using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class Book

{

public Book(int idbook, string name, int idcategory, string language, string author, string translator, string company, int count )

{

this.IdBook = idbook;

this.Name = name;

this.CategoryID = idcategory;

this.Language = language;

this.Author = author;

this.Translator = translator;

this.Company = company;

this.Count = count;

}

public Book(DataRow row)

{

this.IdBook = (int)row["idbook"];

this.Name = row["name"].ToString();

this.CategoryID = (int)row["idcategory"];

this.Language = row["language"].ToString();

this.Author = row["author"].ToString();

this.Translator = row["translator"].ToString();

this.Company = row["company"].ToString();

this.Count = (int)row["count"];

}

private string name;

private int idbook;

private int idcategory;

private string language;

private string author;

private string translator;

private string company;

private int count;

public int IdBook

{

get { return idbook; }

set { idbook = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public int CategoryID

{

get { return idcategory; }

set { idcategory = value; }

}

public string Language

{

get { return language; }

set { language = value; }

}

public string Author

{

get { return author; }

set { author = value; }

}

public string Translator

{

get { return translator; }

set { translator = value; }

}

public string Company

{

get { return company; }

set { company = value; }

}

public int Count

{

get { return count; }

set { count = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class BookDAO

{

private static BookDAO instance;

public static BookDAO Instance

{

get { if (instance == null) instance = new BookDAO(); return BookDAO.instance; }

private set { BookDAO.instance = value; }

}

private BookDAO() { }

public List<Book> GetListBookByCategoryID(int id)

{

List<Book> list = new List<Book>();

string query = "select \* from TableBook where idcategory = " + id;

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Book book = new Book(item);

list.Add(book);

}

return list;

}

public List<Book> GetListBook()

{

List<Book> list = new List<Book>();

string query = "select \* from TableBook";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Book book = new Book(item);

list.Add(book);

}

return list;

}

public List<Book> SearchBookByName(string name)

{

List<Book> list = new List<Book>();

string query = string.Format("select \* from TableBook where name like N'%{0}%'", name);

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Book book = new Book(item);

list.Add(book);

}

return list;

}

public Book GetListBookByBookID(int id)

{

Book book = null;

string query = "select \* from dbo.TableBook where idbook = " + id;

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

book = new Book(item);

return book;

}

return book;

}

public bool InsertBook(string name, int id, string language, string author, string translator, string company, int count)

{

string query = string.Format("insert dbo.TableBook (name, idcategory, language, author, translator, company, count) values (N'{0}', {1}, N'{2}', N'{3}', N'{4}', N'{5}', {6})", name, id, language, author, translator, company, count);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateBook(int idbook, string name, int id, string language, string author, string translator, string company, int count)

{

string query = string.Format("update dbo.TableBook set name = N'{0}', idcategory = {1}, language = N'{2}', author = N'{3}', translator = N'{4}', company = N'{5}', count = {6} where idbook = {7}", name, id, language, author, translator, company, count, idbook);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool DeleteBook(int idbook)

{

BorrowDAO.Instance.DeleteBorrowByBookID(idbook);

string query = string.Format("delete from dbo.TableBook where idbook = {0}", idbook);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class Account

{

public Account(string displayname, string username, string password = null)

{

this.Displayname = displayname;

this.Username = username;

this.Password = password;

}

public Account(DataRow row)

{

this.Displayname = row["displayname"].ToString();

this.Username = row["username"].ToString();

this.Password = row["password"].ToString();

}

private string displayname;

private string username;

private string password;

public string Displayname

{

get { return displayname; }

set { displayname = value; }

}

public string Username

{

get { return username; }

set { username = value; }

}

public string Password

{

get { return password; }

set { password = value; }

}

}

}

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 QLTV1

{

class AccountDAO

{

private static AccountDAO instance;

public static AccountDAO Instance

{

get { if (instance == null) instance = new AccountDAO(); return instance; }

private set { instance = value; }

}

private AccountDAO() { }

public bool Login(string username, string password)

{

string query = "USP\_Login @username , @password";

DataTable result = DataProvider.Instance.ExecuteQuery(query, new object[] { username, password });

return result.Rows.Count > 0;

}

public List<Account> GetListAccount()

{

List<Account> list = new List<Account>();

string query = "select \* from TableAccount";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Account account = new Account(item);

list.Add(account);

}

return list;

