**Форма № Н-6.01у**

Кременчуцький національний університет імені Михайла Остроградського

(повне найменування вищого навчального закладу)

Кафедра автоматизації та інформаційних систем

(повна назва кафедри, циклової комісії)

**КУРСОВИЙ ПРОЄКТ**

**(РОБОТА)**

з дисципліни «Сучасні мови об’єктно-орієнтованого програмування»

(назва дисципліни)

на тему Розробка WPF додатку «Бюро знайомств»

Студента \_2\_ курсу КН–23–1 групи

Ступінь вищої освіти «Бакалавр»

(бакалавр, магістр)

Спеціальність 122 – «Комп’ютерні науки»

Освітньо-професійна програма

«Комп’ютерні науки»

Прізвище І. Б

(прізвище та ініціали)

Керівник старший викладач кафедри АІС

\_\_\_\_\_\_\_ Бельська В. Ю.\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_. Оцінка: ЄКTС \_\_\_\_

Члени комісії   В. Ю. Бельська

(підпис) (ініціали та прізвище)

Н.М.Істоміна

(підпис) (ініціали та прізвище)

О.В.Кліменко

(підпис) (ініціали та прізвище)

м. Кременчук 2024 рік

**Форма № Н-9.01у**

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

Кафедра автоматизації та інформаційних систем

Дисципліна «Сучасні мови об’єктно-орієнтованого програмувння»

Освітній ступінь \_\_ «Бакалавр»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність 122 – «Комп’ютерні науки»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Освітня програма «Комп’ютерні науки»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс 2 група \_\_КН-23-1 семестр \_\_3\_\_

**ЗАВДАННЯ**

**НА КУРСОВИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ляпко Анаствсія Ігорівна \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

(прізвище, ім’я, по-батькові)

1. Тема роботи: Розробка WPF додатку «Бюро знайомств»
2. Термін здачі студентом роботи 09.12. 2024 р
3. Вихідні дані до роботи: Інтегроване середовище розробки Microsoft Visual Studio
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

постановка завдання, аналіз технічного завдання, розробка інтерфейсу програми, структура програмного забезпечення.

1. Перелік графічного матеріалу:
2. Дата видачі завдання: 9.09. 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  пор. | Назва етапів курсового проекту | Терміни виконання етапів проекту | Вказівки та  зауваження викладача (з зазначенням дати консультації) | Оцінювання етапів проекту | | |
| за націо- нальною шкалою | за шкалою ЄКТС | кількість балів |
| 1 | Етап 1  Аналіз предметної області | 01.10.23– 25.10.23 |  |  | | |
| 2 | Етап 2  Створення моделі даних | 16.10.23– 28.10.23 |  |  | | |
| 3 | Етап 3  Розробка віконного інтерфейсу та створення основного програмного коду додатку | 29.10.23– 24.11.23 |  |  | | |
| 4 | Етап 4  Тестування програмного коду | 16.10.23 29.11.23 |  |  | | |
| 5 | Етап 5  Оформлення пояснювальної записки | 25.11.23– 31.11 .23 |  |  | | |
| 6 | Етап 9  Захист | 01.12.23 |  |  | | |
|  | Разом | 8 тижнів |  |  | | |

Студент

(підпис)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_ В. Ю. Бельська \_\_\_\_

(підпис) (ініціали та прізвище)

«9» 09 2024 р.

РЕФЕРАТ

Курсова робота містить 42 сторінки , 2 розділи, 23 рисунки, 1 таблиця, 4 використаних джерел. Об’єкт розробки – WPF додатку «Бюро знаймств» Мета: створення багатовіконного Wpf додатку, що реалізує додаток «Бюро знайосмтв ». Над проектом працювала команда з двох студентів, тому обв’язки було розподілені на такі частини View Ляпко А.І та ViewModel, Model Гур’єв Д.П. Під час виконання завдання, поставленого на курсову роботу, було виконано аналіз основних завдань, які виникають у користувача, якщо він хоче знайти нові знайомства. Як результат було визначено основні функціональні та не функціональні вимоги до додатку, що створюється. В рамках створення View частини додатку було створено 7 вікон, серед яких 4 основних 3 додаткових для виводу супровідної інформації. При створені вікон використовувалися стилі та шаблони, для взаємодії з даними використовувалась двостороння прив’язка даних. Деякі обробники подій реалізовано у вигляді команд. Для створення графічного інтерфейсу використовувся фреймворк WPF та мова C# для обробки подій вікон. Розробка виконувалася у середовищі програмування Visual Studio 2022. Результатом виконання всіх етапів є застосунок «Бюро знайомств», який може реєструвати користувачів, надає ссилку на чат для спілкування , можливість змінювати данні профілю та також можливість повністю видалити його.

ОБ’ЄКТ, ІНКАПСУЛЯЦІЯ, ІНТЕРФЕЙС, КЛАС

**Зміст**

[Вступ 2](#_Toc151681430)

[1 Опис предметної област 3](#_Toc151681431)

[1.1 У кожного студента буде індивідуальна тема для висвітлення 3](#_Toc151681432)

[1.2 Аналіз технічного завдання на роботу 4](#_Toc151681433)

[1.2.1 Функціональні вимоги 4](#_Toc151681434)

[1.2.2 Нефункціональні вимоги 4](#_Toc151681435)

[1.3 Опис алгоритму основних задач/підзадач у роботі 5](#_Toc151681436)

[Висновки до розділу 6](#_Toc151681438)

[2 Опис розробки програмного забезпечення 7](#_Toc151681439)

[2.1 Структура програмного забезпечення 7](#_Toc151681440)

[2.2 Опис роботи програми 10](#_Toc151681441)

[2.3 Функціональна схема програми 11](#_Toc151681442)

[2.4 Опис інтерфейсу програми 12](#_Toc151681443)

[Висновки до розділу 18](#_Toc151681444)

[ВИСНОВКИ 18](#_Toc151681445)

[СПИСОК ЛІТЕРАТУРИ 19](#_Toc151681446)

[Додаток А XAML розмітки вікон 20](#_Toc151681447)

[Додаток Б Тестування додатку 37](#_Toc151681448)

ВСТУП

Технологія WPF (Windows Presentation Foundation) є частиною екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

WPF є дуже поширеною платформою для розробки програмного забезпечення із графічним інтерфейсом на мові програмування C#.

Якщо при створені традиційних додатків на основі WinForms за візуалізацію елементів керування відповідали такі частини ОС Windows, як User32 та GDI+, то додатки WPF засновані на **DirectX**.В цьому і є головна відмінність рендерингу графіки у WPF: велика частина роботи по візуалізації графіки, як і звичайних кнопок, так і складних 3D-моделей, лежить на графічному процесорі на відеокарті, що також дає можливість користуватись апаратним прискоренням графіки.

Ще однією важливою особливістю WPF є використання мови декларативної розмітки XAML, заснованого на XML: ви можете створити гарний графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на керуючих мовах, як C#.

Платформа WPF підтримує багато можливостей для створення різних додатків: створення анімацій, binding, стилі, шаблони, велика кількість елементів.

Тому WPF – непоганий варіант для створення ігри «Крапки», де за допомогою анімацій, ресурсів, стилів та біндингу можна створити гарний та зрозумілий інтерфейс. А за допомогою вбудованих класів та інтерфейсів зручно робиться логіка гри.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Длегати. Лямбда-вирази. (потрібно пояснити ці терміни та показати зв’язок між ними).

### ****1. Делегати****

Делегати — це типи, які представляють методи в .NET. Вони дозволяють зберігати посилання на методи і викликати їх динамічно. У цьому вони схожі на функціональні вказівники в інших мовах програмування (наприклад, у C++).

#### ****Ключові характеристики делегатів:****

1. **Типобезпечність:** Делегат визначає сигнатуру методів, які він може зберігати.
2. **Гнучкість:** Дозволяє передавати методи як параметри в інші методи.
3. **Підтримка multicast:** Делегати можуть зберігати посилання на кілька методів, які викликаються послідовно.

### ****2. Лямбда-вирази****

Лямбда-вирази — це скорочений синтаксис для визначення анонімних методів (методів без імені). Вони використовуються для швидкого створення функцій, які можна передавати у вигляді делегатів або виразів.

