ст. Гр КН-923с

Кузьмін Станіслав

**Звіт з лабораторної роботи №12**

**Варіант 6**

Створення багатовіконних застосунків. Частина 2

**Мета роботи:** набути навички у створенні багатовіконних застосунків з більш розвиненим функціоналом.

Завдання:

1. Створити проект застосунку умовної соціальної мережі з можливістю авторизації, реєстрації, модифікації власного профілю та обміну повідомленнями між користувачами.

2. Додати до проекту форму реєстрації користувачів умовної системи із використанням реєстраційної форми згідно до власного варіанту. Перед занесенням реєстраційних даних до відповідної таблиці, виконати перевірку (валідацію) реєстраційних полів:

UserName: лише числа та символи латиниці кількістю не більше 15 і щоб ім'я було унікальним (раніше не було в таблиці БД);

Password: числа, символи латиниці та обов'язково спеціальні символи довжиною не менше 12, та його коректне повторення у відповідному полі;

Email: коректне значення електронної поштової адреси

using System;

using System.IO;

using System.Linq;

using System.Text.RegularExpressions;

using System.Windows.Forms;

namespace Task\_1

{

public partial class Register : Form

{

string name, username, password1, password2;

int userId;

public Register()

{

InitializeComponent();

InitializeUserId(); // Ініціалізація ID користувача при запуску форми

}

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

{

// Ініціалізація пароля, імені та логіну

if (initializePassword() && initializeUsername() && initializeName())

{

// Збереження даних

SaveData();

// Дії після успішної реєстрації

MessageBox.Show("Реєстрація успішна!", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

this.Close();

}

// Метод для ініціалізації User ID (зчитує останній ID з файлу і збільшує його)

public void InitializeUserId()

{

string filePath = "user\_data.txt";

if (File.Exists(filePath))

{

// Якщо файл існує, зчитуємо останній ID

string[] lines = File.ReadAllLines(filePath);

string lastUserIdLine = lines.LastOrDefault(line => line.StartsWith("UserID:"));

if (lastUserIdLine != null)

{

string lastUserId = lastUserIdLine.Split(':')[1].Trim();

userId = int.Parse(lastUserId) + 1; // Збільшуємо на 1

}

else

{

userId = 1; // Якщо немає користувачів, починаємо з 1

}

}

else

{

userId = 1; // Якщо файлу немає, починаємо з 1

}

}

// Метод для перевірки і ініціалізації імені користувача

public bool initializeUsername()

{

// Перевірка на порожнє значення

if (string.IsNullOrEmpty(textBox2.Text))

{

MessageBox.Show("Username не може бути порожнім.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка на заборонені символи

if (textBox2.Text.Any(c => "!@#$%^&\*()\_+={}[]:;\"'<>,.?/-".Contains(c)))

{

MessageBox.Show("Username не повинно містити символи.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка на наявність кириличних символів

if (textBox2.Text.Any(c => c >= 'а' && c <= 'я' || c >= 'А' && c <= 'Я'))

{

MessageBox.Show("Username не повинно містити кириличні символи.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

username = textBox2.Text;

return true;

}

// Метод для перевірки і ініціалізації імені

public bool initializeName()

{

if (string.IsNullOrEmpty(textBox1.Text))

{

MessageBox.Show("Name не може бути порожнім.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

name = textBox1.Text;

return true;

}

// Метод для перевірки і ініціалізації пароля

public bool initializePassword()

{

// Перевірка збігу паролів

password1 = textBox3.Text;

password2 = textBox4.Text;

if (password1 != password2)

{

MessageBox.Show("Паролі не збігаються.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка довжини пароля

if (password1.Length < 12)

{

MessageBox.Show("Пароль повинен містити мінімум 12 символів.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка на наявність хоча б однієї букви

if (!password1.Any(c => char.IsLetter(c)))

{

MessageBox.Show("Пароль повинен містити хоча б одну букву.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка на наявність хоча б однієї цифри

if (!password1.Any(c => char.IsDigit(c)))