}

public bool InsertAccount(string displayname, string username, string password)

{

string query = string.Format("insert dbo.TableAccount (displayname, username, password) values (N'{0}', N'{1}', N'{2}')", displayname, username, password);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateAccount(string displayname, string username)

{

string query = string.Format("update dbo.TableAccount set displayname = N'{0}' where username = N'{1}'", displayname, username);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool DeleteAccount(string username)

{

string query = string.Format("delete from dbo.TableAccount where username = '{0}'", username);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdatePassword(string password, string username)

{

string query = string.Format("update dbo.TableAccount set password = N'{0}' where username = N'{1}'", password, username);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class Borrow

{

public Borrow(int id, int idBook, int idReader, DateTime dateBorrow, DateTime dateReturn, int countBorrow)

{

this.iD = id;

this.iDBook = idBook;

this.iDReader = idReader;

this.DateBorrow = dateBorrow;

this.DateReturn = dateReturn;

this.CountBorrow = countBorrow;

}

public Borrow(DataRow row)

{

this.iD = (int)row["id"];

this.iDBook = (int)row["idBook"];

this.iDReader = (int)row["idReader"];

this.DateBorrow = (DateTime)row["dateBorrow"];

this.DateReturn = (DateTime)row["dateReturn"];

this.CountBorrow = (int)row["countBorrow"];

}

private int id;

private int idBook;

private int idReader;

private DateTime dateBorrow;

private DateTime dateReturn;

private int countBorrow;

public int iD

{

get { return id; }

set { id = value; }

}

public int iDBook

{

get { return idBook; }

set { idBook = value; }

}

public int iDReader

{

get { return idReader; }

set { idReader = value; }

}

public DateTime DateBorrow

{

get { return dateBorrow; }

set { dateBorrow = value; }

}

public DateTime DateReturn

{

get { return dateReturn; }

set { dateReturn = value; }

}

public int CountBorrow

{

get { return countBorrow; }

set { countBorrow = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class BorrowDAO

{

private static BorrowDAO instance;

public static BorrowDAO Instance

{

get { if (instance == null) instance = new BorrowDAO(); return BorrowDAO.instance; }

private set { BorrowDAO.instance = value; }

}

private BorrowDAO() { }

public List<Borrow> GetListBorrow()

{

List<Borrow> list = new List<Borrow>();

string query = "select \* from dbo.TableBorrow";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Borrow borrow = new Borrow(item);

list.Add(borrow);

}

return list;

}

public List<Borrow> SearchBorrowByBookID(int idBook)

{

List<Borrow> list = new List<Borrow>();

string query = string.Format("select \* from TableBorrow where idBook like N'%{0}%'", idBook);

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Borrow borrow = new Borrow(item);

list.Add(borrow);

}

return list;

}

public List<Borrow> SearchBorrowByReaderID(int idReader)

{

List<Borrow> list = new List<Borrow>();

string query = string.Format("select \* from TableBorrow where idReader like N'%{0}%'", idReader);

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Borrow borrow = new Borrow(item);

list.Add(borrow);

}

return list;

}

public bool InsertBorrow(int idBook, int idReader, DateTime dateBorrow, DateTime dateReturn, int countBorrow)

{

string query = string.Format("insert dbo.TableBorrow (idBook, idReader, dateBorrow, dateReturn, countBorrow) values ({0}, {1}, N'{2}', N'{3}', {4})", idBook, idReader, dateBorrow.ToString("yyyy-MM-dd"), dateReturn.ToString("yyyy-MM-dd"), countBorrow);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateBorrow(int id, int idBook, int idReader, DateTime dateBorrow, DateTime dateReturn, int countBorrow)

{

string query = string.Format("update dbo.TableBorrow set idBook = {0}, idReader = {1}, dateBorrow = N'{2}', dateReturn = N'{3}', countBorrow = {4} where id = {5}", idBook, idReader, dateBorrow.ToString("yyyy-MM-dd"), dateReturn.ToString("yyyy-MM-dd"), countBorrow, id);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public void DeleteBorrowByBookID(int idbook)

{

DataProvider.Instance.ExecuteQuery("delete from dbo.TableBorrow where idBook = " + idbook);

}

public void DeleteBorrowByReaderID(int idreader)

{

DataProvider.Instance.ExecuteQuery("delete from dbo.TableBorrow where idReader = " + idreader);

