

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем автоматизованого проектування

Звіт

З дисципліни

«Екстремальне програмування»

Лабораторної робота №4

«Основи WPF. Розробка клієнта»

Варіант №14

Виконав:

Студент гр.

КН-311

Куриляк Ю.А.

Прийняв:

Щербак С.С.

Львів – 2020

Виконання роботи:

MainWindow

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Windows.Threading;
using ChatClient.ServiceChat;
using System.Windows.Media.Animation;
using ChatClient.Pages;
using ChatClient.ServiceChat;

namespace ChatClient
{
    /// <summary>
    /// Л о г и к а  в з а и м о д е й с т в и я  д л я  MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window, IServiceChatCallback
    {
        ServiceChatClient client;

        int userId = -1;

        private Page pageAuthorization;
```

```

private Page pageRegistration;
private pChat pageChat;
private Page pageProjects;

public double mainFrameOpacity;

bool StateClosed = true;

public MainWindow()
{
    InitializeComponent();

    client = new ServiceChatClient(new
System.ServiceModel.InstanceContext(this));
    pageAuthorization = new pAuthorization(this, mainFrame, client);

    mainFrameOpacity = 1;
    mainFrame.Content = pageAuthorization;
}

public void setUserId(int id)
{
    userId = id;
    changeButtonAccess();
}

private void disconnect()
{
    client.Disconnect(userId);
}

private void changeButtonAccess()
{
    b_hMenuChats.IsEnabled = !b_hMenuChats.IsEnabled;
}

```

```

        b_hMenuProjects.IsEnabled = !b_hMenuProjects.IsEnabled;
    }

    private void ButtonMenu_Click(object sender, RoutedEventArgs e)
    {
        if (StateClosed)
        {
            Storyboard sb = this.FindResource("OpenMenu") as Storyboard;
            sb.Begin();
        }
        else
        {
            Storyboard sb = this.FindResource("CloseMenu") as Storyboard;
            sb.Begin();
        }

        StateClosed = !StateClosed;
    }

    private void b_hMenuProjects_click(object sender, RoutedEventArgs e)
    {
        pageProjects = new pProjects(client, userId);
        mainFrame.Content = pageProjects;
    }

    private void b_hMenuChats_click(object sender, RoutedEventArgs e)
    {
        pageChat = new pChat(client, userId);
        mainFrame.Content = pageChat;
    }

    public void MsgCallback(string msg)
    {
        pageChat.MsgCallback(msg);
    }

```

```

    }
}
}

```

pAuthorization

```

using ChatClient.Pages;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using ChatClient.ServiceChat;

```

```

namespace ChatClient

```

```

{
    /// <summary>
    /// Interaction logic for pAuthorization.xaml
    /// </summary>
    ///

    public partial class pAuthorization : Page, IServiceChatCallback
    {
        private Page pageRegistration;
        ServiceChatClient client;
        MainWindow mainWindow;

        int avtorization;
        Frame mainWindowFrame;

        public pAuthorization(MainWindow _mainWindow, Frame _mainWindowFrame,
ServiceChatClient _client)
        {
            mainWindowFrame = _mainWindowFrame;
            mainWindow = _mainWindow;
            client = _client;
            InitializeComponent();
        }

        public pAuthorization(MainWindow _mainWindow, ServiceChatClient _client)
        {
            mainWindow = _mainWindow;
            client = _client;
            InitializeComponent();
        }
    }
}

```

```

        public void MsgCallback(string msg)
        {
            throw new NotImplementedException();
        }

        public void btnSubmit_Click(object sender, RoutedEventArgs e)
        {
            string login = txtLogin.Text;
            string password = txtPassword.Password;
            checkAuthorization(login, password);
        }

        public void checkAuthorization(string login, string password)
        {
            avtorization = client.UserAuthorization(login, password);

            if (avtorization != -1)
            {
                MessageBox.Show("You authorised");
                mainWindow.setUserId(avtorization);
            }
            else
                MessageBox.Show("Wrong login or password");
        }

        private void btnToRegistration(object sender, RoutedEventArgs e)
        {
            pageRegistration = new pRegistration(mainWindow, mainWindowFrame,
client);
            mainWindowFrame.Content = pageRegistration;
        }
    }
}

```

pRegistration

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using ChatClient.ServiceChat;

namespace ChatClient.Pages
{
    public partial class pRegistration : Page, IServiceChatCallback
    {
        private Page pageAuthorization;
    }
}