{

MessageBox.Show("Пароль повинен містити хоча б одну цифру.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

// Перевірка на наявність хоча б одного спеціального символа

if (!password1.Any(c => "!@#$%^&\*()\_+={}[]:;\"'<>,.?/-".Contains(c)))

{

MessageBox.Show("Пароль повинен містити хоча б один спеціальний символ.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

return true; // Пароль пройшов всі перевірки

}

// Метод для збереження даних у файл

public void SaveData()

{

string filePath = "user\_data.txt";

// Записуємо дані у файл

try

{

using (StreamWriter writer = new StreamWriter(filePath, true))

{

// Зберігаємо ID, ім'я, логін та пароль

writer.WriteLine($"UserID: {userId:D4}"); // Форматуємо ID з ведучими нулями (наприклад, 0001, 0002)

writer.WriteLine("Name: " + name);

writer.WriteLine("UserName: " + username);

writer.WriteLine("Password: " + password1);

writer.WriteLine("--------------------------------------------------");

}

}

catch (Exception ex)

{

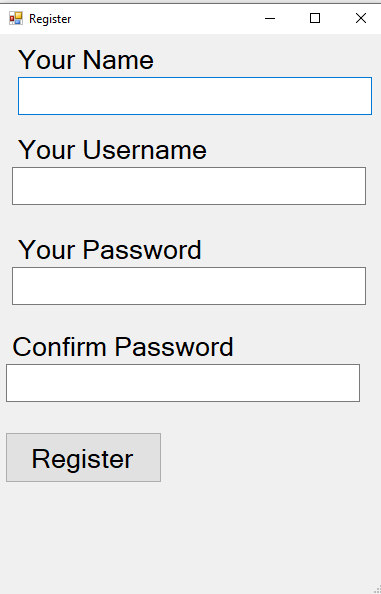
MessageBox.Show("Помилка при збереженні даних: " + ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}



3. Додати до проекту форму авторизації користувачів умовної системи із використанням дизайну реєстраційної форми з минулого завдання відповідного варіанту. Авторизаційна форма має складатися з двох полів: Username та Password, до яких користувач має ввести відповідну інформацію. У випадку коректного вводу система має підтвердити авторизацію користувача, у протилежному випадку повернутися до назад форми та видати повідомлення про помилку в позиції згідно до свого варіанта

using System;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace Task\_1

{

public partial class Authorization : Form

{

private string userName, password;

private int userID;

public Authorization()

{

InitializeComponent();

}

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

{

Register register = new Register();

register.ShowDialog();

}

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

{

userName = textBox2.Text;

password = textBox3.Text;

string filePath = "user\_data.txt";

// Перевірка, чи існує файл

if (File.Exists(filePath))

{

string[] lines = File.ReadAllLines(filePath);

bool isAuthenticated = false;

// Змінні для зберігання даних

string storedUserName = null, storedPassword = null;

int storedUserID = 0;

foreach (var line in lines)

{

if (line.StartsWith("UserID:"))

{

// Отримання UserID

storedUserID = Convert.ToInt32(line.Split(':')[1].Trim());

}

else if (line.StartsWith("UserName:"))

{

// Отримання UserName

storedUserName = line.Split(':')[1].Trim();

}

else if (line.StartsWith("Password:"))

{

// Отримання Password

storedPassword = line.Split(':')[1].Trim();

}

// Перевірка облікових даних з урахуванням регістру

if (storedUserName != null && storedPassword != null)

{

if (string.Equals(storedUserName, userName, StringComparison.OrdinalIgnoreCase) && storedPassword == password)

{

isAuthenticated = true;

userID = storedUserID; // Встановлюємо UserID поточного користувача

break;

}

}

}

if (isAuthenticated)

{

MessageBox.Show($"Авторизація успішна! UserID: {userID:D4}", "Успіх", MessageBoxButtons.OK, MessageBoxIcon.Information);

