

txtGivenName.Text = std.GIVEN\_NAME;

txtLastName.Text = std.LAST\_NAME;

birthDate = std.BIRTHDATE;

cityAddress = std.CITY\_ADDRESS;

dateAndTime = std.DATE\_AND\_TIME;

gender = std.GENDER;

homeAddress = std.HOME\_ADDRESS;

licenseNumber = std.LICENSE\_NUMBER;

middleName = std.MIDDLE\_NAME;

nationality = std.NATIONALITY;

password = std.PASSWORD;

phoneNumber = std.PHONE\_NUMBER;

telephoneNumber = std.TELEPHONE\_NUMBER;

status = std.STATUS;

accountStatus = std.ACCOUNT\_STATUS;

purpose = std.PURPOSE;

}

else {

//MessageBox.Show("There is no QR code Scanned");

}

}

catch (Exception ex)

{

//MessageBox.Show("There is no QR code Scanned");

}

}

private void openNewDashBoad(object obj)

{

Application.Run(new Dashboard()); //Run Dashboard

}

private void openNewLoad(object obj)

{

Application.Run(new Load()); //Run Load Tab

}

private void openNewForm1(object obj)

{

Application.Run(new Form1()); //Run Form1 or LoginForm Tab

}

private void openSplashScreen(object obj)

{

Application.Run(new SplashScreen()); //Run SplashScreen

}

private void openNewDrivers(object obj)

{

Application.Run(new Drivers()); //Run Drivers or User's Info Tab

}

private void openNewAdmins(object obj)

{

Application.Run(new Admins()); //Run Admin's Info Tab

}

private void openVisitorsPass(object obj)

{

Application.Run(new VisitorsPass()); //Run VisitorsPass Tab

}

private void btnDashboard\_Click\_1(object sender, EventArgs e)

{

this.Close(); //command for closing the application

th = new Thread(openSplashScreen); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Refresh();

}

private void btnLoad\_Click\_1(object sender, EventArgs e)

{

this.Close(); //command for closing the application

th = new Thread(openNewLoad); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

private void btnLogout\_Click\_1(object sender, EventArgs e)

{

this.Close(); //command for closing the application

th = new Thread(openNewForm1); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewAdmins); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewDrivers); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

try

{

if (txtId.Text != "")

{

FirebaseResponse res2 = client.Get(@"USER INFO/" + txtId.Text + "/BALANCE");

string holder1 = res2.ResultAs<String>();

Balance = Double.Parse(holder1);

if (Balance >= Double.Parse(txtAmount.Text))

{

FirebaseResponse res = client.Get(@"COUNTER/COUNTER");

int Counter = int.Parse(res.ResultAs<String>());

Counter++;

MyUser std1 = new MyUser()

{

COUNTER = Counter.ToString()

};

var setter1 = client.Set("COUNTER", std1);

if (Counter < 10) { orNumber = "0000000" + Counter.ToString(); }

else if (Counter < 99) { orNumber = "000000" + Counter.ToString(); }

else if (Counter < 999) { orNumber = "00000" + Counter.ToString(); }

else if (Counter < 9999) { orNumber = "0000" + Counter.ToString(); }

else if (Counter < 99999) { orNumber = "000" + Counter.ToString(); }

else if (Counter < 999999) { orNumber = "00" + Counter.ToString(); }

else if (Counter < 9999999) { orNumber = "0" + Counter.ToString(); }

else { Counter.ToString(); }

MyUser std2 = new MyUser()

{

OR\_NUMBER = orNumber.ToString().Trim()

};

var setter2 = client.Set("SERIAL NUMBER/" + Counter.ToString(), std2);

MyUser std3 = new MyUser()

{

OR\_NUMBER = orNumber.ToString(),

USERNAME = txtUsername.Text,

GIVEN\_NAME = txtGivenName.Text,

LAST\_NAME = txtLastName.Text,

DATE = lblDate.Text,

TIME = lblTime.Text,

AMOUNT = txtAmount.Text,

STATUS = status,

PURPOSE = purpose

};

var setter3 = client.Set("TOLL/" + orNumber.ToString(), std3);

Balance = Balance - Double.Parse(txtAmount.Text);

MyUser std4 = new MyUser()

{

USERNAME = txtUsername.Text,

PASSWORD = password,

GIVEN\_NAME = txtGivenName.Text,

MIDDLE\_NAME = middleName,

LAST\_NAME = txtLastName.Text,

PHONE\_NUMBER = phoneNumber,

TELEPHONE\_NUMBER = telephoneNumber,

BIRTHDATE = birthDate,

GENDER = gender,

NATIONALITY = nationality,

HOME\_ADDRESS = homeAddress,

CITY\_ADDRESS = cityAddress,

DATE\_AND\_TIME = dateAndTime,

STATUS = status,

BALANCE = Balance.ToString(),

ACCOUNT\_STATUS = accountStatus,

PURPOSE = purpose

};

var setter4 = client.Set("USER INFO/" + txtId.Text, std4);

MessageBox.Show("Data Inserted Successfully!");

if (status == "Visitor")

{

th = new Thread(openVisitorsPass); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

clear();

}

else

{

MessageBox.Show("Not Enough Balance");

}

}

else

{

txtAmount.Text = "50.00";

FirebaseResponse res = client.Get(@"COUNTER/COUNTER");

int Counter = int.Parse(res.ResultAs<String>());

Counter++;

MyUser std1 = new MyUser()

{

COUNTER = Counter.ToString()

};

var setter1 = client.Set("COUNTER", std1);

if (Counter < 10) { orNumber = "0000000" + Counter.ToString(); }

else if (Counter < 99) { orNumber = "000000" + Counter.ToString(); }

else if (Counter < 999) { orNumber = "00000" + Counter.ToString(); }

else if (Counter < 9999) { orNumber = "0000" + Counter.ToString(); }

else if (Counter < 99999) { orNumber = "000" + Counter.ToString(); }

else if (Counter < 999999) { orNumber = "00" + Counter.ToString(); }

else if (Counter < 9999999) { orNumber = "0" + Counter.ToString(); }

else { Counter.ToString(); }

MyUser std2 = new MyUser()

{

OR\_NUMBER = orNumber.ToString().Trim()

};

var setter2 = client.Set("SERIAL NUMBER/" + Counter.ToString(), std2);

if (txtUsername.Text != "")

{

MyUser std3 = new MyUser()

{

OR\_NUMBER = orNumber.ToString(),

USERNAME = txtUsername.Text,

GIVEN\_NAME = txtGivenName.Text,

LAST\_NAME = txtLastName.Text,

DATE = lblDate.Text,

TIME = lblTime.Text,

AMOUNT = txtAmount.Text,

STATUS = "Passer"

};

var setter3 = client.Set("TOLL/" + orNumber.ToString(), std3);

}

else

{

MyUser std3 = new MyUser()

{

OR\_NUMBER = orNumber.ToString(),

USERNAME = "MANUAL PUNCH",

GIVEN\_NAME = txtGivenName.Text,

LAST\_NAME = txtLastName.Text,

DATE = lblDate.Text,

TIME = lblTime.Text,

AMOUNT = txtAmount.Text,

STATUS = "Passer"

};

var setter3 = client.Set("TOLL/" + orNumber.ToString(), std3);

}

MessageBox.Show("Data Inserted Successfully!");

clear();

}

timer2.Stop();

if (captureDevice.IsRunning)

captureDevice.Stop();

}

catch (Exception ex)

{

timer2.Stop();

//MessageBox.Show(ex.Message);

}

}

}

}