

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;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

frmLocationChooser f = new frmLocationChooser();

f.ShowDialog();

}

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

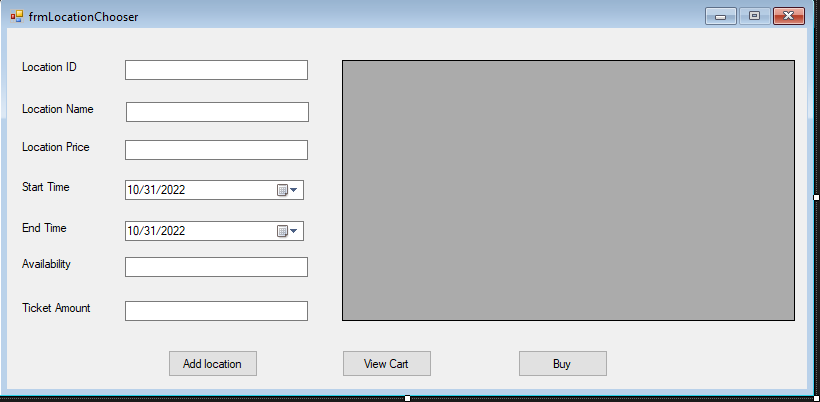
{

Application.Exit();

}

}

}



using System;

Sẽ chaỵ dữ liệu từ table location, display danh sách bên datagridview và đồng thời cũng thể hiện những thông tin cơ bản bên tay trái

Add location để thêm location

View Cart để xem những location mình đã mua

Buy để đi thanh toán những ticket mình đã mua

public static string frmUserSelectedChoice = "";

được dùng để lưu những thông tin mà user đã chọn

public static ArrayList frmInvoiceCreated = new ArrayList(); được dùng để lưu các chi tiết hóa đơn đã được tạo, dữ liệu này sẽ được gửi qua những form khác để lưu trong database

private int ticketCounter = 0; để tạo số hóa đơn tự động, biến này sẽ đếm bên trong database có bao nhiêu hóa đơn

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.Configuration;

using System.Data.SqlClient;

using System.Collections;

namespace WindowsFormsApp1

{

public partial class frmLocationChooser : Form

{

public static string frmUserSelectedChoice = "";

public static ArrayList frmInvoiceCreated = new ArrayList();

private int ticketCounter = 0;

public frmLocationChooser()

{

InitializeComponent();

}

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

{

btnAdd.Enabled = false;

txtbxLocationID.Enabled = false;

txtbxLocationName.Enabled = false;

txtbxLocationPrice.Enabled = false;

txtbxTicketAmount.Enabled = false;

txtbxLocationAva.Enabled = false;

dateTimePickerEndTime.Enabled = false;

dateTimePickerStartTime.Enabled = false;

RefreshData();

}

private void RefreshData()

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "select \* from Location";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

dataGridViewLocation.DataSource = dt;

}

else

{

MessageBox.Show("No Data", "Warning No Data");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error");

}

}

private void dataGridViewLocation\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

this.ActiveControl = txtbxTicketAmount;

btnAdd.Enabled = true;

txtbxTicketAmount.Enabled = true;

int index = dataGridViewLocation.SelectedCells[0].RowIndex;

if(index < 0 || index >= dataGridViewLocation.Rows.Count)

{

return;

}

DataGridViewRow row = dataGridViewLocation.Rows[index];

String locationID = row.Cells[0].Value.ToString().Trim();

String locationName = row.Cells[1].Value.ToString().Trim();

String locationPrice = row.Cells[2].Value.ToString().Trim();

String startDate = row.Cells[3].Value.ToString().Trim();

String endDate = row.Cells[4].Value.ToString().Trim();

int availability = int.Parse(row.Cells[5].Value.ToString().Trim());

txtbxLocationID.Text = locationID;

txtbxLocationName.Text = locationName;

txtbxLocationPrice.Text = locationPrice;

txtbxLocationAva.Text = availability.ToString();

dateTimePickerStartTime.Text = startDate;

dateTimePickerEndTime.Text = endDate;

}

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

{

if(!txtbxTicketAmount.Text.ToString().Trim().Equals(""))

