















**SOURCE CODE**

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 ComputerizedCarRentalSystem

{

public partial class FrmCarBooking : Form

{

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

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

private OleDbCommand cmd = new OleDbCommand();

private OleDbDataReader dr;

private string UserId;

public FrmCarBooking(string user)

{

InitializeComponent();

UserId = user;

}

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

{

cmd.Connection = cn;

clearAllText();

populateCarRegistrationId();

}

private void populateCarRegistrationId()

{

try

{

cboCarRegId.Items.Clear();

cn.Open();

//cmd.CommandText = "select CarRegId from tblCar Where CarStatus = 'Available!'";

cmd.CommandText = "select CarRegId from tblCar";

dr = cmd.ExecuteReader();

while (dr.Read())

cboCarRegId.Items.Add((dr["CarRegId"]).ToString());

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void clearAllText()

{

//cboCarRegId.ResetText();

txtCarType.ResetText(); txtCarType.Enabled = false;

txtCarModel.ResetText(); txtCarModel.Enabled = false;

txtCarMaker.ResetText(); txtCarMaker.Enabled = false;

txtCarColor.ResetText(); txtCarColor.Enabled = false;

txtCarRentageCost.ResetText(); txtCarRentageCost.Enabled = false;

txtCarPlateNumber.ResetText(); txtCarPlateNumber.Enabled = false;

dtpCarRentageDate.ResetText();// dtpCarRentageDate.Enabled = false;

txtCarDueDate.ResetText(); txtCarDueDate.Enabled = false;

txtCustomerId.ResetText(); txtCustomerId.Enabled = false;

txtTerms.ResetText(); txtTerms.ReadOnly = true;

radioButton1.Checked = false; radioButton1.Enabled = false;

radioButton2.Checked = true; radioButton2.Enabled = false;

btnRegister.Enabled = false;

txtCarBookingDuration.Text ="1";

txtTotalCarRentageCost.Text = "0.00"; txtTotalCarRentageCost.ReadOnly = true;

}

private void cboCarRegId\_SelectedIndexChanged(object sender, EventArgs e)

{

clearAllText();

try

{

cn.Open();

cmd.CommandText = "select \* from tblCar Where CarRegId = '"+ cboCarRegId.Text + "'";

dr = cmd.ExecuteReader();

while (dr.Read())

{

//cboCarRegId.Items.Add((dr["CarRegId"]).ToString());

txtCarType.Text = dr["CarType"].ToString();

txtCarModel.Text = dr["CarModel"].ToString();

txtCarMaker.Text = dr["CarMaker"].ToString();

txtCarColor.Text = dr["CarColor"].ToString();

txtCarRentageCost.Text = dr["CarRentageCost"].ToString();

txtCarPlateNumber.Text = dr["CarPlateNumber"].ToString();

}

dr.Close();

cn.Close();

dtpCarRentageDate.Text = DateTime.Now.ToString();//.ToShortDateString();

txtCarBookingDuration.Text = "1";

txtCarDueDate.Text = DateTime.Now.AddDays(1.0).ToString();

txtCustomerId.Text = UserId;

txtTotalCarRentageCost.Text = txtCarRentageCost.Text;

txtTerms.Text = "The Terms and Conditions for Car Rental System are as follows:\r\n AGREEMENT TO HIRE RENTAL VEHICLE \r\nThe Rental Agreement Comprises these terms and Conditions and the rental document setting out the hire details which is provided with each vehicle at the point of hire(referred to in these terms and conditions as the \"Rental Document\").";

txtTerms.Text = txtTerms.Text + "\r\n The Rental Agreement is made between Rent A Car Limited (herein referred to as \"The Hirer\") whose particulars are recorded in the Rental Document. It is hereby agreed as follows: \r\n VEHICLE DESCRIPTION AND TERM OF HIRE \r\n1. The owner will let and the hirer will take the motor vehicle, details of which are described";

txtTerms.Text = txtTerms.Text + " in the Rental Document(herein referred to as \"The Vehicle\"), for the term of hire as described in the Rental Document.\r\n PAYMENT BY HIRER\r\n2. The Hirer shall pay the owner for the hire of the vehicle the sum or sums specified in the Rental Document(known as \"Total Car Rentage Cost\"); and authorizes the owner to charge";

txtTerms.Text = txtTerms.Text + " all amount payable to the hirer's account. The hirer's account means a nominated debit card, credit card, or pre-arranged charge account. ";

radioButton1.Enabled = true; radioButton2.Enabled = true;