}

public bool DeleteBorrow(int id)

{

string query = string.Format("delete from dbo.TableBorrow where id = {0}", id);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class Category

{

public Category(int id, string name)

{

this.ID = id;

this.Name = name;

}

public Category(DataRow row)

{

this.ID = (int)row["id"];

this.Name = row["name"].ToString();

}

private string name;

private int id;

public int ID

{

get { return id; }

set { id = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class CategoryDAO

{

private static CategoryDAO instance;

public static CategoryDAO Instance

{

get { if (instance == null) instance = new CategoryDAO(); return CategoryDAO.instance; }

private set { CategoryDAO.instance = value; }

}

private CategoryDAO() { }

public List<Category> GetListCategory()

{

List<Category> list = new List<Category>();

string query = "select \* from dbo.TableBookCategory";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Category category = new Category(item);

list.Add(category);

}

return list;

}

public Category GetCategoryByID(int id)

{

Category category = null;

string query = "select \* from dbo.TableBookCategory where id = " + id;

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

category = new Category(item);

return category;

}

return category;

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

using System.Threading.Tasks;

using System.Linq;

namespace QLTV1

{

public class DataProvider

{

private static DataProvider instance; // Ctrl R E

public static DataProvider Instance

{

get { if ( instance == null ) instance = new DataProvider(); return DataProvider.instance; }

private set { DataProvider.instance = value; }

}

private DataProvider() { }

private string connectionSTR = @"Data Source=DESKTOP-CN4Q9H1\BOMELLO;Initial Catalog=QLTV1;Integrated Security=True";

public DataTable ExecuteQuery(string query, object[] parameter = null)

{

DataTable data = new DataTable();

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(data);

connection.Close();

}

return data;

}

public int ExecuteNonQuery(string query, object[] parameter = null)

{

int data = 0;

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

data = command.ExecuteNonQuery();

connection.Close();

}

return data;

}

public object ExecuteScalar(string query, object[] parameter = null)

{

object data = 0;

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

data = command.ExecuteScalar();

connection.Close();

}

return data;

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class Reader

{

public Reader (int idreader, string name, int idcard, string address, string email, int phone, DateTime cardissuedate, DateTime expirationdate, string typecard)

{

this.IdReader = idreader;

this.Name = name;

this.IDCard = idcard;

this.Address = address;

this.Email = email;

this.Phone = phone;

this.Cardissuedate = cardissuedate;

this.Expirationdate = expirationdate;

this.Typecard = typecard;

}

public Reader(DataRow row)

{

this.IdReader = (int)row["idreader"];

this.Name = row["name"].ToString();

this.IDCard = (int)row["idcard"];

this.Address = row["address"].ToString();

this.Email = row["email"].ToString();

this.Phone = (int)row["phone"];

this.Cardissuedate = (DateTime)row["cardissuedate"];

this.Expirationdate = (DateTime)row["expirationdate"];

this.Typecard = row["typecard"].ToString();

}

private string name;

private int idreader;

private int idcard;

private string address;

private string email;

private int phone;

private DateTime cardissuedate;

private DateTime expirationdate;

private string typecard;

public int IdReader

{

get { return idreader; }

set { idreader = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public int IDCard

{

get { return idcard; }

set { idcard = value; }

}

public string Address

{

get { return address; }

set { address = value; }

}

public string Email

{

get { return email; }

set { email = value; }

}

public int Phone

{

get { return phone; }

set { phone = value; }

}

public DateTime Cardissuedate

{

get { return cardissuedate; }

set { cardissuedate = value; }

}

public DateTime Expirationdate

{

get { return expirationdate; }

set { expirationdate = value; }

}

public string Typecard

{

get { return typecard; }

set { typecard = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

namespace QLTV1

{

public class ReaderDAO

{

private static ReaderDAO instance;

public static ReaderDAO Instance

{

get { if (instance == null) instance = new ReaderDAO(); return ReaderDAO.instance; }

private set { ReaderDAO.instance = value; }

}

private ReaderDAO() { }

public List<Reader> GetListReader()

{

List<Reader> list = new List<Reader>();

string query = "select \* from dbo.TableReader";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Reader reader = new Reader(item);

list.Add(reader);

}

return list;

}

public List<Reader> SearchReaderByName(string name)

{

List<Reader> list = new List<Reader>();

string query = string.Format("select \* from TableReader where name like N'%{0}%'", name);

