Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут ім. Ігоря Сікорського”

Кафедра автоматизації проектування енергетичних процесів і систем

Звіт

з лабораторної роботи №2

з дисципліни:

«Проектування систем з розподіленими бащами даних»

Виконали:

група ТР-01мп  
Кривошеев Е.В.

Лєсніченко Р.А.

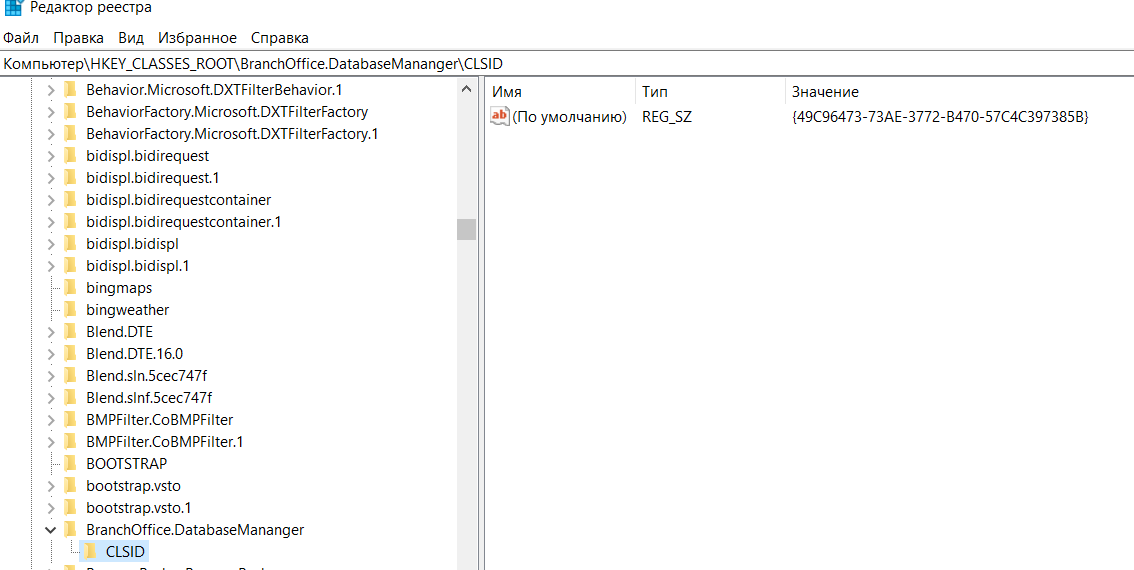
Перевірив:

Варава І.А.

