Міністерствоосвіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет Прикладної математики

Кафедра спеціалізованих комп’ютерних систем

Лабораторна робота №4

*з дисципліни:*

«Комп’ютерна архітектура»

Розробка розподілених додатків на основі технології Web-сервісів

Виконали:

студенти групи КВ-91

Федай Г.Г

Глушко Д. В.

Перевірив:

Петрашенко А. В.

Київ 2012

**Мета роботи:** вивчення засобів Web-орієнтованого клієнт-серверного програмування засобами .NET Framework та мови C# на основі технології Web-сервісів.

**Завдання:** Розробити систему обміну повідомленнями в реальному часі між двома та більшою кількістю користувачів з використанням Web-сервісів.

**Задачі клієнта:**

Створення інтерфейсу користувача з можливостями реєстрації та аутентифікації користувача, передачі повідомлень на сервер, а також отримання повідомлень, які передали інші користувачі.

**Задачі сервера:**

Організація Web-сервісів для отримання та передачі повідомлень від користувачів, ведення журналу передачі повідомлень ( у файлі або базі даних ), а також успішних та неуспішних спроб аутентифікації користувачів. Передбачити можливість створення та блокування облікових записів користувачів.

Додаткове завдання:

Отримати всі повідомлення за поточну дату.

**Сервер**

using System;

using System.Collections.Generic;

using System.IO;

using System.Text;

using System.Linq;

using System.Web;

using System.Web.Services;

namespace KA4\_server

{

/// <summary>

/// Summary description for WebService1

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class WebService1 : System.Web.Services.WebService

{

string LoadFromFile(string filename, string authName)

{

BinaryReader br = new BinaryReader(new FileStream(filename, FileMode.OpenOrCreate, FileAccess.Read));

try

{

string str = br.ReadString();

string[] sequ = str.Split('\n');

for ( int i = 0; i<sequ.Length; i++ )

{

//string[] dat = str.Split(';');

string[] dat = sequ[i].Split(';');

br.Close();

if (dat[0] == authName)

{

br.Close();

return sequ[i];

}

}

}

catch (Exception ex)

{

br.Close();

return "";

}

br.Close();

return "";

}

[WebMethod]

public string Autorization(string name, string pass)

{

string str\_reg = LoadFromFile(@"D:\registered.txt", name);

if (str\_reg == "")

return "user isn't registered";

string str = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\active.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

str = br.ReadString();

string[] data = str.Split(' ');

if (data.Contains(name))

{

br.Close();

return "user has already entered";

}

}

catch(Exception ex)

{

}

br.Close();

User us = new User();

string[] dat = str\_reg.Split(';');

us.UserName = dat[0];

us.Password = dat[1];

if (dat[1] != pass)

return "wrong password! try again";

if (dat[2] == "blocked")

return "current user is blocked";

str += name + " ";

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\active.txt", FileMode.Truncate));

bw.Write(str);

bw.Close();

return "hello, " + us.UserName + "!";

}

[WebMethod]

public string Register(string smth)

{

string[] dat = smth.Split(';');

string curr\_us = LoadFromFile(@"D:\registered.txt", dat[0]);

if (curr\_us != "")

return "This nickname is busy!";

string str = "";

string buf = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\registered.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

while ((str = br.ReadString()) != null)

buf += str + "\n";

}

catch(Exception ex)

{ }

br.Close();

buf += smth + "user\n";

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\registered.txt", FileMode.Truncate));

bw.Write(buf);

bw.Close();

return "You are registered! Please enter using your login";

}

[WebMethod]

public string GetStatus(string Authname)

{

string name = Authname;

string str = LoadFromFile(@"D:\registered.txt", name);

string[] dat = str.Split(';');

return dat[2];

}

[WebMethod]

public string GetRegUsers()

{

string str = "";

string users = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\registered.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

str = br.ReadString();

string[] sequ = str.Split('\n');

for (int i = 0; i < sequ.Length; i++)

{

if (sequ[i] == "")

continue;

string[] dat = sequ[i].Split(';');

users += dat[0] + ";";

}

}

catch (Exception ex)

{

}

br.Close();

return users;

}

[WebMethod]

public void ChangeStatus(string name, string status)