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Reader reader = new Reader(item);

list.Add(reader);

}

return list;

}

public Reader GetListReaderByReaderID(int id)

{

Reader reader = null;

string query = "select \* from dbo.TableReader where idreader = " + id;

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

reader = new Reader(item);

return reader;

}

return reader;

}

public bool InsertReader(string name, int idcard, string address, string email, int phone, DateTime cardissuedate, DateTime expirationdate, string typecard)

{

string query = string.Format("insert dbo.TableReader (name, idcard, address, email, phone, cardissuedate, expirationdate, typecard) values (N'{0}', {1}, N'{2}', N'{3}', {4}, N'{5}', N'{6}', N'{7}')", name, idcard, address, email, phone, cardissuedate.ToString("yyyy-MM-dd"), expirationdate.ToString("yyyy-MM-dd"), typecard);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool UpdateReader(int idreader, string name, int idcard, string address, string email, int phone, DateTime cardissuedate, DateTime expirationdate, string typecard)

{

string query = string.Format("update dbo.TableReader set name = N'{0}', idcard = {1}, address = N'{2}', email = N'{3}', phone = {4}, cardissuedate = N'{5}', expirationdate = N'{6}', typecard = N'{7}' where idreader = {8}", name, idcard, address, email, phone, cardissuedate.ToString("yyyy-MM-dd"), expirationdate.ToString("yyyy-MM-dd"), typecard, idreader);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

public bool DeleteReader(int idreader)

{

BorrowDAO.Instance.DeleteBorrowByReaderID(idreader);

string query = string.Format("delete from dbo.TableReader where idreader = {0}", idreader);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Linq;

namespace QLTV1

{

public partial class fAdmin : Form

{

BindingSource bookList = new BindingSource();

BindingSource readerList = new BindingSource();

BindingSource borrowList = new BindingSource();

BindingSource accountList = new BindingSource();

public fAdmin()

{

InitializeComponent();

dtgvBook.DataSource = bookList;

dtgvReader.DataSource = readerList;

dtgvBorrow.DataSource = borrowList;

dtgvAccount.DataSource = accountList;

LoadCategory();

LoadListBook();

LoadCategoryIntoCombobox(cbBookCategory);

AddBookBinding();

LoadListReader();

AddReaderBinding();

LoadListBorrow();

AddBorrowBinding();

LoadBookIntoCombobox(cbBorrowBook);

LoadReaderIntoCombobox(cbBorrowReader);

LoadListAccount();

AddAccountBinding();

}

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

{

}

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

{

}

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

{

}

void LoadCategory()

{

List<Category> listCategory = CategoryDAO.Instance.GetListCategory();

cbBookCategory.DataSource = listCategory;

cbBookCategory.DisplayMember = "name";

}

void LoadBookListByCategoryID(int id)

{

List<Book> listBook = BookDAO.Instance.GetListBookByCategoryID(id);

}

void AddBookBinding()

{

txbBookID.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "IdBook", true, DataSourceUpdateMode.Never));

txbBookName.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "Name", true, DataSourceUpdateMode.Never));

txbBookLanguage.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "Language", true, DataSourceUpdateMode.Never));

txbBookAuthor.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "Author", true, DataSourceUpdateMode.Never));

txbBookTranslator.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "Translator", true, DataSourceUpdateMode.Never));

txbBookCompany.DataBindings.Add(new Binding("Text", dtgvBook.DataSource, "Company", true, DataSourceUpdateMode.Never));

nmBookCount.DataBindings.Add(new Binding("Value", dtgvBook.DataSource, "Count", true, DataSourceUpdateMode.Never));

}

void AddReaderBinding()

{

txbReaderID.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "IdReader", true, DataSourceUpdateMode.Never));

txbReaderName.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "Name", true, DataSourceUpdateMode.Never));

txbReaderIDCard.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "IDCard", true, DataSourceUpdateMode.Never));

txbReaderAddress.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "Address", true, DataSourceUpdateMode.Never));

txbReaderEmail.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "Email", true, DataSourceUpdateMode.Never));

txbReaderPhone.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "Phone", true, DataSourceUpdateMode.Never));

dtpkReaderFromDate.DataBindings.Add(new Binding("Value", dtgvReader.DataSource, "Cardissuedate", true, DataSourceUpdateMode.Never));

dtpkReaderToDate.DataBindings.Add(new Binding("Value", dtgvReader.DataSource, "Expirationdate", true, DataSourceUpdateMode.Never));

txbReaderTypeCard.DataBindings.Add(new Binding("Text", dtgvReader.DataSource, "Typecard", true, DataSourceUpdateMode.Never));