UserForm userForm = new UserForm(userID);

this.Hide();

userForm.ShowDialog();

this.Close();

}

else

{

MessageBox.Show("Невірне ім’я користувача або пароль.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

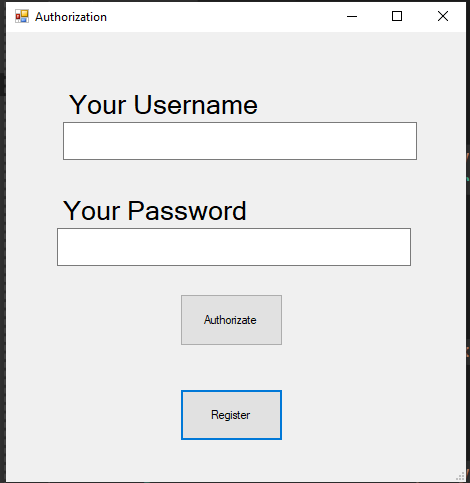
MessageBox.Show("Файл даних користувачів не знайдений.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}



1. Додати до проекту профільну форму користувача умовної системи із використанням дизайну реєстраційної та авторизаційної форм з минулих завдань із розташованими на ній реєстраційними (профільними) параметрами користувача, яка повинна мати можливість змінювати ці параметри. Особливості зміни параметрів визначаються згідно до власного варіанту

using System;

using System.Windows.Forms;

using System.IO;

namespace Task\_1

{

public partial class UserForm : Form

{

private int userID; // Змінна для зберігання ID користувача

private string userName; // Змінна для зберігання імені користувача

// Конструктор форми, приймає ID користувача

public UserForm(int userID)

{

InitializeComponent();

this.userID = userID; // Ініціалізація userID

InitializeUser(); // Завантаження даних користувача

}

// Метод для ініціалізації даних користувача

public void InitializeUser()

{

label1.Text = $"UserID: {userID:D4}"; // Відображення поточного UserID на екрані

string filePath = "user\_data.txt"; // Шлях до файлу з даними користувачів

// Перевірка, чи існує файл з даними

if (File.Exists(filePath))

{

string[] lines = File.ReadAllLines(filePath); // Читання всіх рядків з файлу

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

{

// Пошук рядка, що містить UserID

if (lines[i].StartsWith("UserID:"))

{

int storedUserID = Convert.ToInt32(lines[i].Split(':')[1].Trim()); // Отримання ID з файлу

// Якщо ID користувача збігається з поточним

if (storedUserID == userID)

{

// Ініціалізуємо змінні для зберігання даних

string name = string.Empty;

userName = string.Empty;

string password = string.Empty;

// Перевірка наступних рядків на наявність даних користувача

if (i + 1 < lines.Length && lines[i + 1].StartsWith("Name:"))

name = lines[i + 1].Split(':')[1].Trim(); // Отримання імені

if (i + 2 < lines.Length && lines[i + 2].StartsWith("UserName:"))

userName = lines[i + 2].Split(':')[1].Trim(); // Отримання логіну

if (i + 3 < lines.Length && lines[i + 3].StartsWith("Password:"))

password = lines[i + 3].Split(':')[1].Trim(); // Отримання пароля

// Оновлюємо елементи інтерфейсу для відображення даних користувача

label2.Text = $"Name: {name}";

label3.Text = $"UserName: {userName}";

label4.Text = $"Password: {password}";

break; // Зупиняємо пошук, оскільки користувача знайдено

}

}

}

}

else

{

// Виведення повідомлення, якщо файл не знайдений

MessageBox.Show("Файл даних користувачів не знайдений.", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Обробник події для кнопки зміни імені

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

{

NameChange nameChange = new NameChange(userID, this); // Створюємо форму для зміни імені

nameChange.ShowDialog(); // Відображаємо форму

}

// Обробник події для кнопки зміни логіну

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

{

UserNameChange userNameChange = new UserNameChange(userID, this);

userNameChange.ShowDialog();

