**Лабораторна робота №1**

**Тема: Система контролю версій GIT.**

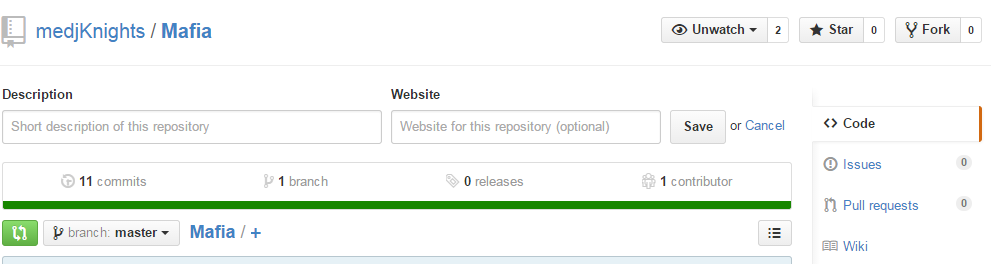
**Мета: Навчитись працювати у git, trello та на сайті github.com.**

**Хід роботи**

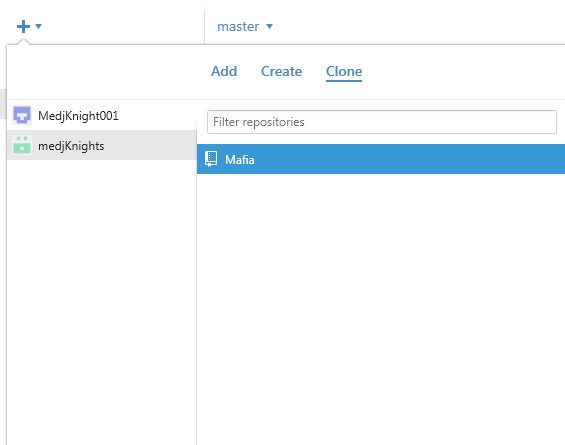
Першим кроком було встановлення git з офіційного сайту:

[**http://github.com/downloads**](http://github.com/downloads)

Після установки тім-лідер створив репозиторій на github.com. Відповідно наступним кроком є реєстрація на сайті та вхід у організацію даного проекту та знаходження репозиторію:

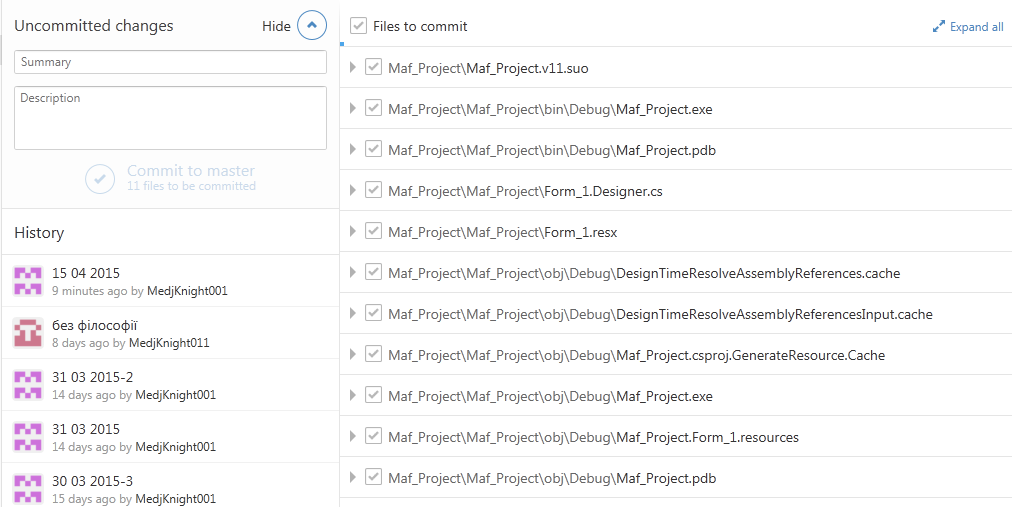


Після входження у організацію заходимо у додаток GitHub та копіюємо репозиторій організації на свій комп’ютер, тобто створюємо локальний репозиторій:

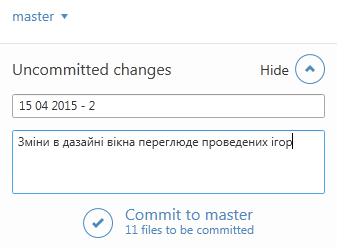


Наступним кроком буде саме написання коду програми, його коміти та відправлення на сервер.