{

string str;

string buf = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\registered.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

str = br.ReadString();

string[] sequ = str.Split('\n');

for (int i = 0; i < sequ.Length; i++)

{

if (sequ[i] == "")

continue;

string[] dat = sequ[i].Split(';');

if (dat[0] == name)

buf += dat[0] + ";" + dat[1] + ";" + status + "\n";

else buf += sequ[i] + "\n";

}

}

catch (Exception ex)

{

}

br.Close();

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\registered.txt", FileMode.Truncate));

bw.Write(buf);

bw.Close();

}

[WebMethod]

public void AddMsg(string msg)

{

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\session.txt", FileMode.Append));

//string buf = user + " " + DateTime.Now.ToString() + "\n" + msg + "\n" + "-----------------------------------------------\n";

bw.Write(msg);

bw.Close();

}

[WebMethod]

public string Refresh(string fname)

{

string buf = "";

BinaryReader br = new BinaryReader(new FileStream(fname, FileMode.OpenOrCreate, FileAccess.Read));

try

{

string str = "";

while ((str = br.ReadString()) != null)

{

buf += str;

}

}

catch (Exception ex)

{

br.Close();

return buf;

}

br.Close();

return buf;

}

[WebMethod]

public string refreshUsers()

{

string buf = "";

string str = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\active.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

str = br.ReadString();

}

catch (Exception ex)

{

br.Close();

return buf;

}

br.Close();

string[] dat = str.Split(' ','\t','\n');

for ( int i = 0; i<dat.Length; i++)

if ( dat[i] != "" )

{

return str;

}

string msg = Refresh(@"D:\session.txt");

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\session.txt", FileMode.Truncate));

bw.Close();

bw = new BinaryWriter(new FileStream(@"D:\journal.txt", FileMode.Truncate));

bw.Close();

bw = new BinaryWriter(new FileStream(@"D:\active.txt", FileMode.Truncate));

bw.Close();

bw = new BinaryWriter(new FileStream(@"D:\history.txt", FileMode.Append));

bw.Write(msg);

bw.Close();

return "";

}

[WebMethod]

public string GetHistory()

{

string buf = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\history.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

string str = "";

while ((str = br.ReadString()) != null)

{

buf += str;

}

}

catch (Exception ex)

{

//br.Close();

//return buf;

}

br.Close();

string[] sequ = buf.Split('\n');

buf = "";

for (int i = 0; i < sequ.Length; i+=3)

{

string[] dat = sequ[i].Split(' ');

int j = 1;

string temp = "";

for (j = 1; j < dat.Length; j++)

{

if (dat[j] != "")

{

temp += dat[j] + " 0:00:00";

break;

}

}

if ( temp == DateTime.Now.Date.ToString() )

buf += sequ[i] + "\n" + sequ[i + 1] + "\n" + sequ[i + 2] + "\n";

}

return buf + Refresh(@"D:\session.txt");

}

[WebMethod]

public string UserLogOff(string name)

{

string buf = "";

string ret = "";

BinaryReader br = new BinaryReader(new FileStream(@"D:\active.txt", FileMode.OpenOrCreate, FileAccess.Read));

try

{

string str = br.ReadString();

string[] dat = str.Split(' ');

for (int i = 0; i < dat.Length; i++)

{

if (dat[i] != name)

buf += dat[i] + " ";

}

}

catch (Exception ex)

{

//br.Close();

//return ret;

}

br.Close();

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\active.txt", FileMode.Truncate));

bw.Write(buf);

bw.Close();

return buf;

}

[WebMethod]

public void WriteDownToJourn(string sign)

{

BinaryWriter bw = new BinaryWriter(new FileStream(@"D:\journal.txt", FileMode.Append));

bw.Write(sign);

bw.Close();

}

[Serializable]

class User

{

public string UserName;

public string Password;

public bool blocked;

public User()

{

}

public User(string name, string pass, bool isblock)

{

UserName = name;

Password = pass;

blocked = isblock;

}

}

}

}

**Клієнт**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.IO;

using System.Web.Services;

using System.Windows.Forms;

using KA4\_client.localhost;

namespace KA4\_client

{

public partial class Form1 : Form

{

localhost.WebService1 service;

string login, passwd;