}

// Обробник події для кнопки зміни пароля

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

{

ChangePassword changePassword = new ChangePassword(userID, this);

changePassword.ShowDialog();

}

// Обробник події для кнопки відправлення повідомлення

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

{

SendMessageForm sendMessageForm = new SendMessageForm(userName, this);

sendMessageForm.ShowDialog();

}

// Обробник події для кнопки завантаження повідомлення

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

{

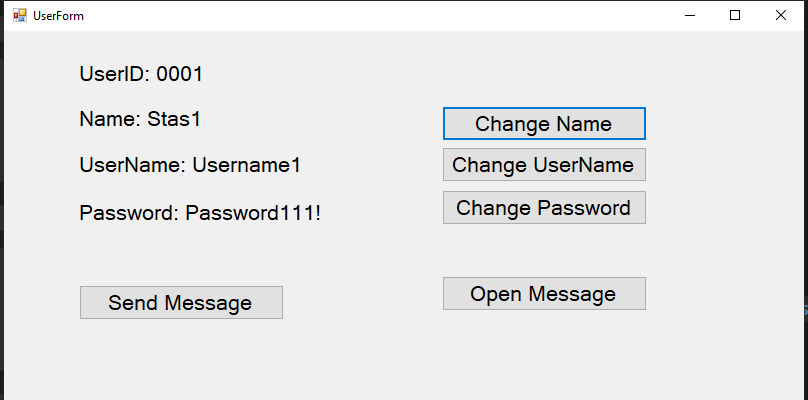
LoadMessage loadMessage = new LoadMessage(userName);

loadMessage.ShowDialog();

}

}

}



5. Додати до проекту форму відправки повідомлень користувачів умовної системи із використанням дизайну форм минулих завдань, яка повинна мати поля ТО для визначення користувача-отримувача

повідомлення, ТНЕМЕ та ТЕХТ для визначення теми і тексту повідомлення відповідно. Повідомлення мають зберігатись у відповідному файлі повідомлень для форми отримання цих повідомлень.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

namespace Task\_1

{

public partial class SendMessageForm : Form

{

private UserForm UserForm;

private string userName;

public SendMessageForm(string userName, UserForm userForm)

{

InitializeComponent();

this.userName = userName;

}

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

{

string recipient = textBox1.Text;

string theme = textBox2.Text;

string messageText = textBox3.Text;

if (string.IsNullOrEmpty(recipient) || string.IsNullOrEmpty(theme) || string.IsNullOrEmpty(messageText))

{

MessageBox.Show("Please fill in all fields.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string filePath = "messages.txt";

try

{

using (StreamWriter writer = new StreamWriter(filePath, true))

{

// ID, имя, логин и пароль

writer.WriteLine($"FROM: {userName}");

writer.WriteLine("TO: " + recipient);

writer.WriteLine("THEME: " + theme);

writer.WriteLine("TEXT: " + messageText);

writer.WriteLine("--------------------------------------------------");

}

MessageBox.Show("Message sent successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

catch (Exception ex)

{

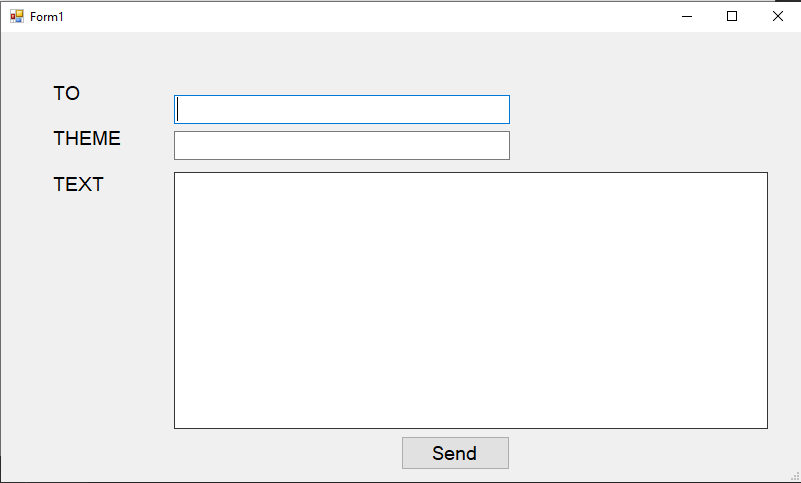
MessageBox.Show("Ошибка при сохранении данных: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}