}

void AddBorrowBinding()

{

txbBorrowID.DataBindings.Add(new Binding("Text", dtgvBorrow.DataSource, "iD", true, DataSourceUpdateMode.Never));

txbBorrowBookID.DataBindings.Add(new Binding("Text", dtgvBorrow.DataSource, "iDBook", true, DataSourceUpdateMode.Never));

txbBorrowReaderID.DataBindings.Add(new Binding("Text", dtgvBorrow.DataSource, "iDReader", true, DataSourceUpdateMode.Never));

dtpkBorrow.DataBindings.Add(new Binding("Value", dtgvBorrow.DataSource, "DateBorrow", true, DataSourceUpdateMode.Never));

dtpkReturn.DataBindings.Add(new Binding("Value", dtgvBorrow.DataSource, "DateReturn", true, DataSourceUpdateMode.Never));

nmBorrowCount.DataBindings.Add(new Binding("Text", dtgvBorrow.DataSource, "CountBorrow", true, DataSourceUpdateMode.Never));

}

void AddAccountBinding()

{

txbDisplayname.DataBindings.Add(new Binding("Text", dtgvAccount.DataSource, "Displayname", true, DataSourceUpdateMode.Never));

txbUsername.DataBindings.Add(new Binding("Text", dtgvAccount.DataSource, "Username", true, DataSourceUpdateMode.Never));

txbPassword.DataBindings.Add(new Binding("Text", dtgvAccount.DataSource, "Password", true, DataSourceUpdateMode.Never));

}

void LoadCategoryIntoCombobox(ComboBox cb)

{

cb.DataSource = CategoryDAO.Instance.GetListCategory();

cb.DisplayMember = "name";

}

void LoadBookIntoCombobox(ComboBox cb)

{

cb.DataSource = BookDAO.Instance.GetListBook();

cb.DisplayMember = "name";

}

void LoadReaderIntoCombobox(ComboBox cb)

{

cb.DataSource = ReaderDAO.Instance.GetListReader();

cb.DisplayMember = "name";

}

void LoadListBook()

{

bookList.DataSource = BookDAO.Instance.GetListBook();

}

void LoadListReader()

{

readerList.DataSource = ReaderDAO.Instance.GetListReader();

}

void LoadListBorrow()

{

borrowList.DataSource = BorrowDAO.Instance.GetListBorrow();

}

void LoadListAccount()

{

accountList.DataSource = AccountDAO.Instance.GetListAccount();

}

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

{

}

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

{

try

{

if (dtgvBook.SelectedCells.Count > 0)

{

int id = (int)dtgvBook.SelectedCells[0].OwningRow.Cells["CategoryID"].Value;

Category category = CategoryDAO.Instance.GetCategoryByID(id);

cbBookCategory.SelectedItem = category;

int index = -1;

int i = 0;

foreach (Category item in cbBookCategory.Items)

{

if (item.ID == category.ID)

{

index = i;

break;

}

i++;

}

cbBookCategory.SelectedIndex = index;

}

}

catch { }

}

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

{

}

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

{

if (dtgvBorrow.SelectedCells.Count > 0)

{

int idbook = Convert.ToInt32(txbBookID.Text);

int id = (int)dtgvBorrow.SelectedCells[0].OwningRow.Cells["iDBook"].Value;

Book book = BookDAO.Instance.GetListBookByBookID(id);

cbBorrowBook.SelectedItem = book;

int index = -1;

int i = 0;

if (book != null)

{

foreach (Book item in cbBorrowBook.Items)

{

if (item.IdBook == book.IdBook)

{

index = i;

break;

}

i++;

}

cbBorrowBook.SelectedIndex = index;

}

else

{

DataProvider.Instance.ExecuteQuery("delete from dbo.TableBorrow where idBook = " + idbook);

}

}

if (dtgvBorrow.SelectedCells.Count > 0)

{

int id = (int)dtgvBorrow.SelectedCells[0].OwningRow.Cells["iDReader"].Value;