```

```

        Frame mainWindowFrame;
        ServiceChatClient client;
        MainWindow mainWindow;

        public pRegistration(MainWindow _mainWindow, Frame _mainWindowFrame,
ServiceChatClient _client)
        {
            InitializeComponent();
            mainWindowFrame = _mainWindowFrame;
            mainWindow = _mainWindow;
            client = _client;
            client = new ServiceChatClient(new
System.ServiceModel.InstanceContext(this));
        }

        public void MsgCallback(string msg)
        {
            throw new NotImplementedException();
        }

        private void btnSubmit_Click(object sender, RoutedEventArgs e)
        {
            string username = txtRegUsername.Text;
            string email = txtRegEmail.Text;
            string password = txtRegPassword.Password;
            //MessageBox.Show((username + " " + email + " " + password));
            bool registration = client.UserRegistration(username, email,
password);
            if (registration)
            {
                btnToAutorization(sender, e);
                MessageBox.Show("You successfully registered");
            }
            else
            {
                MessageBox.Show("This username or e-mail already exists");
            }
        }

        private void btnToAutorization(object sender, RoutedEventArgs e)
        {
            pageAuthorization = new pAuthorization(mainWindow, mainWindowFrame,
client);
            mainWindowFrame.Content = pageAuthorization;
        }
    }
}

```

pProjects

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;

```

```

using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using ChatClient.ServiceChat;
using MySql.Data.MySqlClient.Memcached;

namespace ChatClient.Pages
{
    /// <summary>
    /// Interaction logic for pProjects.xaml
    /// </summary>
    public partial class pProjects : Page
    {
        //private BindingList<Task> tasks;
        //private BindingList<Project> projects;
        ServiceChatClient client;
        int userId;
        string activeProject;
        ChatClient.ServiceChat.Project[] projects;
        ChatClient.ServiceChat.Task[] tasks;
        string activeProj;

        public pProjects(ServiceChatClient _client, int _userId)
        {
            InitializeComponent();
            client = _client;
            userId = _userId;
            SetProjectList();
        }

        public void SetProjectList()
        {
            projects = client._GetAllProjects();
            //lbMain.ItemsSource = pprojects;
            lbMain.ItemsSource = projects;
        }

        private void bFindProject(object sender, RoutedEventArgs e)
        {
        }

        private void SetProjectnameField_KeyDown(object sender, KeyEventArgs e)
        {
        }

        void HideProjects()
        {
            projectsGrid.Visibility = Visibility.Hidden;
        }
    }
}

```



```

void ShowProjects()
{
    projectsGrid.Visibility = Visibility.Visible;
}

void OpenTasks(string id)
{
    tasksGrid.Visibility = Visibility.Visible;
    tasks = client.GetTasksInProject(id);
    if (tasks != null)
    {
        lbTasks.ItemsSource = tasks;
    }
}

void HideTasks()
{
    tasksGrid.Visibility = Visibility.Hidden;
}

void OpenNewProjectForm()
{
    newProjectsGrid.Visibility = Visibility.Visible;
}

void HideNewProjectFrom()
{
    newProjectsGrid.Visibility = Visibility.Hidden;
}

private void ibMainProjectsButton_click(object sender, RoutedEventArgs e)
{
    HideProjects();
    activeProj = (sender as Button).Tag.ToString();
    OpenTasks(activeProj);
}

private void ibMainTasksButton_click(object sender, RoutedEventArgs e)
{
}

private void bCreateProject(object sender, RoutedEventArgs e)
{
    HideProjects();
    OpenNewProjectForm();
}

private void bAddProject(object sender, RoutedEventArgs e)
{
    Project project = new Project()
    {
        name = txtProjectName.Text,
        manager = txtProjectManager.Text,
        methodology = txtProjectMethodology.Text,
        deadline = txtProjectDeadline.Text
    };
}

```