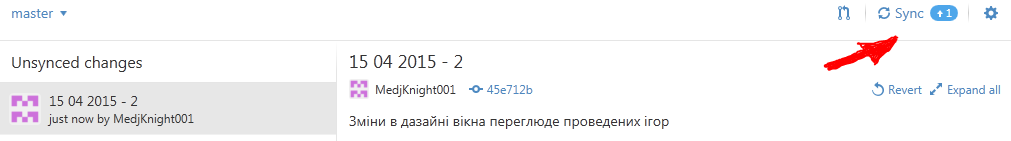
Для того щоб подивитись зміни, які були внесені ми можемо відкрити відповідне вікно додатку:



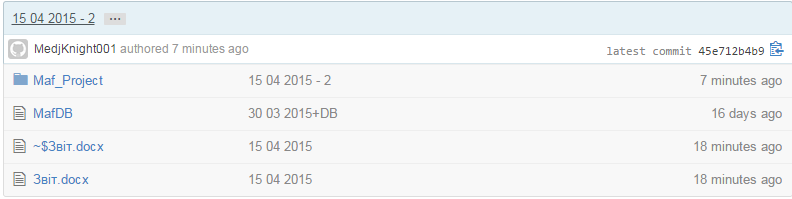
Щоб зафіксувати зміни введемо назву коміту та його опис й натиснемо “Commit to master”:



Після цього проведемо синхронізацію з сервером:



На даному рисунку зображені останні коміти проекту:

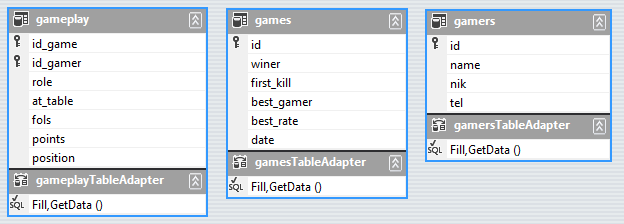


Завданням на лабораторну роботу було створити репозиторій на GitHub, координувати роботу команди та виконувати свою частину кодування.

Моїм завданням було розробити базу даних проекту, дизайн проекту для зручної взаємодії з користувачем та написати свою частину коду для взаємодії інтерфейсу із базою.

Темою нашого проекту є «Автоматизована інформаційна система ведення обліку ігор в «мафію» для клубів міста Хмельницький».

База даних проекту створювалась на MS SQLServer 2012 та включає у собі такі таблиці:

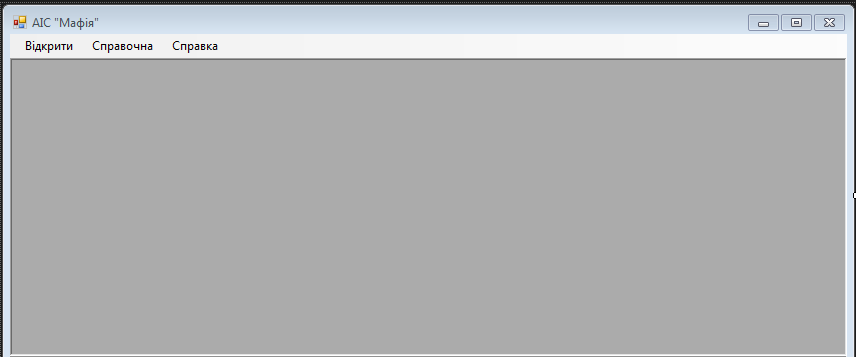


Таблиця «gamers» включає у себе інформацію про гравців.