#### ****Переваги лямбда-виразів:****

1. **Лаконічність:** Лямбда-вирази займають менше місця у коді, ніж традиційні методи.
2. **Гнучкість:** Часто використовуються у LINQ і делегатах для скорочення коду.
3. **Виразність:** Код стає більш зрозумілим і простим.

### ****3. Зв’язок між делегатами та лямбда-виразами****

Делегати та лямбда-вирази тісно пов’язані:

1. **Делегати як основа:** Лямбда-вирази реалізуються через делегати. Коли ви створюєте лямбда-вираз, компілятор автоматично створює делегат відповідного типу.
2. **Лямбда-вирази скорочують код:** Замість того, щоб оголошувати окремий метод і потім посилатися на нього через делегат, ви можете використати лямбда-вираз для безпосереднього визначення функції.
3. **Практичність:** У поєднанні з делегатами (наприклад, Action, Func) лямбда-вирази використовуються у LINQ, подіях, асинхронному програмуванні та багатьох інших аспектах C#.

1.2 Аналіз технічного завдання на роботу:

Мета курсового проєкту – WPF додатку «Бюро знайомств».

Над цим проектом працювала група з двох студентів, тому необхідно використовувати шаблон MVVM, задля розмежування обов’язків при виконанні проекту. Ми розподілили таким чином: я працює над зовнішнім інтерфейсом, а колега працює над моделями та моделями представлень.

Під час розробки моделей даних та графічного інтерфейсу були висунуті наступні вимоги:

**1.2.1 Функціональні вимоги:**

1. Можливість реєстрації та авторизації(користувач або адміністратор).
2. Видаляння або редагування профілю користувача.
3. Можливість зберігання даних користувачів.
4. Розмежування прав доступу: користувач має доступ до вікна свого профілю з можливістю зміни міста та отрмання нового чату, а також редагування та видалення його. Для адміністратора доступно додавати, редагувати, видаляти міста та їх посилання на чати, а також переглядати зареєстрованих користувачів.

**1.2.2 Нефункціональні вимоги:**

1. Використання патерну MVVM.
2. Усі дані повинні зберігатися у json форматі.
3. Реалізація логіки моделей та моделей вигляду.
4. Забезпечити коректність даних

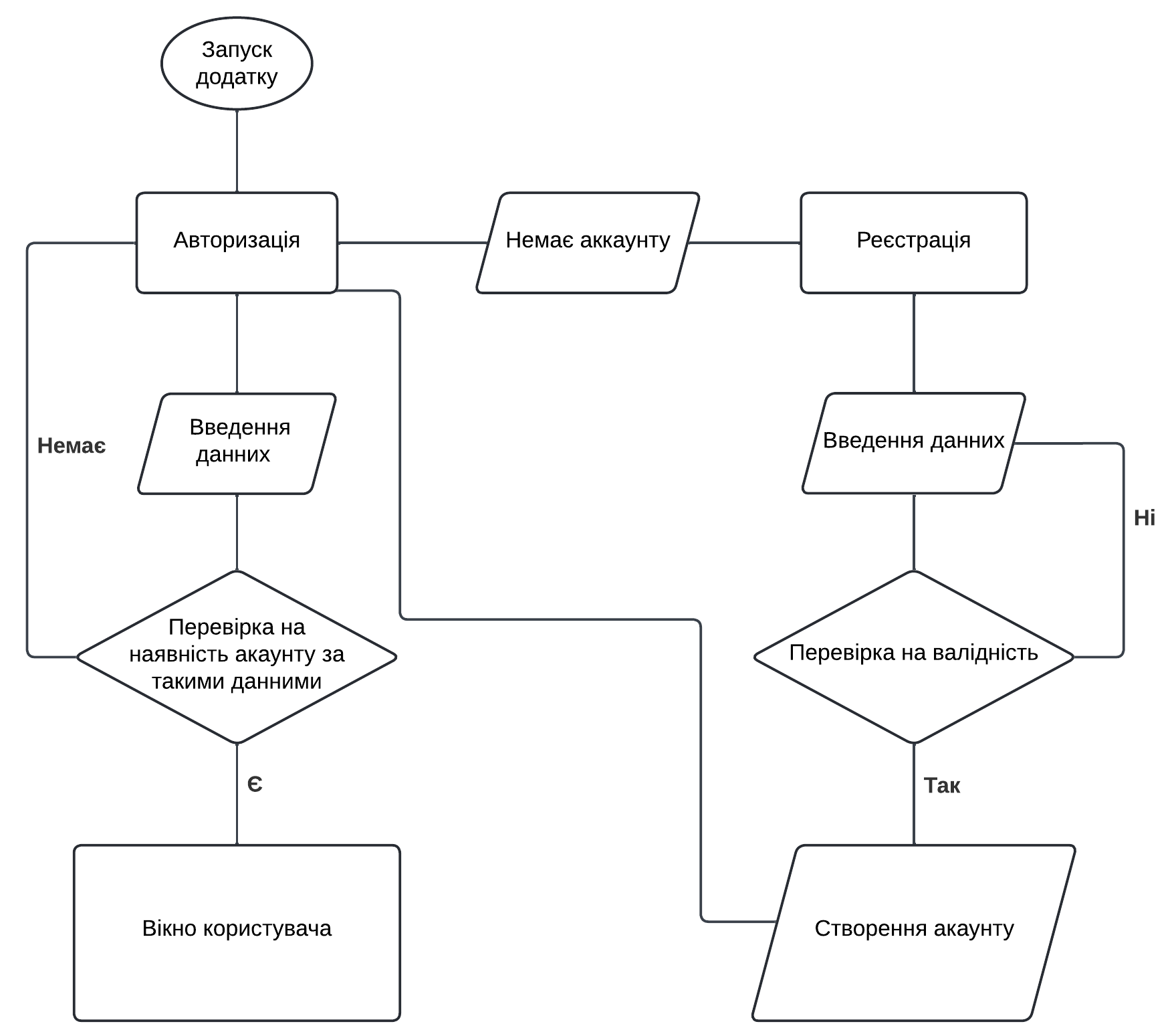
1.3 Опис алгоритму основних задач/підзадач у роботі

У рамках даної курсової роботи основною задачею є створення додатку, використовуючи патерн MVVM (рис. 1.1), так як цей додаток створюється групою, тому необхідне розмежування обов’язків. Моїми обов’язками в рамках даної роботи є створення моделей даних та моделей вигляду.



Мал. 1.1 Схема патерну MVVM

Мають бути реалізовані як мінімум 2 моделі: модель користувач, для того щоб працював механізм авторизації, реєстрації (рис 1.2) – якщо користувач має обліковий запис, то він вводить вхідні дані, з файлу зчитуються дані про усіх користувачів, якщо вхідні дані підібрано правильно, то користувач отримує доступ до основної частини, у випадку реєстрації вхідні дані користувача перевіряються на валідність, якщо всі данні введено коректно, створюється його обліковий запис та з’являється доступ до основної частини.



Мал. 1.2 Блок-схема реєстрації та авторизації

Повинно бути збереження даних у коректному вигляді, для чого потріно створити такі моделі: модель білету, без якої не можна буде реалізувати те, для чого цей додаток й створений – отримання чатів, та модель користувача, для реалізації механізмів з роботою з обліковим записом.

Висновки до розділу

У першому розділі розглянуті базові принципи ООП, створена розширена постановка задачі, проаналізована предметна область, сформовано вимоги до функціональних можливостей програми та розглянуто технічне завдання. Складений план розробки програми та функції, що повинні бути реалізовані у програмі. Також, розроблений алгоритм роботи програми.

2 ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Структура програмного забезпечення

Структура програмного забезпечення є поєднанням усіх застосованих модулів програми, що утворюють працюючу систему. Кожний модуль повинен реалізовувати функції програми, що вирішують основні завдання проєкту. Так як це груповий проект, буде висвітле тільки частина, виконана мною, а саме класи моделей вигляду та власне моделей.

Застосунок створений за допомою модулю Windows Presentation Foundation (WPF) на базі мови с#.