```

        if (client.AddProject("'" + project.name + "'", '"' + project.manager +
"'', '"' + project.methodology + '"'))
        {
            MessageBox.Show("Project Added");
            HideNewProjectForm();
            ShowProjects();
        }
    }

    private void addTasks(object sender, RoutedEventArgs e)
    {
        OpenNewTaskFrom();
        HideTasks();
    }

    private void OpenNewTaskFrom()
    {
        newTaskGrid.Visibility = Visibility.Visible;
    }

    private void HideNewtaskForm()
    {
        newTaskGrid.Visibility = Visibility.Hidden;
    }

    private void bCreateTask(object sender, RoutedEventArgs e)
    {
        Task task = new Task()
        {
            name = txtTaskName.Text,
            project = activeProj,
            developer = txtTaskDeveloper.Text,
            complexity = Int32.Parse(txtTaskComplexity.Text),
            deadline = txtTaskDeadline.Text
        };

        if(client.AddTask("'" + task.project + "'", '"' + task.name + "'", '"' +
task.developer + "'", '"' + task.complexity + "'", '"' + task.deadline + '"'))
        {
            MessageBox.Show("Task added");
            HideNewtaskForm();
            ShowProjects();
        }
    }
}

```

pChat

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;

```

```

using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using ChatClient.ServiceChat;

namespace ChatClient.Pages
{
    /// <summary>
    /// Interaction logic for pChat.xaml
    /// </summary>
    public partial class pChat : Page, IServiceChatCallback
    {
        bool isConnected = false;
        ServiceChatClient client;
        int userId;
        string currentUser;

        public pChat(ServiceChatClient _client, int _userId)
        {
            InitializeComponent();
            client = _client;
            userId = _userId;
            //GetUserContacts();
        }

        /*private void GetUserContacts()
        {
            client.UserContactListUsernames(userId);
        }*/

        private void SetUsernameField_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.Key == Key.Enter)
            {
                SelectUser(txtfindUsername.Text);
            }
        }

        private void SelectUser(string username)
        {
            currentUser = txtfindUsername.Text;
            lbChat.Items.Clear();
            ShowDialog();
        }

        private void ShowDialog()
        {
            // з а п и т у б д з а п о в і д о м л е н н я м и
            lbChat.Visibility = Visibility.Visible;
            tbMessage.Visibility = Visibility.Visible;
            messagePageLabel.Visibility = Visibility.Hidden;
        }
    }
}

```

```

    }

    /*
        вивести діалог з юзером при виборі його у
        меню або через find
    */

    /*
        * Вивести збоку усіх юзерів з якими
        контактував користувач
    */
    /*private void changeContactList()
    {
        < ListViewItem Width = "200" Height = "50" >

            < StackPanel Orientation = "Horizontal" >

                < Button Name = "b_hMenuChats" Content = "Chats"
                IsEnabled = "False" Command = "{Binding b_hMenuChat_click}" HorizontalAlignment =
                "Right" Click = "b_hMenuChats_click" ></ Button >

                <!--< TextBlock Text = "TextFild 2"
                HorizontalAlignment = "Right" ></ TextBlock > -->

            </ StackPanel >

        </ ListViewItem >

    }*/

    // вивести у діалог з юзером
    public void MsgCallback(string msg)
    {
        //lbChat.Items.Add("< Button x: Name = \"bConnDicon\" Click =
        \"Button_Click\" Content = \"\" + (msg) + \"\" Margin = \"10,10,0,0\" />");

        Button b = new Button();
        string sender = msg.Substring(0, msg.IndexOf(';'));
        msg = msg.Substring(msg.IndexOf(';') + 1, (msg.Length -
        sender.Length-1));

        b.Name = "a";
        //b.Click =
        b.Content = msg;
        if (currentUser != sender) {
            b.Visibility = Visibility.Collapsed;
            //позначити у меню користувачів нове
            повідомлення
        }
        lbChat.Items.Add(b);
        lbChat.ScrollIntoView(lbChat.Items[lbChat.Items.Count - 1]);
    }

    //переробити на find user
    private void tbMessage_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.Key == Key.Enter)
        {
            if (client != null)

```

```

        {
            client.SendMsg(tbMessage.Text, userId, currentUser); //
Відправити ід отримувача
            Button b = new Button();
            b.Name = "a";
            b.Content = DateTime.Now.ToShortTimeString() + ", Me : " +
tbMessage.Text;
            lbChat.Items.Add(b);
            lbChat.ScrollIntoView(lbChat.Items[lbChat.Items.Count - 1]);

            tbMessage.Text = string.Empty;
        }
    }

private void bFindUser(object sender, RoutedEventArgs e)
{
    if (txtfindUsername.Text != "Find User")
    {
        SelectUser(txtfindUsername.Text);
    }
}
}

```