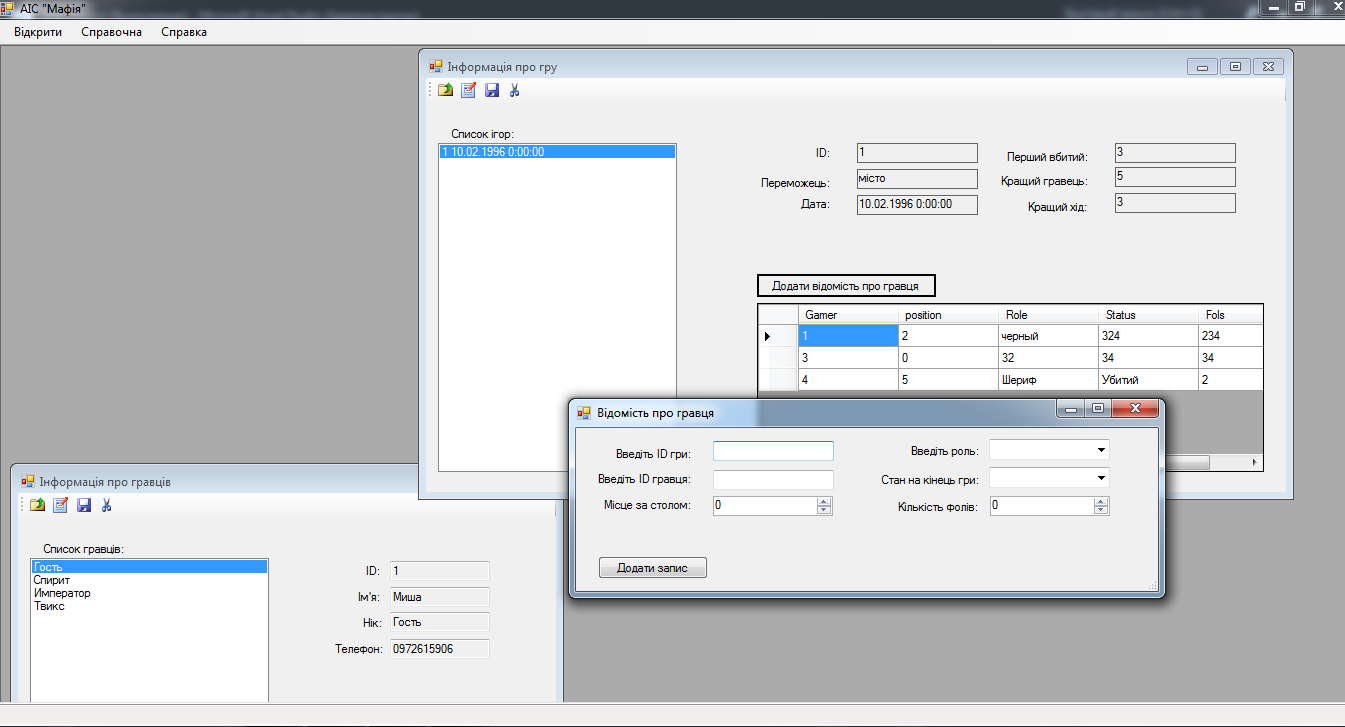
Таблиця «games» включає інформацію про гру, що вводиться по її закінченню.

Таблиця «gameplay» вказує на взаємодію гравців у кожній грі, їх роль, позицію та рейтинг. По сумі рейтингу з цієї таблиці складається всезагальний рейтинг гравця.

Далі був розроблений інтерфейс головного вікна:



Було вирішено, що уся інформація відображатиметься у окремих вікнах, вкладених у головне вікно програми таким чином:



Код вікна для роботи з інформацією про гравців:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

using System.Data.OleDb;

using System.Data.Common;

using System.Threading;

using System.Data.SqlClient;

namespace Maf\_Project

{

public partial class Form\_2 : Form

{

DataSet dsEmployee = new DataSet();

DataSetTableAdapters.gamersTableAdapter dagamers = new Maf\_Project.DataSetTableAdapters.gamersTableAdapter();

BindingManagerBase bmEmployee;

public void EmployeeFill()

{

dagamers.Fill(dsEmployee.gamers);

AddListBoxEmployeeDataSource();

textBox1.DataBindings.Add("Text", dsEmployee, "gamers.id");

textBox2.DataBindings.Add("Text", dsEmployee, "gamers.name");

textBox3.DataBindings.Add("Text", dsEmployee, "gamers.nik");

textBox4.DataBindings.Add("Text", dsEmployee, "gamers.tel");