{

if (int.Parse(txtbxLocationAva.Text.ToString().Trim()) < int.Parse(txtbxTicketAmount.Text.ToString().Trim()))

{

MessageBox.Show("No more space", "Warning Box");

return;

}

String userSelectedLocation = txtbxLocationID.Text.ToString().Trim() + ";" + txtbxLocationName.Text.ToString().Trim() + ";" + txtbxLocationPrice.Text.ToString().Trim() + ";" + dateTimePickerStartTime.Text.ToString().Trim() + ";" + dateTimePickerEndTime.Text.ToString().Trim() + ";" + txtbxLocationAva.Text.ToString().Trim() + ";" + txtbxTicketAmount.Text.ToString().Trim();

frmUserSelectedChoice = userSelectedLocation;

frmCheckout f = new frmCheckout();

f.ShowDialog();

}

else

{

MessageBox.Show("Please enter amount", "Warning Box");

}

}

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

{

if(frmInvoiceCreated.Count == 0)

{

MessageBox.Show("Cart Empty", "Notification");

}

else

{

String displayIndex = "";

foreach (String i in frmInvoiceCreated) {

displayIndex = displayIndex + i + "\n";

}

MessageBox.Show(displayIndex, "Notification");

}

}

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

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "select \* from Invoice";

SqlCommand cmd = new SqlCommand(sSQL, conn);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

frmCreateInvoice.frmInvoiceID = invoiceID; để đưa cái invoiceID xuyên các form

frmCreateInvoice.frmInvoiceCreated = frmInvoiceCreated; bên form frmCreateInvoice cũng có 1 array list là frmInvoiceCreated, chúng ta gán this.frmInvoiceCreated sang frmCreateInvoice.frmInvoiceCreated

invoiceID được tạo bằng cách lấy ngày hiện tại + ticketCounter (được đếm trong database)

adapter.Fill(dt);

this.ticketCounter = dt.Rows.Count + 1;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error");

}

String invoiceID = DateTime.Now.ToString().Split(' ')[0].Replace("/", "").Trim() + this.ticketCounter.ToString().Trim();

frmCreateInvoice.frmInvoiceID = invoiceID;

frmCreateInvoice.frmInvoiceCreated = frmInvoiceCreated;

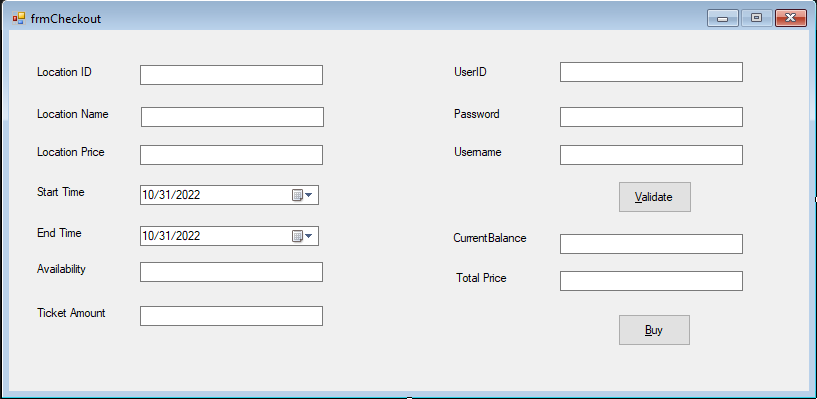
frmCreateInvoice f = new frmCreateInvoice();

f.ShowDialog();

}

}

}



using System;

Khi validate thông tin chính xác thỉ sẽ được lưu vào trong cart và trả về frmLocationChooser, sau đó có những biến này

private String locationID = "";

private String userID = "";

private String ticketAmount = "";

private String totalPrice = "";

được dùng để trả giá trị cho frmCreateInvoice

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.Configuration;

using System.Data.SqlClient;

namespace WindowsFormsApp1

{

public partial class frmCheckout : Form

{

private String locationID = "";

private String userID = "";

private String ticketAmount = "";

private String totalPrice = "";

public frmCheckout()

{

InitializeComponent();

}

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

{

this.ActiveControl = txtbxUserID;

String[] locationInfo = frmLocationChooser.frmUserSelectedChoice.Split(';');

String locationID = locationInfo[0].Trim();