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

btnRegister.Enabled = true;

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

btnRegister.Enabled = false;

}

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

{

try

{

if ((dtpCarRentageDate.Text == "")|| (txtCarDueDate.Text == ""))

{

MessageBox.Show("Please Select the Car Registration Id", "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if ((dtpCarRentageDate.Text != "") && (txtCarDueDate.Text != "") && (!checkAlreadyRented(UserId) && (!isAlreadyBooked(cboCarRegId.Text))))

{

string CarRegId = cboCarRegId.Text;

string BookingDate = dtpCarRentageDate.Text;

string DueDate = txtCarDueDate.Text;

string ReturnDate = " ";

int CarBookingDuration = Convert.ToInt16(txtCarBookingDuration.Text);

double totalCarRentageCost = Convert.ToDouble(txtTotalCarRentageCost.Text);

int BookingStatus = 1;

int CustomerBookingStatus = GetCustomerBookingStatus(UserId);

{

cn.Open();

cmd.CommandText = "insert into tblBooking values('" + CarRegId + "','" + UserId + "'," + BookingStatus + ",'" + BookingDate + "','" + DueDate + "','" + ReturnDate + "'," + CarBookingDuration + "," + totalCarRentageCost + ")";

cmd.ExecuteNonQuery();

cmd.CommandText = "update tblCar set CarStatus = 'Not Available! Car already rented' where (CarRegId = '" + CarRegId + "')";

cmd.ExecuteNonQuery();

cmd.CommandText = "update tblCustomer set BookingStatus = "+ CustomerBookingStatus + " where (CustomerId = '" + UserId + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("This Car has been successfully Rented!!!", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

clearAllText();

populateCarRegistrationId();

}

}

else

{

MessageBox.Show("Dear Customer, Am sorry to inform you can't complete this operation because you need to return one of the three cars already rented by you", "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception Er)

{

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private bool checkAlreadyRented(string user)

{

bool isAlreadyRented = false;

try

{

cn.Open();

cmd.CommandText = "select BookingStatus from tblCustomer Where (CustomerId = '" + user + "') AND (BookingStatus = 3)";

dr = cmd.ExecuteReader();

if (dr.HasRows)

{

isAlreadyRented = true; //the user has already rented a car before

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return isAlreadyRented;

}

private bool isAlreadyBooked(string carId)

{

bool isAlreadyRented = false;

try

{

cn.Open();

cmd.CommandText = "select CarRegId from tblCar Where (CarRegId = '" + carId +"') AND (CarStatus = 'Available!')";

dr = cmd.ExecuteReader();

if (!dr.HasRows)

{

MessageBox.Show("The Selected car is already rented before (i.e. not available), Book another car", "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

isAlreadyRented = true; //the car is already rented before (i.e. not available)

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return isAlreadyRented;

}

private int GetCustomerBookingStatus(string user)

{

int bookingStatus = 0;

try

{

cn.Open();

cmd.CommandText = "select BookingStatus from tblCustomer Where (CustomerId = '" + user + "') AND (BookingStatus < 3)";

dr = cmd.ExecuteReader();

while (dr.Read())

{

bookingStatus = Convert.ToInt16(dr["BookingStatus"]); //to get the number of booking made by user

}

if (bookingStatus < 3) bookingStatus++;

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return bookingStatus;

}

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

{

this.Hide();

}

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

{

if (txtCarBookingDuration.Text == "")