}

public Form\_2()

{

InitializeComponent();

bmEmployee = this.BindingContext[dsEmployee,"gamers"];

}

private void AddListBoxEmployeeDataSource()

{

listBoxEmployee.DataSource = this.dsEmployee;

listBoxEmployee.DisplayMember = "gamers.nik";

}

private void Form\_1\_Load(object sender, EventArgs e)

{

EmployeeFill();

textBox1.ReadOnly = true;

textBox2.ReadOnly = true;

textBox3.ReadOnly = true;

textBox4.ReadOnly = true;

}

public void Undo()

{

bmEmployee.EndCurrentEdit();

dsEmployee.gamers.RejectChanges();

}

public void New()

{

DataRow rowEmployee = this.dsEmployee.gamers.NewgamersRow();

dsEmployee.gamers.Rows.Add(rowEmployee);

int pos = this.dsEmployee.gamers.Rows.Count - 1;

this.BindingContext[dsEmployee, "gamers"].Position = pos;

Edit();

}

public void Edit()

{

textBox2.ReadOnly = false;

textBox3.ReadOnly = false;

textBox4.ReadOnly = false;

listBoxEmployee.Enabled = false;

}

public void Save()

{

bmEmployee.EndCurrentEdit();

DataSet.gamersDataTable ds2 = (DataSet.gamersDataTable)dsEmployee.gamers.GetChanges(DataRowState.Added);

if (ds2 != null)

try

{

dagamers.Update(ds2);

ds2.Dispose();

dsEmployee.gamers.AcceptChanges();

}

catch (Exception x)

{

string mes2 = x.Message;

MessageBox.Show("Ошибка вставки записи в базу данных 'Podatku' " + mes2, "Предупреждение");

this.dsEmployee.gamers.RejectChanges();

}

DataSet.gamersDataTable ds1 = (DataSet.gamersDataTable)dsEmployee.gamers.GetChanges(DataRowState.Modified);

if (ds1 != null)

try

{

this.dagamers.Update(ds1);

ds1.Dispose();

dsEmployee.gamers.AcceptChanges();

MessageBox.Show("Збережено");

}

catch (Exception x)

{

string mes = x.Message;

MessageBox.Show("Ошибка обновления базы данных 'Gamers' " + mes, "Предупреждение");

this.dsEmployee.gamers.RejectChanges();

}

textBox2.ReadOnly = true;

textBox3.ReadOnly = true;

textBox4.ReadOnly = true;

listBoxEmployee.Enabled = true;

}

public void Remove()

{

int pos = -1;

pos = this.BindingContext[dsEmployee, "gamers"].Position;

DialogResult result = MessageBox.Show("Видалити данні \n" + "по "+ textBox2.Text + " ?","Попередження!",

MessageBoxButtons.YesNo,

MessageBoxIcon.Warning,

MessageBoxDefaultButton.Button2);

switch (result)

{

case DialogResult.Yes:

{

this.dsEmployee.gamers.Rows[pos].Delete();

if (this.dsEmployee.gamers.GetChanges(DataRowState.Deleted) != null)

{

try

{

this.dagamers.Update(dsEmployee.gamers);

this.dsEmployee.gamers.AcceptChanges();

}

catch (Exception x)

{

string er = x.Message.ToString();

MessageBox.Show("Ошибка удаления записи в базе данных 'Gamers' " + er, "Предупреждение");

this.dsEmployee.gamers.RejectChanges();

}

}

MessageBox.Show("Удаление данных");

break;

}

case DialogResult.No:

{

this.dsEmployee.gamers.RejectChanges();

MessageBox.Show("Отмена удаления данных");

break;

}

}

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

New();

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

Edit();

}

private void toolStripButton4\_Click(object sender, EventArgs e)

{

Save();

}

private void toolStripButton5\_Click(object sender, EventArgs e)

{

Remove();

}

}

}

Код вікна для роботи з інформацією про ігри:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

using System.Data.OleDb;

using System.Data.Common;

using System.Threading;

using System.Data.SqlClient;

namespace Maf\_Project

{

public partial class Form\_1 : Form

{

DataSet dsEmployee = new DataSet();