String locationName = locationInfo[1].Trim();

String locationPrice = locationInfo[2].Trim();

String startTime = locationInfo[3].Trim();

String endTime = locationInfo[4].Trim();

String locationAvailability = (int.Parse(locationInfo[5]) - int.Parse(locationInfo[6])).ToString().Trim();

String userAmount = locationInfo[6].Trim();

this.locationID = locationID;

this.ticketAmount = userAmount;

txtbxLocationID.Text = locationID;

txtbxLocationName.Text = locationName;

txtbxLocationPrice.Text = locationPrice;

txtbxTicketAmount.Text = userAmount;

txtbxLocationAva.Text = locationAvailability;

dateTimePickerStartTime.Text = startTime;

dateTimePickerEndTime.Text = endTime;

txtbxTotalPrice.Text = (int.Parse(locationPrice) \* int.Parse(userAmount)).ToString().Trim();

this.totalPrice = (int.Parse(locationPrice) \* int.Parse(userAmount)).ToString().Trim();

txtbxLocationID.Enabled = false;

txtbxLocationName.Enabled = false;

txtbxLocationPrice.Enabled = false;

txtbxTicketAmount.Enabled = false;

txtbxLocationAva.Enabled = false;

dateTimePickerStartTime.Enabled = false;

dateTimePickerEndTime.Enabled = false;

txtbxTotalPrice.Enabled = false;

txtbxUsername.Enabled = false;

txtbxCurrentBalance.Enabled = false;

btnBuy.Enabled = false;

}

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

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "select \* from UserCreditCard where userID = @userid and userPassword = @password";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@userid", txtbxUserID.Text);

cmd.Parameters.AddWithValue("@password", txtbxPassword.Text);

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

MessageBox.Show("Validation Success", "Success");

String userId = (String) dt.Rows[0][0];

String username = (String) dt.Rows[0][1];

int int\_balance = (int) dt.Rows[0][3];

String balance = int\_balance.ToString().Trim();

this.userID = userId;

txtbxUserID.Text = userId;

txtbxUsername.Text = username;

txtbxCurrentBalance.Text = balance;

txtbxPassword.Enabled = false;

txtbxUserID.Enabled = false;

btnUserValidate.Enabled = false; ;

btnBuy.Enabled = true;

}

else

{

MessageBox.Show("Wrong username or password! Try again!", "Warning No User");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error");

}

}

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

{

if(int.Parse(txtbxCurrentBalance.Text.ToString().Trim()) < int.Parse(this.totalPrice.ToString().Trim()))

{

MessageBox.Show("Not enough credit", "Warning");

return;

}

//create invoiceID

//this.ticketCounter += 1;

//String invoiceID = DateTime.Now.ToString().Split(' ')[0].Replace("/","").Trim() + this.ticketCounter.ToString().Trim();

//save locationID, userID, ticketamount, totalPrice

String createInvoice = this.locationID.ToString().Trim() + ";" + this.userID.ToString().Trim() + ";" + this.ticketAmount.ToString().Trim() + ";" + this.totalPrice.ToString().Trim();

frmLocationChooser.frmInvoiceCreated.Add(createInvoice.ToString().Trim());

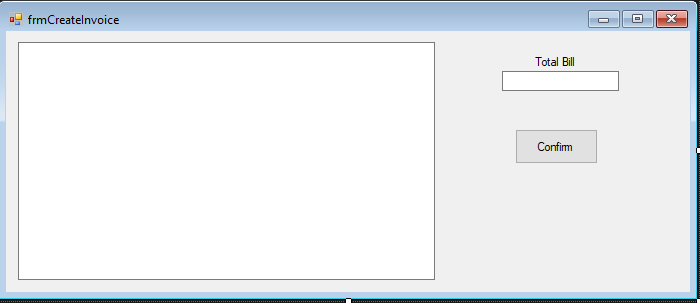
btnBuy.Enabled = false;

this.Close();

}

}

}



public static String frmInvoiceID = "";

public static ArrayList frmInvoiceCreated = new ArrayList();

sẽ lấy dữ liệu của những form trước đó để thực hiện lưu vào trong database

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Configuration;

namespace WindowsFormsApp1

{

public partial class frmCreateInvoice : Form

{

public static String frmInvoiceID = "";