На рис. 2.1 зображено оглядач рішень програмного застосунку. В табл.2.1 представлено опис модулів(класів) проекту курсової роботи

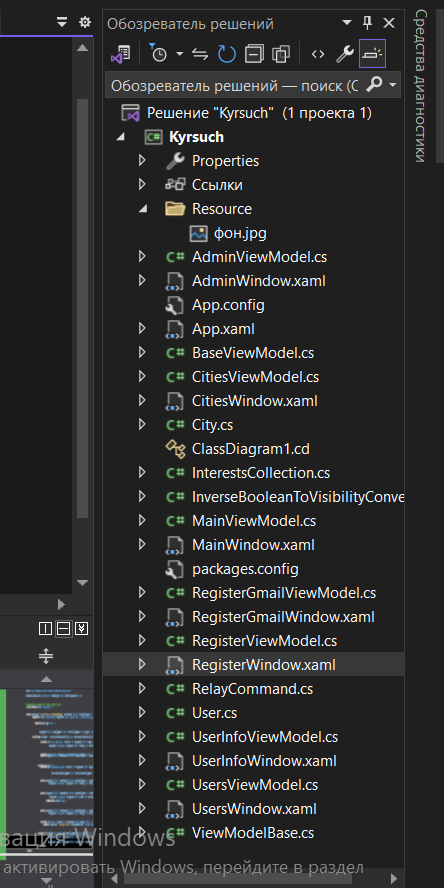


Рисунок 2.1 – Оглядач рішень

Таблиця 2.1 – Призначення модулів програми

|  |  |
| --- | --- |
| **Назва модуля** | **Призначення модуля** |
| User.cs | Клас, що зберігає статистику та дані користувача |
| City.cs | Клас, що зберігає дані про міста та посилання на чати. |
| CityCollection.cs | Клас, що відповідає за завантаженння данних з файлу та зберігання в файл. |
| RelayCommand.cs | Клас, який використовується для реалізації патерна команди (Command pattern) в програмній розробці. |
| InverseBooleanToVisibilityConverter.cs | Клас InverseBooleanToVisibilityConverter реалізує інтерфейс IValueConverter. |
| BaseViewModel.cs | Базовий клас, для класів моделей вигляду. |
| CitiesViewModel.cs | Клас який відповідає за логіку додавання, зберігання та редагування місто з їхї посиланнями. |
| AdminViewModel.cs | Клас який відповідає за логіку вибору дії адміністратора. |
| MainViewModel.cs | Клас який відповідає за логіку авторизації. |
| RegisterGmailViewModel.cs | Клас який відповідає за логіку реєстрації за поштою gmail. |
| RegisterViewModel.cs | Клас який відповідає за логіку реєстрації. |
| UserInfoViewModel.cs | Клас який відповідає за логіку виведення інформації користувачу про нього, отримання чату та можливість її редагування інформації. |

Продовження таблиці 2.1

|  |  |
| --- | --- |
| **Назва модуля** | **Призначення модуля** |
| UsersViewModel.cs | Клас який відповіає за роботу з користувачами перегляд хто заеєстрований та можливість видалення користувача |

Під час створення програми було створено 7 вікон, і відповідно до них було створено 7 моделей вигляду її діаграми зображені на рис. 2.2. Всі класи моделей вигляду наслідують клас ViewModel, який реалізує інтерфейс INotifyPropertyChanged

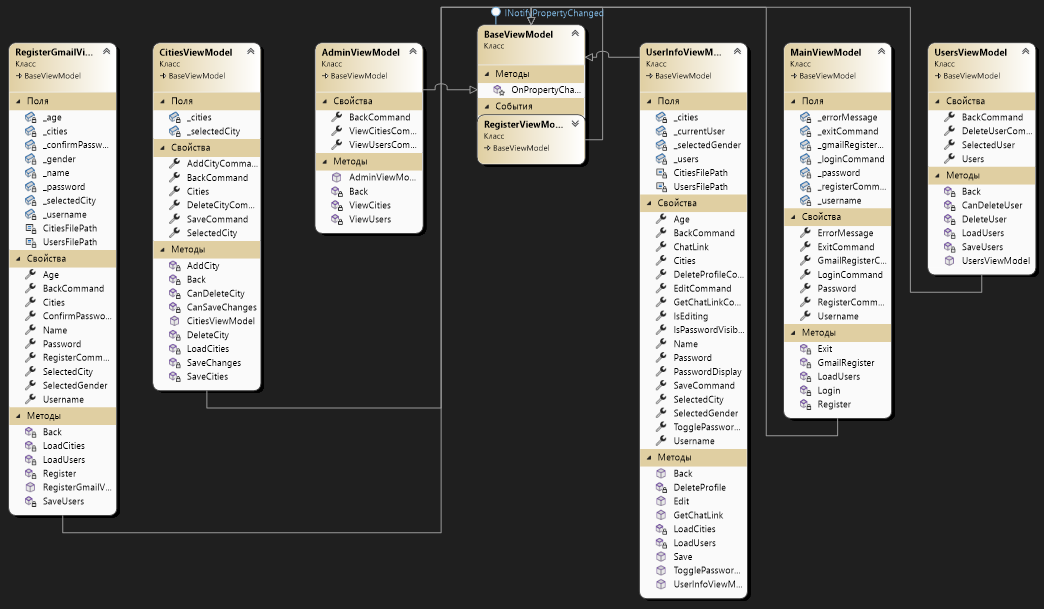


Рисунок 2.2 – Діаграма модулей вигляду

Також у всіх классах використовуются класи RelayCommand, який реалізує ICommand. На рис. 2.4 можна побачити схему моделей.

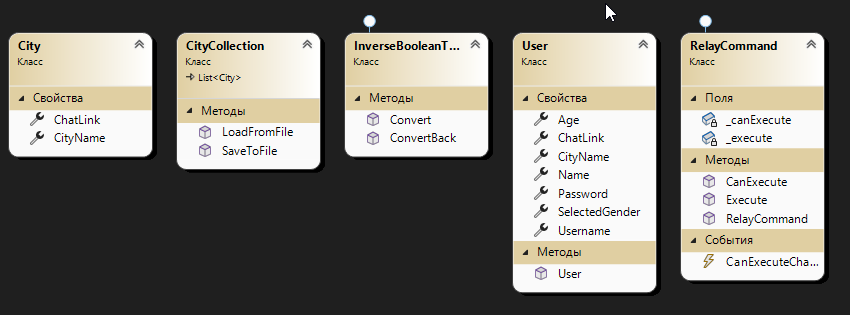


Рисунок 2.3 – Схема класів Models

2.2 Опис роботи програми

Після запуску програми відкривається вікно з авторизацією, з якого можна перейти у інші вікна такі як вікно з інформацією про користувача(Якщо є аккаунт) відкриваєтся вікно користувача де він бачить свій профіль та може його редагувати або видалити на цьому вікні він і отримує посилання на чат для спілкування за обраним містом.

-Вікно інформації показує користувачу всі данні що користувач вводив при реєстрації.

Якщо аккаунту немає користувач настискає кнопки зареєструватися(або зареєструватися через Gmail) тоді відкриваєтся вікно реєстрації де треба запомниити всі піля які також перевіряють введені данні.

-Вікно реєстрації має поля для вводу логіну, паролю , підтвердження паролю, вибору статі, введення віку та вибору міста

Якщо користувач заходить з данними адміністратора він отримує можливість перейти у два вікна.

-Вікно перегляду користувачів дає можливість переглянути зареєстрованих користувачів та видаляти їх.

-Вікно перегляду міст де адміністратор бачить міста та посилання на чати для знайомства в цих містах з можливістью редагувати, додавати та видаляти їх.

**2.3 Функціональна схема програми**

Детальний опис й пояснення процесів окремих функцій розробляємої системи утворюють функціональну схему програми (рис. 2.4).