DataSetTableAdapters.gamesTableAdapter dagames = new Maf\_Project.DataSetTableAdapters.gamesTableAdapter();

DataSetTableAdapters.gamersTableAdapter dagamers = new Maf\_Project.DataSetTableAdapters.gamersTableAdapter();

DataSetTableAdapters.gameplayTableAdapter dagameplay = new Maf\_Project.DataSetTableAdapters.gameplayTableAdapter();

BindingManagerBase bmEmployee;

BindingManagerBase bmEmployee2;

public void EmployeeFill()

{

dagames.Fill(dsEmployee.games);

AddColumsFullName();

AddListBoxEmployeeDataSource();

textBox1.DataBindings.Add("Text", dsEmployee, "games.id");

textBox2.DataBindings.Add("Text", dsEmployee, "games.winer");

textBox3.DataBindings.Add("Text", dsEmployee, "games.first\_kill");

textBox4.DataBindings.Add("Text", dsEmployee, "games.best\_gаmer");

textBox5.DataBindings.Add("Text", dsEmployee, "games.best\_rate");

textBox6.DataBindings.Add("Text", dsEmployee, "games.date");

}

public Form\_1()

{

InitializeComponent();

bmEmployee = this.BindingContext[dsEmployee,"games"];

bmEmployee2 = this.BindingContext[dsEmployee, "gameplay"];

}

private void AddColumsFullName()

{

dsEmployee.games.Columns.Add("FullName", typeof(string),

"id+' '+date");

}

private void AddListBoxEmployeeDataSource()

{

listBoxEmployee.DataSource = this.dsEmployee;

listBoxEmployee.DisplayMember = "games.FullName";

}

public void getHistory()

{

SqlConnection c = new SqlConnection(Properties.Settings.Default.MafiaConnectionString);

SqlCommand query = new SqlCommand("SELECT id\_gamer as Gamer, position as position, role as Role, at\_table as Status, fols as Fols FROM gameplay WHERE id\_game=" + textBox1.Text, c);

c.Open();

ArrayList w = new ArrayList(0);

SqlDataReader reader;

reader = query.ExecuteReader();

// зчитування даних з бази в масив

foreach (DbDataRecord t in reader)

{

w.Add(t);

}

c.Close();

dataGridView1.DataBindings.Clear();

// занесення даних з масива в DataGrid View

dataGridView1.DataSource = w;

}

private void Form\_1\_Load(object sender, EventArgs e)

{

EmployeeFill();

textBox1.ReadOnly = true;

textBox2.ReadOnly = true;

textBox3.ReadOnly = true;

textBox4.ReadOnly = true;

textBox5.ReadOnly = true;

textBox6.ReadOnly = true;

//comboBox1.DataBindings.Add("SelectedValue", dsEmployee, "gamers.ID");

}

public void Undo()

{

bmEmployee.EndCurrentEdit();

dsEmployee.games.RejectChanges();

}

public void New()

{

DataRow rowEmployee = this.dsEmployee.games.NewgamesRow();

rowEmployee["winer"] = "default";

rowEmployee["first\_kill"] = "0";

rowEmployee["best\_gamer"] = "0";

rowEmployee["best\_rate"] = "0";

rowEmployee["date"] = DateTime.Now.ToString();

dsEmployee.games.Rows.Add(rowEmployee);

int pos = this.dsEmployee.games.Rows.Count - 1;

this.BindingContext[dsEmployee, "games"].Position = pos;

Edit();

}

public void Edit()

{

//textBox1.ReadOnly = false;

textBox2.ReadOnly = false;

textBox3.ReadOnly = false;

textBox4.ReadOnly = false;

textBox5.ReadOnly = false;

textBox6.ReadOnly = false;

listBoxEmployee.Enabled = false;

//MessageBox.Show("Edit");

}

public void Save()

{

bmEmployee.EndCurrentEdit();

DataSet.gamesDataTable ds2 = (DataSet.gamesDataTable)dsEmployee.games.GetChanges(DataRowState.Added);

if (ds2 != null)

try

{

dagames.Update(ds2);

ds2.Dispose();

dsEmployee.games.AcceptChanges();

}

catch (Exception x)

{

string mes2 = x.Message;