Form2 frm;

Form3 usinfo;

string status;

public string uslogin;

public string usstat;

public Form1()

{

InitializeComponent();

frm = new Form2();

usinfo = new Form3(this);

}

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

{

service = new localhost.WebService1();

}

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

{

string str = service.Autorization(LogName.Text, LogPass.Text);

login = LogName.Text;

string[] buf = str.Split(' ',',','!');

if (buf[0] == "hello")

{

button3.Enabled = true;

curr\_msg.Enabled = true;

history.Text += DateTime.Now.ToString() + " : " + LogName.Text + " logged in\n";

service.WriteDownToJourn(DateTime.Now.ToString() + " : " + LogName.Text + " logged in\n");

status = service.GetStatus(login);

if (status == "admin")

button5.Visible = true;

button1.Enabled = false;

button2.Enabled = false;

ext.Enabled = true;

} else

MessageBox.Show(str);

}

private void button2\_Click(object sender, EventArgs e)

{

passwd = LogPass.Text;

string str = service.Register(LogName.Text + ";" + LogPass.Text + ";");

//login = LogName.Text;

MessageBox.Show(str);

//button1.Enabled = false;

//button2.Enabled = false;

//ext.Enabled = true;

}

private void button3\_Click(object sender, EventArgs e)

{

string msg = curr\_msg.Text;

string buf = login + " " + DateTime.Now.ToString() + "\n" + msg + "\n" + "-----------------------------------------------" + "\n";

all\_messages.Text += buf;

service.AddMsg(buf);

curr\_msg.Text = "";

}

private void timer1\_Tick(object sender, EventArgs e)

{

string str = service.Refresh(@"D:\session.txt");

all\_messages.Text = str;

str = "";

str = service.refreshUsers();

online.Text = str;

str = service.Refresh(@"D:\journal.txt");

history.Text = str;

}

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

{

button1.Enabled = true;

button2.Enabled = true;

button3.Enabled = false;

ext.Enabled = false;

listBox1.Visible = false;

button5.Visible = false;

LogName.Text = "";

LogPass.Text = "";

if (login == "")

return;

string s = service.UserLogOff(login);

history.Text += DateTime.Now.ToString() + " : " + login + " logged off\n";

service.WriteDownToJourn(DateTime.Now.ToString() + " : " + login + " logged off\n");

online.Text = s;

login = "";

passwd = "";

status = "";

listBox1.Visible = false;

button5.Visible = false;

}

private void button4\_Click(object sender, EventArgs e)

{

if (frm.Created)

frm.Focus();

else

{

frm = new Form2();

frm.Show();

}

frm.richTextBox1.Text = service.GetHistory();

}

private void Form\_Close(object sender, FormClosingEventArgs e)

{

ext\_Click(sender, e);

}

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

{

listBox1.Visible = !listBox1.Visible;

if (listBox1.Visible == true)

{

string str = service.GetRegUsers();

string[] users = str.Split(';');

listBox1.Items.Clear();

for (int i = 0; i < users.Length; i++)

listBox1.Items.Add(users[i]);

}

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

uslogin = listBox1.SelectedItem.ToString();

usstat = service.GetStatus(uslogin);

if (usinfo.Created)

usinfo.Focus();

else

{

usinfo = new Form3(this);

usinfo.name = uslogin;

usinfo.stat = usstat;

usinfo.Show();

}

}

private void timer2\_Tick(object sender, EventArgs e)

{

timer2.Stop();

service.ChangeStatus(uslogin, usstat);

}

}

}

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;

namespace KA4\_client

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

}

}

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.Web.Services;

using KA4\_client.localhost;

namespace KA4\_client

{

public partial class Form3 : Form

{

Form1 f = null;

public Form3( Form1 basis )

{

InitializeComponent();

f = basis;

}

public string name;

public string pass;

public string stat;

private void Form3\_Load(object sender, EventArgs e)

{

nick.Text = name;

if (stat == "user" || stat == "admin")

{

radioButton2.Checked = true;

}

else radioButton1.Checked = true;

}

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

{

f.timer2.Start();

this.Close();

}

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

{

f.usstat = radioButton2.Text;

}

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

{

f.usstat = radioButton1.Text;

}

}

}