{

MessageBox.Show("Dear Customer, Your Car booking Duration should not be empty ", "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtCarBookingDuration.Text = "1";

}

else if (Convert.ToInt16(txtCarBookingDuration.Text) > 7)

{

MessageBox.Show("Dear Customer, Your Car booking Duration should not exceed 7 Days ", "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtCarBookingDuration.Text = "7";

}

DateTime bookingDate = dtpCarRentageDate.Value;

int CarBookingDuration = Convert.ToInt16(txtCarBookingDuration.Text);

txtCarDueDate.Text = bookingDate.AddDays(CarBookingDuration).ToString();

txtTotalCarRentageCost.Text = (Convert.ToDouble(txtCarRentageCost.Text) \* CarBookingDuration).ToString();

}

}

}

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 ComputerizedCarRentalSystem

{

public partial class FrmCarRegistration : Form

{

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

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

private OleDbCommand cmd = new OleDbCommand();

private OleDbDataReader dr;

public FrmCarRegistration()

{

InitializeComponent();

}

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

{

this.Hide();

}

private void btnRefresh()

{

txtCarRegistrationId.Text = genCarRegistrationId();

cboCarType.ResetText();

cboCarModel.ResetText();

cboCarMaker.ResetText();

txtCarColor.ResetText();

txtCarRentageCost.ResetText();

txtCarPlateNumber.ResetText();

}

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

{

try

{

if (isAllEntryFilled())

{

string CarRegId = txtCarRegistrationId.Text;

string CarType = cboCarType.Text;

string CarModel = cboCarModel.Text;

string CarMaker = cboCarMaker.Text;

string CarColor = txtCarColor.Text;

string CarRentageCost = txtCarRentageCost.Text;

string CarPlateNumber = txtCarPlateNumber.Text;

string CarStatus = "Available!";

{

cn.Open();

cmd.CommandText = "insert into tblCar values('" + CarRegId + "','" + CarType + "','" + CarModel + "','" + CarMaker + "','" + CarColor + "','" + CarRentageCost + "','" + CarPlateNumber + "','" + CarStatus + "')";

cmd.ExecuteNonQuery();

MessageBox.Show("New Car Registration Details successfully saved!!!", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Information);

cn.Close();

btnRefresh();

}

}

}

catch (Exception er)

{

cn.Close();

MessageBox.Show(er.Message, "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private bool isAllEntryFilled()

{

bool isEntryFilled = false;

if ((txtCarRegistrationId.Text == "CAR/RENT/"))

{

MessageBox.Show("Please click refresh button to regenerate car registration id", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboCarType.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please specify the Car Type", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboCarType.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please set the Car Model", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboCarModel.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please set the Car Maker", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

cboCarMaker.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please enter the Car Color", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtCarColor.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please type the Car Rentage Cost ", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtCarRentageCost.Focus();

return isEntryFilled;

}

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

{

MessageBox.Show("Please type the Car Plate Number ", "Car Rental App.", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

txtCarPlateNumber.Focus();

return isEntryFilled;

}

else

return true;

}

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

{

cmd.Connection = cn;

txtCarRegistrationId.Text = genCarRegistrationId();

txtCarRegistrationId.Enabled = false;

}

private string genCarRegistrationId()

{

string CarId = "CAR/RENT/";

try

{

int rowCount = 0;

cn.Open();

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

dr = cmd.ExecuteReader();

while (dr.Read())

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

rowCount++;

if (rowCount < 10)

{

CarId = CarId + "000" + rowCount;

}

else if (rowCount < 100)

{

CarId = CarId + "00" + rowCount;

}

else if (rowCount < 1000)

{

CarId = CarId + "0" + rowCount;

}

else if (rowCount < 10000)

{

CarId = CarId + rowCount;

}

dr.Close();

cn.Close();

}

catch (Exception Er)

{

dr.Close();

cn.Close();

MessageBox.Show(Er.Message, "Car Rental App", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return CarId;

}

}

}