На рис. 2.4 видно, що програма починається з авторизації. В цьому вікні користувач може ввести вхідні данні, це логін та пароль. Програма шукає, чи є користовач з таким логіном, якщо є, то перевіряється пароль. Якщо все вірно, далі іде перевірка, чи є користувач адміністратором, якщо так, то відкривається вікно адміністратора, якщо ні, то відкривається вікно користувача

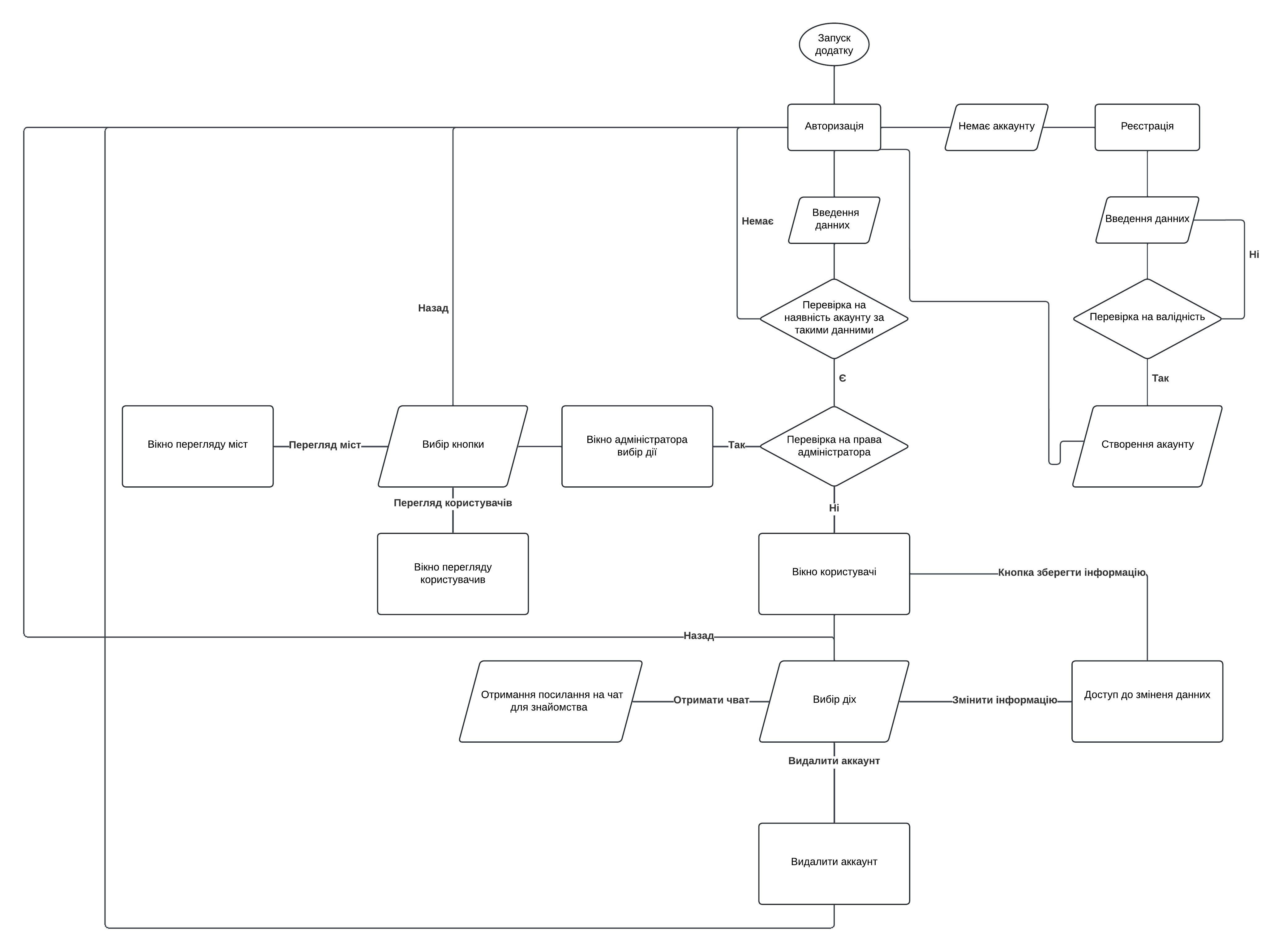
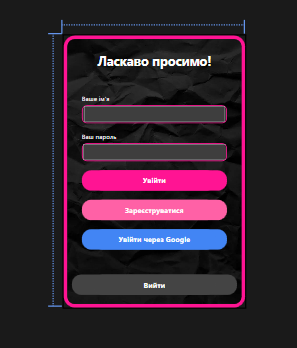


Рисунок 2.4 – Функціональна схема програмного коду

Також з вікна авторизації можна потрапити у вікно реєстрації. В ньому користувач вводить дані, і якщо дані пройшли перевірку на валідність, то на основі цих даних створюється новий акаунт. Після цього відкривається вікно з авторизацією.

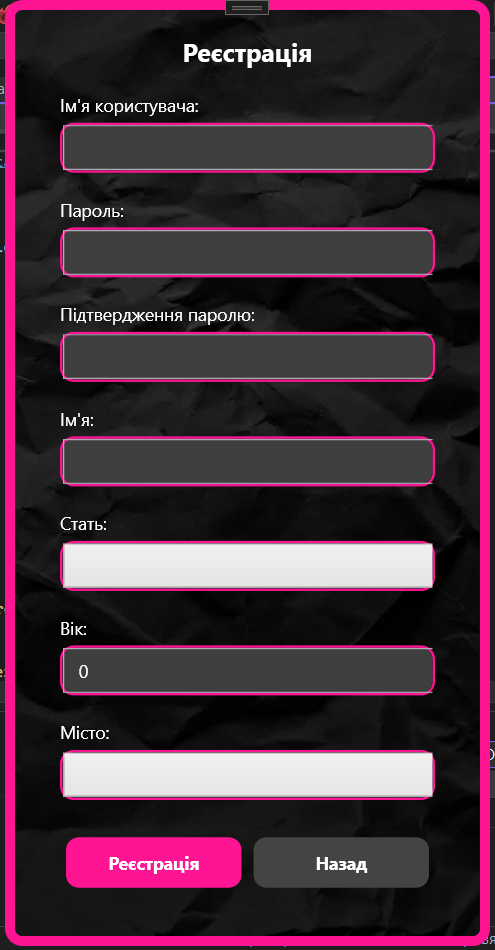
2.4 Опис інтерфейсу програми

При розробці програми створено зручний інтерфейс користувача. Опишемо основні вікна програми:



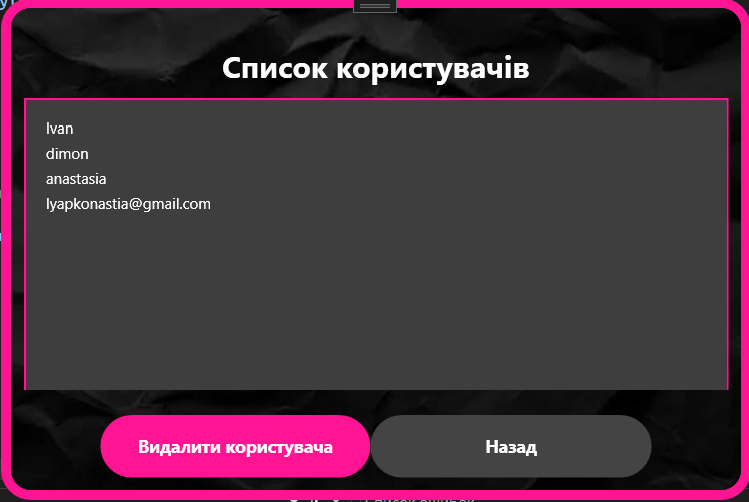
TextBlock TextBox Button Border ImageBrush PasswordBox

Рисунок 2.5 – Інтерфейс вікна MainWindow



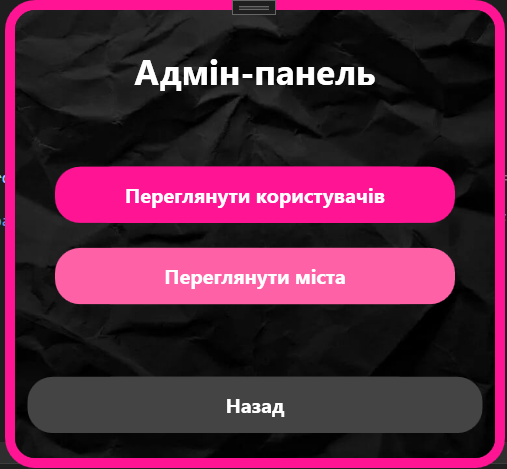
Button Border TextBlock TextBox PasswordBox ComboBox ImageBrush