MessageBox.Show("Ошибка вставки записи в базу данных 'Podatku' " + mes2, "Предупреждение");

this.dsEmployee.games.RejectChanges();

}

DataSet.gamesDataTable ds1 = (DataSet.gamesDataTable)dsEmployee.games.GetChanges(DataRowState.Modified);

if (ds1 != null)

try

{

this.dagames.Update(ds1);

ds1.Dispose();

dsEmployee.games.AcceptChanges();

MessageBox.Show("Збережено");

}

catch (Exception x)

{

string mes = x.Message;

MessageBox.Show("Ошибка обновления базы данных 'Podatku' " + mes, "Предупреждение");

this.dsEmployee.games.RejectChanges();

}

//MessageBox.Show("Save");

textBox1.ReadOnly = true;

textBox2.ReadOnly = true;

textBox3.ReadOnly = true;

textBox4.ReadOnly = true;

textBox5.ReadOnly = true;

textBox6.ReadOnly = true;

listBoxEmployee.Enabled = true;

}

public void Save2()

{

bmEmployee2.EndCurrentEdit();

DataSet.gameplayDataTable ds2 = (DataSet.gameplayDataTable)dsEmployee.gameplay.GetChanges(DataRowState.Added);

if (ds2 != null)

try

{

dagameplay.Update(ds2);

ds2.Dispose();

dsEmployee.games.AcceptChanges();

}

catch (Exception x)

{

string mes2 = x.Message;

MessageBox.Show("Ошибка вставки записи в базу данных 'Podatku' " + mes2, "Предупреждение");

this.dsEmployee.games.RejectChanges();

}

DataSet.gameplayDataTable ds1 = (DataSet.gameplayDataTable)dsEmployee.gameplay.GetChanges(DataRowState.Modified);

if (ds1 != null)

try

{

this.dagameplay.Update(ds1);

ds1.Dispose();

dsEmployee.gameplay.AcceptChanges();

MessageBox.Show("Збережено");

}

catch (Exception x)

{

string mes = x.Message;

MessageBox.Show("Ошибка обновления базы данных 'Podatku' " + mes, "Предупреждение");

this.dsEmployee.gameplay.RejectChanges();

}

//MessageBox.Show("Save");

textBox1.ReadOnly = true;

textBox2.ReadOnly = true;

textBox3.ReadOnly = true;

textBox4.ReadOnly = true;

textBox5.ReadOnly = true;

textBox6.ReadOnly = true;

listBoxEmployee.Enabled = true;

}

public void Remove()

{

int pos = -1;

pos = this.BindingContext[dsEmployee, "games"].Position;

DialogResult result = MessageBox.Show("Видалити данні \n" + "по "+ textBox2.Text + " ?","Попередження!",

MessageBoxButtons.YesNo,

MessageBoxIcon.Warning,

MessageBoxDefaultButton.Button2);

switch (result)

{

case DialogResult.Yes:

{

this.dsEmployee.games.Rows[pos].Delete();

if (this.dsEmployee.games.GetChanges(DataRowState.Deleted) != null)

{

try

{

this.dagames.Update(dsEmployee.games);

this.dsEmployee.games.AcceptChanges();

}

catch (Exception x)

{

string er = x.Message.ToString();

MessageBox.Show("Ошибка удаления записи в базе данных 'Podatku' " + er, "Предупреждение");

this.dsEmployee.games.RejectChanges();

}

}

MessageBox.Show("Удаление данных");

break;

}

case DialogResult.No:

{

this.dsEmployee.games.RejectChanges();

MessageBox.Show("Отмена удаления данных");

break;

}

}

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

New();

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

Edit();

}

private void toolStripButton4\_Click(object sender, EventArgs e)

{

Save();

}

private void toolStripButton5\_Click(object sender, EventArgs e)

{

Remove();

}

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

{

gameplay gp = new gameplay();

gp.ShowDialog();

getHistory();

}

private void splitContainerEmployee\_Panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

getHistory();

}

private void splitContainerEmployee\_Panel2\_Paint(object sender, PaintEventArgs e)

{

}

}

}

Код вікна для введення інформації про взаємодію гравців у кожній грі:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

using System.Data.OleDb;

using System.Data.Common;

using System.Threading;

using System.Data.SqlClient;

namespace Maf\_Project

{

public partial class gameplay : Form

{

public gameplay()