Reader reader = ReaderDAO.Instance.GetListReaderByReaderID(id);

cbBorrowReader.SelectedItem = reader;

int index = -1;

int i = 0;

foreach (Reader item in cbBorrowReader.Items)

{

if (item.IdReader == reader.IdReader)

{

index = i;

break;

}

i++;

}

cbBorrowReader.SelectedIndex = index;

}

}

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

{

int id = 0;

ComboBox cb = sender as ComboBox;

if (cb.SelectedItem == null)

return;

Category selected = cb.SelectedItem as Category;

id = selected.ID;

LoadBookListByCategoryID(id);

}

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

{

}

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

{

string name = txbBookName.Text;

int idcategory = (cbBookCategory.SelectedItem as Category).ID;

string language = txbBookLanguage.Text;

string author = txbBookAuthor.Text;

string translator = txbBookTranslator.Text;

string company = txbBookCompany.Text;

int count = (int)nmBookCount.Value;

if (BookDAO.Instance.InsertBook(name, idcategory, language, author, translator, company, count))

{

MessageBox.Show(" Thêm thành công");

LoadListBook();

LoadBookIntoCombobox(cbBorrowBook);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

string name = txbBookName.Text;

int idcategory = (cbBookCategory.SelectedItem as Category).ID;

string language = txbBookLanguage.Text;

string author = txbBookAuthor.Text;

string translator = txbBookTranslator.Text;

string company = txbBookCompany.Text;

int count = (int)nmBookCount.Value;

int idbook = Convert.ToInt32(txbBookID.Text);

if (BookDAO.Instance.UpdateBook(idbook, name, idcategory, language, author, translator, company, count))

{

MessageBox.Show(" Sửa thành công");

LoadListBook();

LoadBookIntoCombobox(cbBorrowBook);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

int idbook = Convert.ToInt32(txbBookID.Text);

if (BookDAO.Instance.DeleteBook(idbook))

{

MessageBox.Show(" Xóa thành công");

LoadListBook();

LoadBookIntoCombobox(cbBorrowBook);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

LoadListBook();

}

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

{

}

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

{

string name = txbReaderName.Text;

int idcard = Convert.ToInt32(txbReaderIDCard.Text);

string address = txbReaderAddress.Text;

string email = txbReaderEmail.Text;

int phone = Convert.ToInt32(txbReaderPhone.Text);

DateTime cardissuedate = dtpkReaderFromDate.Value;

DateTime expirationdate = dtpkReaderToDate.Value;

string typecard = txbReaderTypeCard.Text;

if (ReaderDAO.Instance.InsertReader(name, idcard, address, email, phone, cardissuedate, expirationdate, typecard))

{

MessageBox.Show(" Thêm thành công");

LoadListReader();

LoadReaderIntoCombobox(cbBorrowReader);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

int idreader = Convert.ToInt32(txbReaderID.Text);

string name = txbReaderName.Text;

int idcard = Convert.ToInt32(txbReaderIDCard.Text);

string address = txbReaderAddress.Text;

string email = txbReaderEmail.Text;

int phone = Convert.ToInt32(txbReaderPhone.Text);

DateTime cardissuedate = dtpkReaderFromDate.Value;

DateTime expirationdate = dtpkReaderToDate.Value;

string typecard = txbReaderTypeCard.Text;

if (ReaderDAO.Instance.UpdateReader(idreader, name, idcard, address, email, phone, cardissuedate, expirationdate, typecard))

{

MessageBox.Show(" Sửa thành công");

LoadListReader();

LoadReaderIntoCombobox(cbBorrowReader);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

int idreader = Convert.ToInt32(txbReaderID.Text);

if (ReaderDAO.Instance.DeleteReader(idreader))

{

MessageBox.Show(" Xóa thành công");

LoadListReader();

LoadReaderIntoCombobox(cbBorrowReader);

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

LoadListReader();

}

private void button5\_Click(object sender, EventArgs e)

{

int idBook = (cbBorrowBook.SelectedItem as Book).IdBook;

int idReader = (cbBorrowReader.SelectedItem as Reader).IdReader;

DateTime dateBorrow = dtpkBorrow.Value;

DateTime dateReturn = dtpkReturn.Value;

int countBorrow = (int)nmBorrowCount.Value;

if (BorrowDAO.Instance.InsertBorrow(idBook, idReader, dateBorrow, dateReturn, countBorrow))