Рисунок 2.6 – Вікно реєстрації



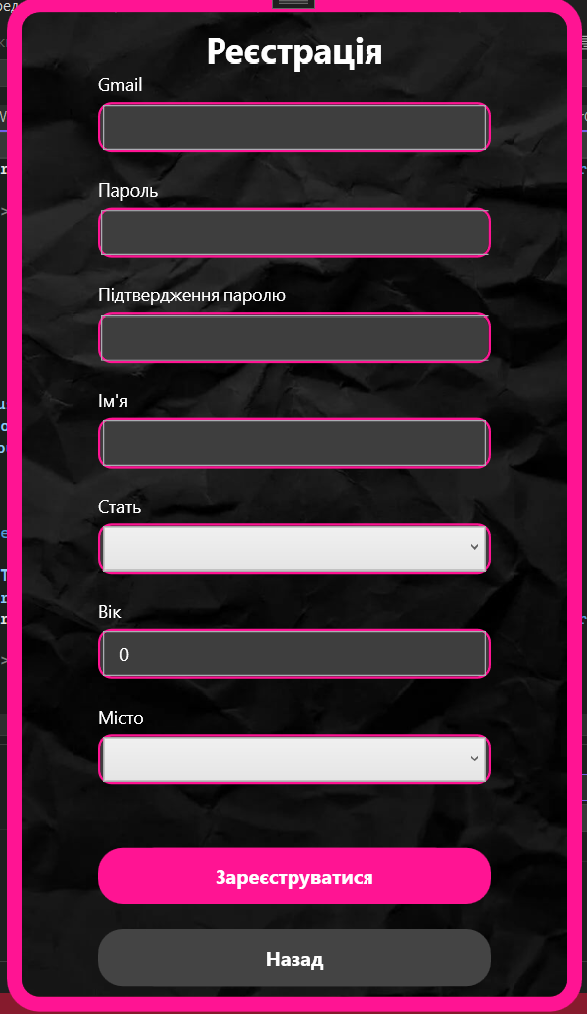
Border ImageBrush TextBlock ListBox Button

Рисунок 2.7 – Вікно адміністратора для керуванням списоком користувачів



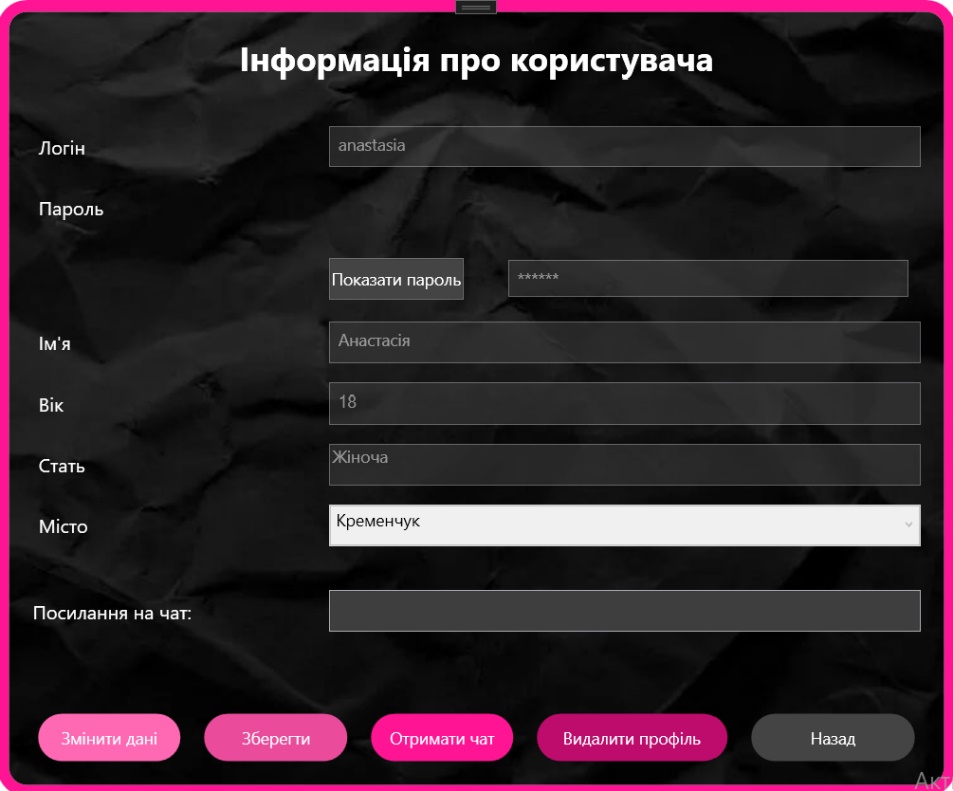
Border TextBlock Button ImageBrush

Рисунок 2.8 – Вікно адміністратора



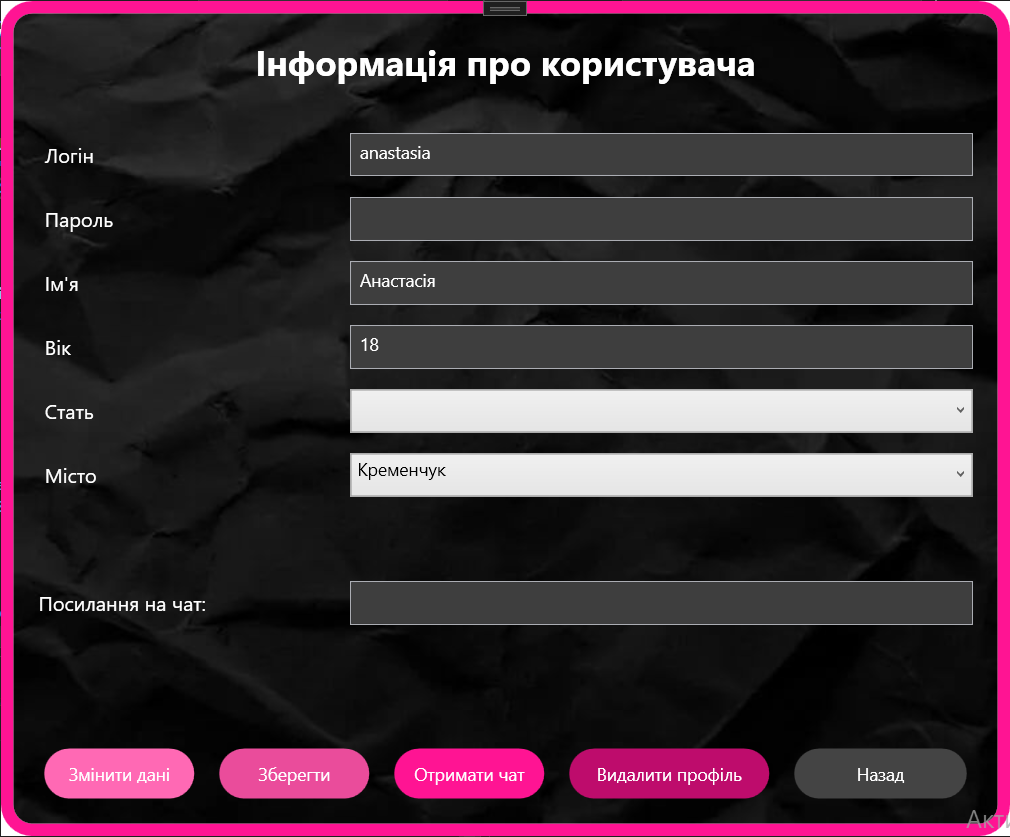
Border ImageBrush TextBlock TextBox PasswordBox ComboBox Button

Рисунок 2.9 – Вікно реєстрації через Google



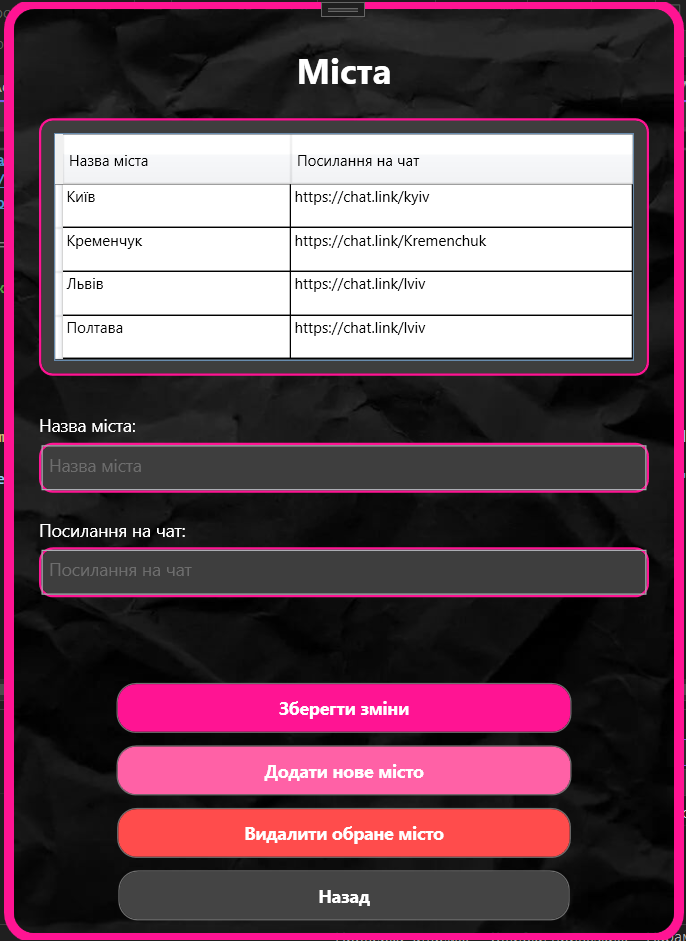
Border ImageBrush TextBlock TextBox ComboBox Label Button