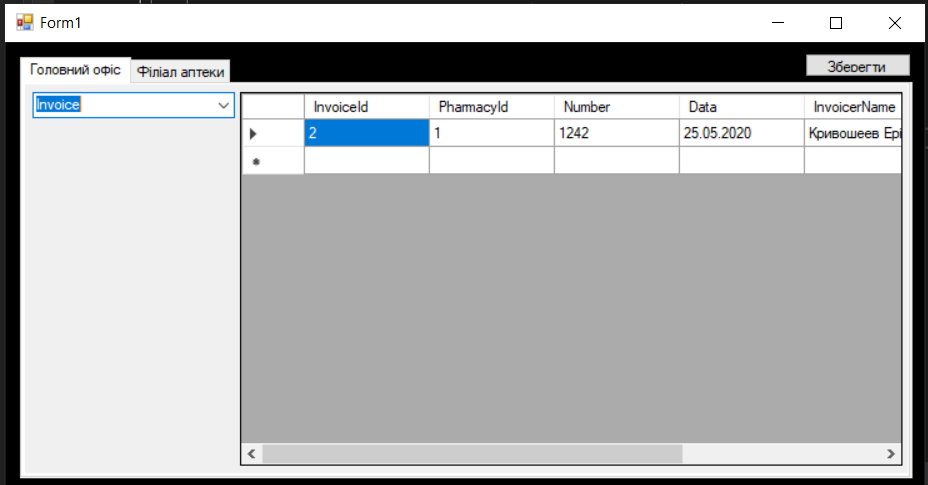
Київ – 2020

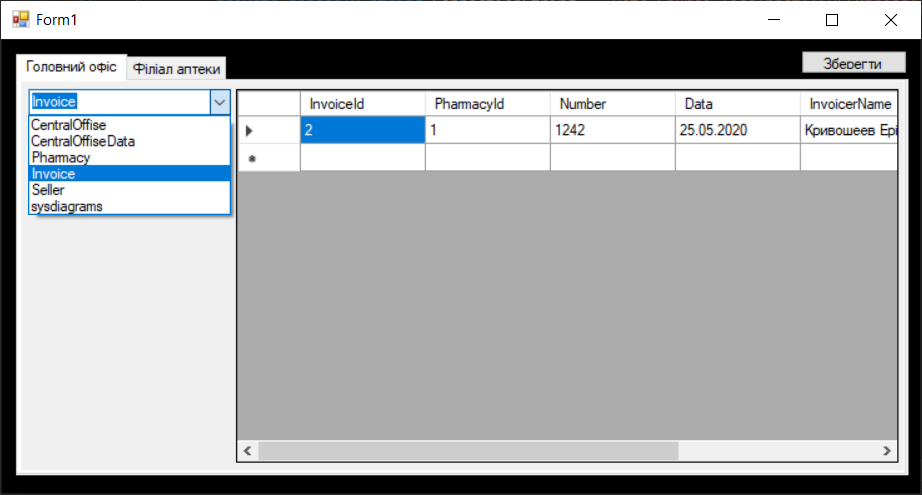
Завдання: Триланкова система СОM/DCOM

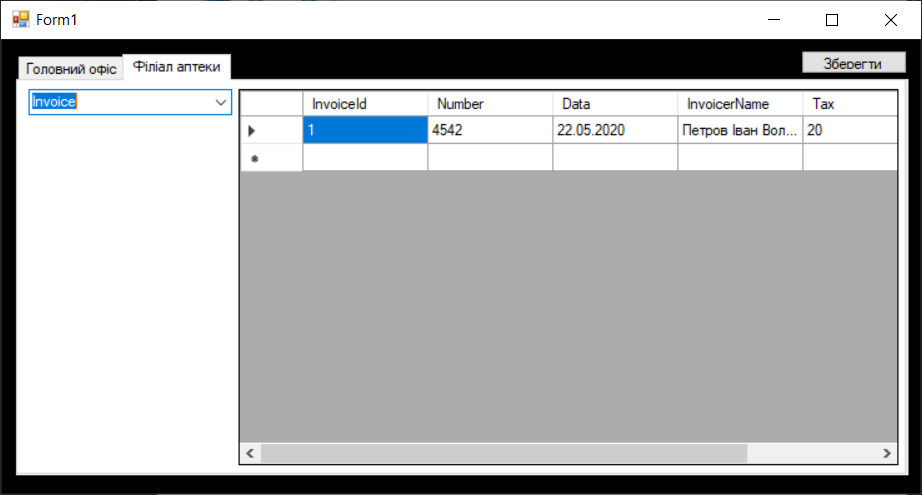
Скріншоти зареєстрованих серверних com застосунків



Скріншоти клієнта:







Код:

public class DatabaseMananger: IDatabaseManager

{

private readonly string ConnectionString = "Data Source=localhost\\SQLExpress;Initial Catalog=FilialDB;Integrated Security=SSPI;";

public DataTable GetTable(string query)

{

var conection = new SqlConnection(ConnectionString);

conection.Open();

var command = new SqlCommand(query, conection);

var r = command.ExecuteReader();

var dataTable = new DataTable();

dataTable.Load(r);

conection.Close();

return dataTable;

}

public List<string> GetTables()

{

var resutlList = new List<string>();

var conection = new SqlConnection(ConnectionString);

conection.Open();

var command = new SqlCommand("SELECT \* FROM sys.tables", conection);

SqlDataReader r = command.ExecuteReader();

while (r.Read())

{

resutlList.Add(r.GetString(0));

}

conection.Close();

return resutlList;

}

}

}

[Guid("E637D6B1-41BA-4A70-952E-9A3D0477044A")]

public interface IDatabaseManager

{

DataTable GetTable(string query);

List<string> GetTables();

}

public partial class Form1 : Form

{

Type \_headOfficeType = Type.GetTypeFromProgID("HeadOffice.DatabaseMananger");

Type \_branchOfficeType = Type.GetTypeFromProgID("BranchOffice.DatabaseMananger");

object \_headOfficeComObject;

object \_branchOfficeComObject;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

if (\_headOfficeType != null)

{

\_headOfficeComObject = Activator.CreateInstance(\_headOfficeType);

if (\_headOfficeComObject != null)

{

var mehtodInvoke = \_headOfficeType.GetMethod("GetTables");

List<string> ls = (List<string>)mehtodInvoke.Invoke(\_headOfficeComObject, new object[] { });

foreach (var item in ls)

{

headOfficeTablesCombobox.Items.Add(item);

}

headOfficeTablesCombobox.SelectedValueChanged += HeadOfficeTableSelected;

}

}

if (\_branchOfficeType != null)

{

\_branchOfficeComObject = Activator.CreateInstance(\_branchOfficeType);

if (\_branchOfficeComObject != null)

{

var mehtodInvoke = \_branchOfficeType.GetMethod("GetTables");

List<string> ls = (List<string>)mehtodInvoke.Invoke(\_branchOfficeComObject, new object[] { });

foreach (var item in ls)

{

branchOfficeTablesCombobox.Items.Add(item);

}

headOfficeTablesCombobox.SelectedValueChanged += BranchOfficeTableSelected;

}

}

}

private void HeadOfficeTableSelected(object sender, EventArgs e)

{

var mi = \_headOfficeType.GetMethod("GetTable");

DataTable dt = (DataTable)mi.Invoke(\_headOfficeComObject, new object[] { "SELECT \* FROM " + headOfficeTablesCombobox.Text });

headOfficeDataGridView.Columns.Clear();

headOfficeDataGridView.DataSource = dt;

headOfficeDataGridView.Refresh();

}

private void BranchOfficeTableSelected(object sender, EventArgs e)

{

var mi = \_branchOfficeType.GetMethod("GetTable");

DataTable dt = (DataTable)mi.Invoke(\_branchOfficeComObject, new object[] { "SELECT \* FROM " + branchOfficeTablesCombobox.Text });

branchOfifceDataGridView.Columns.Clear();

branchOfifceDataGridView.DataSource = dt;

branchOfifceDataGridView.Refresh();

}

}