1. Додати до проекту форму отримання повідомлень користувачів умовної системи із використанням дизайну форм минулих завдань, яка повинна мати перелік повідомлень для поточного користувача із можливістю перегляду кожного повідомлення в окремій формі.

using System;

using System.IO;

using System.Text;

using System.Windows.Forms;

using static System.Windows.Forms.LinkLabel;

using System.Xml.Linq;

namespace Task\_1

{

public partial class LoadMessage : Form

{

private string userName;

public LoadMessage(string userName)

{

InitializeComponent();

this.userName = userName;

LoadMessages(userName);

}

// Завантаження повідомлень з файлу та відображення їх у ListBox

private void LoadMessages(string userName)

{

string filePath = "messages.txt";

if (File.Exists(filePath))

{

try

{

listBoxMessages.Items.Clear(); // Очищаємо список

string[] allMessages = File.ReadAllLines(filePath);

string currentTheme = string.Empty;

// Перебираємо всі рядки файлу

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

{

// Перевіряємо рядок на наявність TO

if (allMessages[i].StartsWith("TO:"))

{

string storedUserName = allMessages[i].Split(':')[1].Trim();

if (storedUserName == userName)

{

// Якщо наступний рядок починається з THEME це тема повідомлення

if (i + 1 < allMessages.Length && allMessages[i + 1].StartsWith("THEME:"))

{

currentTheme = allMessages[i + 1].Split(':')[1].Trim();

listBoxMessages.Items.Add(currentTheme);

}

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Помилка при завантаженні повідомлень: " + ex.Message);

}

}

else

{

MessageBox.Show("Файл повідомлень не знайдений.");

}

}

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

{

if (listBoxMessages.SelectedItem != null)

{

string selectedTheme = listBoxMessages.SelectedItem.ToString();

ShowMessageDetails(selectedTheme);

}

}

// Метод для відображення деталей повідомлення

private void ShowMessageDetails(string theme)

{

string filePath = "messages.txt";

if (File.Exists(filePath))

{

textBox1.Text = null;

textBox2.Text = null;

textBox3.Text = null;

string[] allMessages = File.ReadAllLines(filePath);

bool isMessageFound = false; // Прапор для відстеження

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

{

// Шукаємо рядок з темою повідомлення

if (allMessages[i].StartsWith("THEME:"))

{

string storedTheme = allMessages[i].Split(':')[1].Trim();

if (storedTheme == theme)

{

// Отримуємо FROM

if (i - 2 >= 0 && allMessages[i - 2].StartsWith("FROM:"))

{

string currentFROM = allMessages[i - 2].Split(':')[1].Trim();

textBox1.Text = $"From: {currentFROM}";

}

// Отримуємо THEME

if (i < allMessages.Length && allMessages[i].StartsWith("THEME:"))

{

string currentTheme = allMessages[i].Split(':')[1].Trim();

textBox2.Text = $"Theme: {currentTheme}";

}

// Отримуємо "TEXT"

if (i + 1 < allMessages.Length && allMessages[i + 1].StartsWith("TEXT:"))

{

string currentTEXT = allMessages[i + 1].Split(':')[1].Trim();

textBox3.Text = $"{currentTEXT}";

}

isMessageFound = true;

break;

}

}

}

if (!isMessageFound)

{

MessageBox.Show("Повідомлення з такою темою не знайдено.");

}

}

else

{

MessageBox.Show("Файл повідомлень не знайдений.");

}

}

}

}

1. 