{

MessageBox.Show(" Thêm thành công");

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

int idBook = (cbBorrowBook.SelectedItem as Book).IdBook;

int idReader = (cbBorrowReader.SelectedItem as Reader).IdReader;

DateTime dateBorrow = dtpkBorrow.Value;

DateTime dateReturn = dtpkReturn.Value;

int countBorrow = (int)nmBorrowCount.Value;

int id = Convert.ToInt32(txbBorrowID.Text);

if (BorrowDAO.Instance.UpdateBorrow(id, idBook, idReader, dateBorrow, dateReturn, countBorrow))

{

MessageBox.Show(" Sửa thành công");

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

int id = Convert.ToInt32(txbBorrowID.Text);

if (BorrowDAO.Instance.DeleteBorrow(id))

{

MessageBox.Show(" Xóa thành công");

LoadListBorrow();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

LoadListBorrow();

}

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

{

string displayname = txbDisplayname.Text;

string username = txbUsername.Text;

string password = txbPassword.Text;

if (AccountDAO.Instance.InsertAccount(displayname, username, password))

{

MessageBox.Show(" Thêm thành công");

LoadListAccount();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

string displayname = txbDisplayname.Text;

string username = txbUsername.Text;

if (AccountDAO.Instance.UpdateAccount(displayname, username))

{

MessageBox.Show(" Sửa thành công");

LoadListAccount();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

string password = txbNewPass.Text;

string username = txbUsername.Text;

string renewpass = txbReNewPass.Text;

if (!password.Equals(renewpass))

{

MessageBox.Show(" Nhập lại mật khẩu đúng với mật khẩu mới");

}

else

{

if (AccountDAO.Instance.UpdatePassword(password, username))

{

MessageBox.Show(" Sửa thành công");

LoadListAccount();

txbNewPass.Text = null;

txbReNewPass.Text = null;

}

else

{

MessageBox.Show(" Lỗi");

}

}

}

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

{

string username = txbUsername.Text;

if (AccountDAO.Instance.DeleteAccount(username))

{

MessageBox.Show(" Xóa thành công");

LoadListAccount();

}

else

{

MessageBox.Show(" Lỗi");

}

}

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

{

LoadListAccount();

}

List<Book> SearchBookByName(string name)

{

List<Book> listBook = BookDAO.Instance.SearchBookByName(name);

return listBook;

}

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

{

bookList.DataSource = SearchBookByName(txbSearchBook.Text);

}

List<Reader> SearchReaderByName(string name)

{

List<Reader> listReader = ReaderDAO.Instance.SearchReaderByName(name);

return listReader;

}

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

{

readerList.DataSource = SearchReaderByName(txbSearchReader.Text);

}

List<Borrow> SearchBorrowByBookID(int idBook)

{

List<Borrow> listBorrow = BorrowDAO.Instance.SearchBorrowByBookID(idBook);

return listBorrow;

}

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

{

borrowList.DataSource = SearchBorrowByBookID(Convert.ToInt32(txbSearchBorrowBook.Text));

}

List<Borrow> SearchBorrowByReaderID(int idReader)

{

List<Borrow> listBorrow = BorrowDAO.Instance.SearchBorrowByReaderID(idReader);

return listBorrow;

}

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

{

borrowList.DataSource = SearchBorrowByReaderID(Convert.ToInt32(txbSearchBorrowReader.Text));

}

}

}

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 QLTV1

{

public partial class fLogin : Form

{

public fLogin()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string username = txbUserName.Text;

string password = txbPassWord.Text;

if (Login(username, password))

{

fTableManager f = new fTableManager();

this.Hide();

f.ShowDialog();

this.Show();

}

else

{

MessageBox.Show(" Sai tài khoản hoặc mật khẩu");

}

}

bool Login(string username, string password)

{

return AccountDAO.Instance.Login(username, password);

}

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

{

Application.Exit();

}

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

{

if (MessageBox.Show(" THOÁT CHƯƠNG TRÌNH ?", "Thông báo", MessageBoxButtons.OKCancel) != System.Windows.Forms.DialogResult.OK)

{

e.Cancel = true;

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace QLTV1

{

public partial class fTableManager : Form

{

public fTableManager()

{

InitializeComponent();

}

private void menuStrip2\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

fAdmin f = new fAdmin();

f.ShowDialog();

}

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

{

}

}

}