Рисунок 2.10 – Вікно користувача



Border ImageBrush TextBlock TextBox PasswordBox Label ComboBox Button

Рисунок 2.11 – Вікно користувача при натискані нопки “Змінити дані”



Border TextBlock DataGrid TextBox Buttom

Рисунок 2.12 – Вікно адміністратора для керування містами

Висновки до розділу

У другому розділі розглянуті: ієрархія класів, ініціалізація даних, функціональна схема програми та опис основних функціональних вікон програми, їх призначення та властивості.

Крім того, розроблено інтерфейс програми і детально описано керівництво користувача системи.

**ВИСНОВКИ**

Метою курсової роботи є створення багатовіконного Wpf додатку, що реалізує додаток для пошуку знайомств «Бюро знайомств».

Над проектом працювала команда з двох студентів, тому обв’язки було розподілені на такі частини View– Ляпко Н.І. та ViewModel, Model– Гур’єв Д.П.

У проекті розглянуті базові принципи об'єктно-орієнтованого програмування (ООП), проведений аналіз предметної області та сформовано технічне завдання. Усі функціональні вимоги були реалізовані у коді проекту.

Проект розроблено на мові програмування C#. Основним інструментом

для створення графічного інтерфейсу використовувалась бібліотека WPF.

Створення проекту відбувалось за шаблоном MVVM. Основна увага у розробці проекту була зосереджена на моделі даних та моделях представлення. Але в проекті також реалізовано зручний інтерфейс для забезпечення комфортної роботи користувача. Прив’язка даних, що реалізована в програмному коді, є забезпеченням швидкодії обміну даними та зручним механізмом корегування та створення об’єктів моделі даних.

СПИСОК ЛІТЕРАТУРИ

1. Повне керівництво з мови програмування С# та платформі .NET 8. Інформаційний ресурс. <https://metanit.com/sharp/tutorial/>
2. Блог про патерн MVVM: [MVVM: Model-View-ViewModel Explained](https://www.mvvmcross.com/) – Все про MVVM для .NET розробників.
3. [MVVM Examples on GitHub](file:///C:\Users\lyapk\Downloads\Telegram%20Desktop\MVVM%20Examples%20on%20GitHub%20–%20Відкриті%20репозиторії%20з%20прикладами%20WPF%20додатків.) – Відкриті репозиторії з прикладами WPF додатків.
4. Офіційна документація Microsoft: [ICommand Interface](https://learn.microsoft.com/en-us/dotnet/api/system.windows.input.icommand?view=windowsdesktop-7.0) – Як створювати та використовувати команди в WPF.

Додаток А  
XAML розмітки вікон

**Файл MainWindow.xaml**

<Window x:Class="Kyrsuch.Windows.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Kyrsuch"

Title="Dating App" Height="600" Width="400" WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<Window.DataContext>

<local:MainViewModel/>

</Window.DataContext>

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Window.Resources>

<!-- Стиль для кнопок без анімації наведення -->

<Style TargetType="Button">

<Setter Property="OverridesDefaultStyle" Value="True" />

<Setter Property="Cursor" Value="Hand" />

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

CornerRadius="20"

BorderThickness="0">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid>

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="8" CornerRadius="20" Padding="10">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" FontSize="28" FontWeight="Bold" Text="Ласкаво просимо!"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,20,0,10" />

<StackPanel Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" Width="320">

<StackPanel Margin="0,10">

<TextBlock Text="Ваше ім'я" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Username}" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"/>

</Border>

</StackPanel>

<StackPanel Margin="0,10">

<TextBlock Text="Ваш пароль" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<PasswordBox x:Name="PasswordBox" PasswordChanged="PasswordBox\_PasswordChanged" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="318"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"/>

</Border>

</StackPanel>

<Border Background="#FF1493" CornerRadius="20" Margin="0,10">

<Button Content="Увійти" Command="{Binding LoginCommand}" Width="270" Height="45"

Background="#FF1493" Foreground="White" FontSize="16" />

</Border>

<Border Background="#FF61A6" CornerRadius="20" Margin="0,10">

<Button Content="Зареєструватися" Command="{Binding RegisterCommand}" Width="270" Height="45"

Background="#FF61A6" Foreground="White" FontSize="16" />

</Border>

<Border Background="#4285F4" CornerRadius="20" Margin="0,10">

<Button Content="Увійти через Google" Command="{Binding GmailRegisterCommand}" Width="270" Height="45"

Background="#4285F4" Foreground="White" FontSize="16" />

</Border>

</StackPanel>

<Border Background="#444444" CornerRadius="20" Margin="0,10" Grid.Row="2">

<Button Content="Вийти" Command="{Binding ExitCommand}" Width="270" Height="45"

Background="#444444" Foreground="White" FontSize="16" />

</Border>

</Grid>

</Border>

</Grid>

</Window>

**Файл RegisterWindow.xaml**

<Window x:Class="Kyrsuch.RegisterWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Kyrsuch"

Title="Реєстрація" Height="757" Width="388"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<Window.DataContext>

<local:RegisterViewModel />

</Window.DataContext>

<!-- Прозорий фон для вікна -->

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Window.Resources>

<!-- Стиль кнопок -->

<Style TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

CornerRadius="10"

BorderThickness="0"

BorderBrush="{TemplateBinding BorderBrush}">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Padding" Value="10,5"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Width" Value="140"/>

</Style>

</Window.Resources>

<Grid>

<!-- Основний фон -->

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="8" CornerRadius="10" Padding="10">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<!-- Заголовок -->

<TextBlock Grid.Row="0" FontSize="20" FontWeight="Bold" Text="Реєстрація"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,10,0,10" />

<!-- Вміст -->

<StackPanel Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Top" Width="300">

<!-- Поле "Ім'я користувача" -->

<StackPanel Margin="0,10">

<TextBlock Text="Ім'я користувача:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Username}" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center" />

</Border>

</StackPanel>

<!-- Поле "Пароль" -->

<StackPanel Margin="0,10">

<TextBlock Text="Пароль:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<PasswordBox FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"

PasswordChanged="PasswordBox\_PasswordChanged" />

</Border>

</StackPanel>

<!-- Поле "Підтвердження паролю" -->

<StackPanel Margin="0,10">

<TextBlock Text="Підтвердження паролю:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<PasswordBox FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"

PasswordChanged="ConfirmPasswordBox\_PasswordChanged" />

</Border>

</StackPanel>

<!-- Поле "Ім'я" -->

<StackPanel Margin="0,10">

<TextBlock Text="Ім'я:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Name}" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center" />

</Border>

</StackPanel>

<!-- Поле "Стать" -->

<StackPanel Margin="0,10">

<TextBlock Text="Стать:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<ComboBox SelectedValue="{Binding SelectedGender}" SelectedValuePath="Content"

Background="Transparent" Foreground="Black" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center">

<ComboBoxItem Content="Чоловіча" />

<ComboBoxItem Content="Жіноча" />

</ComboBox>

</Border>

</StackPanel>

<!-- Поле "Вік" -->

<StackPanel Margin="0,10">

<TextBlock Text="Вік:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Age}" FontSize="14"

Background="Transparent" Foreground="Black" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center" />

</Border>

</StackPanel>

<!-- Поле "Місто" -->

<StackPanel Margin="0,10">

<TextBlock Text="Місто:" FontSize="14" Foreground="White" Margin="0,0,0,5"/>

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<ComboBox ItemsSource="{Binding Cities}" DisplayMemberPath="CityName" SelectedItem="{Binding SelectedCity}"