//save locationID, userID, ticketamount, totalPrice

public static ArrayList frmInvoiceCreated = new ArrayList();

public frmCreateInvoice()

{

InitializeComponent();

}

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

{

String displayText = "";

int totalBill = 0;

for(int i = 0; i < frmInvoiceCreated.Count; i++)

{

String[] invoiceInfo = frmInvoiceCreated[i].ToString().Trim().Split(';');

String locationID = invoiceInfo[0];

String userID = invoiceInfo[1];

String ticketAmount = invoiceInfo[2];

String totalPrice = invoiceInfo[3];

displayText = displayText + "LocationID: "+ locationID + " - Ticket Amount: "+ ticketAmount + " - Total: " + totalPrice + "\r\n";

totalBill = totalBill + int.Parse(totalPrice.ToString().Trim());

}

txtDisplay.Text = displayText;

txtDisplay.Enabled = false;

txtbxTotalBill.Text = totalBill.ToString().Trim();

txtbxTotalBill.Enabled = false;

}

private void insertInvoice(String invoiceID, String userID, int totalPrice)

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "insert into Invoice values (@invoiceID, @userID, @totalPrice)";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@invoiceID", invoiceID);

cmd.Parameters.AddWithValue("@userID", userID);

cmd.Parameters.AddWithValue("totalPrice", totalPrice);

int i = cmd.ExecuteNonQuery();

if (i != 0)

{

}

else

{

MessageBox.Show("No Data", "Warning No Data");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error 1");

}

}

private void insertInvoiceDetail(String invoiceID, String locationID, int ticketAmount)

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "insert into InvoiceDetail values (@invoiceID, @locationID, @ticketAmount)";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@invoiceID", invoiceID);

cmd.Parameters.AddWithValue("@locationID", locationID);

cmd.Parameters.AddWithValue("@ticketAmount", ticketAmount);

int i = cmd.ExecuteNonQuery();

if (i != 0)

{

}

else

{

MessageBox.Show("No Data", "Warning No Data");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error 2");

}

}

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

{

String userID = "";

int totalBill = 0;

for (int i = 0; i < frmInvoiceCreated.Count; i++)

{

String[] invoiceInfo = frmInvoiceCreated[i].ToString().Trim().Split(';');

userID = invoiceInfo[1];

String totalPrice = invoiceInfo[3];

totalBill = totalBill + int.Parse(totalPrice.ToString().Trim());

}

insertInvoice(frmInvoiceID, userID, totalBill);

String locationID = "";

int ticketAmount = -1;

for (int i = 0; i < frmInvoiceCreated.Count; i++)

{

String[] invoiceInfo = frmInvoiceCreated[i].ToString().Trim().Split(';');

locationID = invoiceInfo[0];

ticketAmount = int.Parse(invoiceInfo[2]);

insertInvoiceDetail(frmInvoiceID, locationID, ticketAmount);

decreaseAvailability(locationID, ticketAmount);

}

decreaseUserBalanace(totalBill, userID);

MessageBox.Show("Collect your ticket and receipt below!", "Success Purchase");

Application.Exit();

}

private void decreaseUserBalanace(int totalBill, String userID)

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "exec decreaseCash @userCash, @userID";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@userID", userID);

cmd.Parameters.AddWithValue("@userCash", totalBill);

int i = cmd.ExecuteNonQuery();

if (i != 0)

{

MessageBox.Show("Transaction Success", "Success Notification");

}

else

{

MessageBox.Show("No Data", "Warning No Data");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error 2");

}

}

private void decreaseAvailability(String locationID, int freeSpace)

{

try

{

SqlConnection conn = new SqlConnection(Program.strConn);

conn.Open();

String sSQL = "exec decreaseAvailability @locationID, @availability";

SqlCommand cmd = new SqlCommand(sSQL, conn);

cmd.Parameters.AddWithValue("@locationID", locationID);

cmd.Parameters.AddWithValue("@availability", freeSpace);

int i = cmd.ExecuteNonQuery();

if (i != 0)

{

}

else

{

MessageBox.Show("No Data", "Warning No Data");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error 2");

}

}

}

}