{

InitializeComponent();

}

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

{

SqlConnection c = new SqlConnection(Properties.Settings.Default.MafiaConnectionString);

SqlCommand query = new SqlCommand("INSERT INTO gameplay (id\_game, id\_gamer, role, at\_table, fols, position)" +

"values ('" + textBox1.Text + "', '" + textBox2.Text + "', '" + comboBox1.Text + "', '" + comboBox2.Text + "', '" + numericUpDown1.Text + "', '" + numericUpDown2.Text + "')", c);

c.Open();

try

{

query.BeginExecuteNonQuery();

}

catch

{

}

}

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

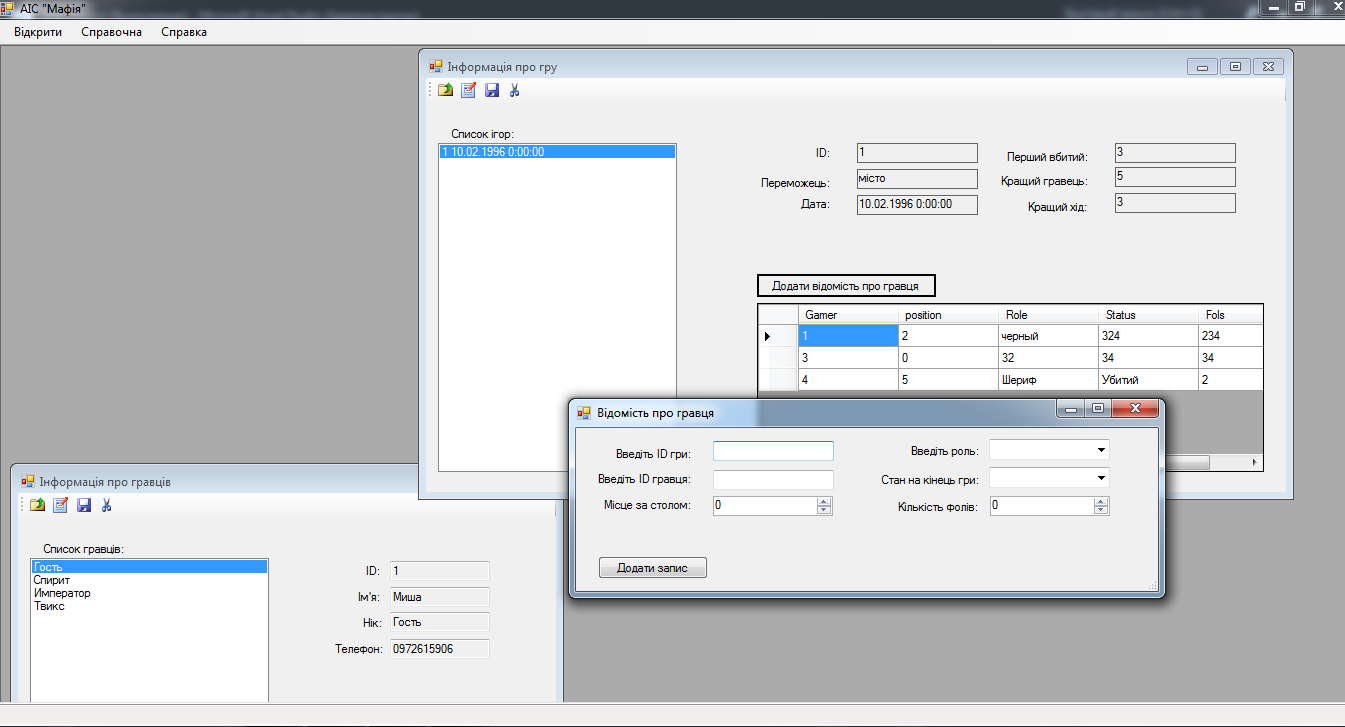
{

}

}

}

Приклад відображення інформації у програмі:



# **Висновки:**

Під час виконання лабораторної роботи, я навчився працювати з системою контролю версій git, використовувати для створння та споглядання результатів сайт github.com та навчився роботі у команді.