Background="Transparent" Foreground="Black" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"/>

</Border>

</StackPanel>

<!-- Кнопки -->

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0,20">

<Border Background="#FF1493" CornerRadius="10" Margin="0,0,10,0">

<Button Content="Реєстрація" Command="{Binding RegisterCommand}"

Background="#FF1493" Foreground="White" Width="140" Height="40"/>

</Border>

<Border Background="#444444" CornerRadius="10">

<Button Content="Назад" Command="{Binding BackCommand}"

Background="#444444" Foreground="White" Width="140" Height="40"/>

</Border>

</StackPanel>

</StackPanel>

</Grid>

</Border>

</Grid>

</Window>

**Файл RegiserGmailWindow.xaml**

<Window x:Class="Kyrsuch.RegisterGmailWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Kyrsuch"

Title="Реєстрація" Height="810" Width="470"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<Window.DataContext>

<local:RegisterViewModel />

</Window.DataContext>

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Window.Resources>

<!-- Стиль для кнопок без зміни кольору на наведення -->

<Style TargetType="Button">

<Setter Property="OverridesDefaultStyle" Value="True" />

<Setter Property="Cursor" Value="Hand" />

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="#FF1493" />

<Setter Property="Foreground" Value="White" />

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

CornerRadius="20"

BorderThickness="0">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="12" CornerRadius="20" Padding="10" Height="805" VerticalAlignment="Center">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid Margin="0,0,0,-44">

<TextBlock FontSize="28" FontWeight="Bold" Text="Реєстрація"

Foreground="White" TextAlignment="Center" Margin="0,0,0,27" />

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Width="320">

<!-- Gmail -->

<StackPanel Margin="0,10">

<TextBlock Text="Gmail" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Username}" FontSize="14" TextChanged="TextBox\_TextChanged\_1"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"/>

</Border>

</StackPanel>

<!-- Пароль -->

<StackPanel Margin="0,10">

<TextBlock Text="Пароль" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<PasswordBox FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="318"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"

PasswordChanged="PasswordBox\_PasswordChanged" />

</Border>

</StackPanel>

<!-- Підтвердження пароля -->

<StackPanel Margin="0,10">

<TextBlock Text="Підтвердження паролю" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<PasswordBox FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="318"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"

PasswordChanged="ConfirmPasswordBox\_PasswordChanged" />

</Border>

</StackPanel>

<!-- Ім'я -->

<StackPanel Margin="0,10">

<TextBlock Text="Ім'я" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Name}" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center" />

</Border>

</StackPanel>

<!-- Стать -->

<StackPanel Margin="0,10">

<TextBlock Text="Стать" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<ComboBox SelectedValue="{Binding SelectedGender}" SelectedValuePath="Content"

Background="Transparent" Foreground="Black" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center">

<ComboBoxItem Content="Не хочу вказувати" />

<ComboBoxItem Content="Чоловік" />

<ComboBoxItem Content="Жінка" />

</ComboBox>

</Border>

</StackPanel>

<!-- Вік -->

<StackPanel Margin="0,10">

<TextBlock Text="Вік" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<TextBox Text="{Binding Age}" FontSize="14"

Background="Transparent" Foreground="White" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center" TextChanged="TextBox\_TextChanged" />

</Border>

</StackPanel>

<!-- Місто -->

<StackPanel Margin="0,10">

<TextBlock Text="Місто" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<ComboBox ItemsSource="{Binding Cities}" DisplayMemberPath="CityName" SelectedItem="{Binding SelectedCity}"

Background="Transparent" Foreground="Black" Padding="10,0" Width="312"

HorizontalContentAlignment="Left" VerticalContentAlignment="Center"/>

</Border>

</StackPanel>

<!-- Кнопки -->

<StackPanel Margin="0,30,0,5">

<Border Background="#FF1493" CornerRadius="20" Margin="0,10">

<Button Content="Зареєструватися" Command="{Binding RegisterCommand}" Width="270" Height="45"

Background="#FF1493" Foreground="White" FontSize="16" />

</Border>

<Border Background="#444444" CornerRadius="20" Margin="0,10">

<Button Content="Назад" Command="{Binding BackCommand}" Width="270" Height="45"

Background="#444444" Foreground="White" FontSize="16" />

</Border>

</StackPanel>

</StackPanel>

</Grid>

</Border>

</Window>

**Файл UserInfoWindow.xaml**

<Window x:Class="Kyrsuch.Windows.UserInfoWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Kyrsuch"

Title="Інформація про користувача" Height="668" Width="807"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<Window.Resources>

<BooleanToVisibilityConverter x:Key="BooleanToVisibilityConverter"/>

<local:InverseBooleanToVisibilityConverter x:Key="InverseBooleanToVisibilityConverter"/>

<!-- Загальний стиль для кнопок -->

<Style TargetType="Button" x:Key="RoundedButtonStyle">

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Width" Value="120"/>

<Setter Property="Margin" Value="10"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

CornerRadius="20"

BorderThickness="0">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

<Setter Property="Cursor" Value="Hand"/>

</Style>

<!-- Відтінки рожевого і сірий для кнопок -->

<SolidColorBrush x:Key="PinkLight" Color="#FFC0CB"/>

<!-- Світло-рожевий -->

<SolidColorBrush x:Key="PinkMedium" Color="#FF69B4"/>

<!-- Рожевий -->

<SolidColorBrush x:Key="PinkDark" Color="#FF1493"/>

<!-- Темно-рожевий -->

<SolidColorBrush x:Key="PinkSoft" Color="#F8BBD0"/>

<!-- М'який рожевий -->

<SolidColorBrush x:Key="GrayButton" Color="#A9A9A9"/>

<!-- Сірий -->

<!-- Стиль для кнопки "Зберегти" -->

<Style TargetType="Button" x:Key="SaveButtonStyle" BasedOn="{StaticResource RoundedButtonStyle}">

<Setter Property="Background" Value="{StaticResource PinkMedium}"/>

<Style.Triggers>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="Background" Value="{StaticResource PinkMedium}"/>

<!-- Сірий для неактивної кнопки -->

<Setter Property="Foreground" Value="White"/>

</Trigger>

</Style.Triggers>

</Style>

</Window.Resources>

<Grid>

<!-- Рамка з фоном -->

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="10" CornerRadius="20" Padding="10">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<!-- Заголовок -->

<TextBlock Grid.Row="0" FontSize="28" FontWeight="Bold" Text="Інформація про користувача"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,10,0,20" />

<!-- Поля даних -->

<Grid Grid.Row="1" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*" />

<ColumnDefinition Width="2\*" />

</Grid.ColumnDefinitions>

<!-- Логін -->

<Label Content="Логін" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="0" Grid.Column="0" Margin="0,10,10,10" />

<TextBox Text="{Binding Username}" FontSize="14" Background="#3E3E3E" Foreground="White" Padding="5"

IsEnabled="{Binding IsEditing}" Grid.Row="0" Grid.Column="1" Height="35" />

<!-- Пароль -->

<Label Content="Пароль" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="1" Grid.Column="0" Margin="0,10,10,10" />

<PasswordBox x:Name="PasswordBox" Background="#3E3E3E" Foreground="White" Padding="5"

IsEnabled="{Binding IsEditing}" PasswordChanged="PasswordBox\_PasswordChanged" Visibility="{Binding IsEditing, Converter={StaticResource BooleanToVisibilityConverter}}" Grid.Row="1" Grid.Column="1" Height="35" />

<Button Content="Показати пароль" Command="{Binding TogglePasswordVisibilityCommand}" Visibility="{Binding IsEditing, Converter={StaticResource InverseBooleanToVisibilityConverter}}"

Background="#444444" Foreground="White" FontSize="14" Margin="0,16,0,10"

Grid.Row="2" Grid.Column="1" Height="35" HorizontalAlignment="Left" />

<TextBox Text="{Binding PasswordDisplay, Mode=OneWay}" FontSize="14" Background="#3E3E3E" Foreground="White" Padding="5"

IsReadOnly="True" Grid.Row="2" Grid.Column="1" Margin="151,18,10,13" IsEnabled="False" Visibility="{Binding IsEditing, Converter={StaticResource InverseBooleanToVisibilityConverter}}"/>

<!-- Ім'я -->

<Label Content="Ім'я" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="3" Grid.Column="0" Margin="0,10,10,10" />

<TextBox Text="{Binding Name}" FontSize="14" Background="#3E3E3E" Foreground="White" Padding="5"

IsEnabled="{Binding IsEditing}" Grid.Row="3" Grid.Column="1" Height="35" />

<!-- Вік -->

<Label Content="Вік" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="4" Grid.Column="0" Margin="0,10,10,10" />

<TextBox Text="{Binding Age}" FontSize="14" Background="#3E3E3E" Foreground="White" Padding="5"

IsEnabled="{Binding IsEditing}" Grid.Row="4" Grid.Column="1" Height="35" />

<!-- Стать -->

<Label Content="Стать" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="5" Grid.Column="0" Margin="0,10,10,10" />

<TextBox Text="{Binding SelectedGender}" FontSize="14" Background="#3E3E3E" Foreground="White" IsEnabled="False" Visibility="{Binding IsEditing, Converter={StaticResource InverseBooleanToVisibilityConverter}}" TextChanged="TextBox\_TextChanged" Grid.Row="5" Grid.Column="1" Height="35"/>

<ComboBox SelectedValue="{Binding SelectedGender, Mode=TwoWay}" FontSize="14" Background="#3E3E3E" Foreground="Black" IsEnabled="{Binding IsEditing}" Visibility="{Binding IsEditing, Converter={StaticResource BooleanToVisibilityConverter}}" SelectedValuePath="Tag" Grid.Row="5" Grid.Column="1" Height="35">

<ComboBoxItem Content="Жінка" Tag="Жінка" />

<ComboBoxItem Content="Чоловік" Tag="Чоловік" />

</ComboBox>

<!-- Місто -->

<Label Content="Місто" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="6" Grid.Column="0" Margin="0,10,10,10" />

<ComboBox SelectedItem="{Binding SelectedCity}" DisplayMemberPath="CityName" FontSize="14" Background="#3E3E3E" Foreground="Black"

IsEnabled="{Binding IsEditing}" Grid.Row="6" Grid.Column="1" Height="35" >

<ComboBox.ItemsSource>

<Binding Path="Cities"/>

</ComboBox.ItemsSource>

</ComboBox>

<!-- Посилання на чат -->

<TextBlock Text="Посилання на чат:" FontSize="16" Foreground="White" VerticalAlignment="Center" Grid.Row="7" Grid.Column="0" Margin="0,10,10,10" />

<TextBox Text="{Binding ChatLink}" FontSize="14" Background="#3E3E3E" Foreground="White" Padding="5"

IsReadOnly="True" Grid.Row="7" Grid.Column="1" Height="35" />

</Grid>

<!-- Кнопки -->

<StackPanel Grid.Row="2" Orientation="Horizontal" HorizontalAlignment="Center" Margin="0,20,0,0">

<Button Content="Змінити дані" Command="{Binding EditCommand}" Style="{StaticResource RoundedButtonStyle}" Background="{DynamicResource PinkMedium}" />

<Button Content="Зберегти" Command="{Binding SaveCommand}" Style="{StaticResource SaveButtonStyle}" IsEnabled="{Binding IsEditing}" Background="#FFEA4C9B" />

<Button Content="Отримати чат" Command="{Binding GetChatLinkCommand}" Style="{StaticResource RoundedButtonStyle}" Background="{StaticResource PinkDark}" />

<Button Content="Видалити профіль" Command="{Binding DeleteProfileCommand}" Style="{StaticResource RoundedButtonStyle}" Background="#FFBE0C6C" Width="160" />

<Button Content="Назад" Command="{Binding BackCommand}" Style="{StaticResource RoundedButtonStyle}" Background="#444444" Click="Button\_Click\_1" Width="138" />

</StackPanel>

</Grid>

</Border>

</Grid>

</Window>

**Файл AdminWindow.xaml**

<Window x:Class="Kyrsuch.AdminWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Admin Window" Height="374" Width="400"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<!-- Основний фон: прозорий для створення ефекту рамки -->

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Grid>

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="8" CornerRadius="20" Padding="10">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<!-- Заголовок -->

<TextBlock Grid.Row="0" FontSize="28" FontWeight="Bold" Text="Адмін-панель"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,20,0,10" />

<!-- Вміст -->

<StackPanel Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center" Width="320">

<!-- Переглянути користувачів -->

<Border Background="#FF1493" CornerRadius="20" Margin="0,10">

<Button Content="Переглянути користувачів" Command="{Binding ViewUsersCommand}" Width="270" Height="45"

Background="#FF1493" Foreground="White" FontSize="16" FontWeight="Bold"

BorderThickness="0">

<Button.Style>

<Style TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Button.Style>

</Button>

</Border>

<!-- Переглянути міста -->

<Border Background="#FF61A6" CornerRadius="20" Margin="0,10">

<Button Content="Переглянути міста" Command="{Binding ViewCitiesCommand}" Width="270" Height="45"

Background="#FF61A6" Foreground="White" FontSize="16" FontWeight="Bold"

BorderThickness="0">

<Button.Style>

<Style TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Button.Style>

</Button>

</Border>

</StackPanel>

<!-- Кнопка "Назад" -->

<Border Background="#444444" CornerRadius="20" Margin="0,10" Grid.Row="2">

<Button Content="Назад" Command="{Binding BackCommand}" Width="270" Height="45"

Background="#444444" Foreground="White" FontSize="16" FontWeight="Bold"

BorderThickness="0">

<Button.Style>

<Style TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}" CornerRadius="20">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Button.Style>

</Button>

</Border>

</Grid>

</Border>

</Grid>

</Window>

**Файл CitiesWindow.xaml**

<Window x:Class="Kyrsuch.CitiesWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

Title="Міста" Height="750" Width="544"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<!-- Основний фон: прозорий для створення ефекту рамки -->

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Grid>

<!-- Рамка з фоном -->

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="8" CornerRadius="20" Padding="10" Height="755" VerticalAlignment="Center">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<!-- Заголовок -->

<TextBlock Grid.Row="0" FontSize="28" FontWeight="Bold" Text="Міста"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,20,0,10" />

<!-- Основний контент -->

<StackPanel Grid.Row="1" Margin="10" VerticalAlignment="Top">

<!-- Таблиця міст -->

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10">

<DataGrid x:Name="CitiesDataGrid" AutoGenerateColumns="False"

CanUserAddRows="False" CanUserDeleteRows="True"

ItemsSource="{Binding Cities}"

SelectedItem="{Binding SelectedCity}"

Margin="10" Background="Transparent" Foreground="Black"

ColumnHeaderHeight="40" RowHeight="35">

<DataGrid.Columns>

<DataGridTextColumn Header="Назва міста" Binding="{Binding CityName}" Width="2\*" />

<DataGridTextColumn Header="Посилання на чат" Binding="{Binding ChatLink}" Width="3\*" />

</DataGrid.Columns>

</DataGrid>

</Border>

<!-- Поля введення -->

<StackPanel Margin="0,20,0,0">

<StackPanel Margin="0,10">

<TextBlock Text="Назва міста:" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<xctk:WatermarkTextBox Text="{Binding SelectedCity.CityName}"

Watermark="Назва міста"

Background="Transparent" Foreground="White"

Padding="5" FontSize="14" />

</Border>

</StackPanel>

<StackPanel Margin="0,10">

<TextBlock Text="Посилання на чат:" FontSize="14" Foreground="White" Margin="0,0,0,5" />

<Border Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2" CornerRadius="10" Height="40">

<xctk:WatermarkTextBox Text="{Binding SelectedCity.ChatLink}"

Watermark="Посилання на чат"

Background="Transparent" Foreground="White"

Padding="5" FontSize="14" />

</Border>

</StackPanel>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" HorizontalAlignment="Center" Margin="0,20,0,0">

<!-- Додавання стилю для кнопок -->

<StackPanel.Resources>

<Style TargetType="Button">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="15"

Padding="5">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</StackPanel.Resources>

<!-- Кнопки з відступами -->

<Button Content="Зберегти зміни" Command="{Binding SaveCommand}" Width="364" Height="40"

Background="#FF1493" Foreground="White" FontSize="14" FontWeight="Bold" Margin="0,5,0,5" />

<Button Content="Додати нове місто" Command="{Binding AddCityCommand}" Width="364" Height="40"

Background="#FF61A6" Foreground="White" FontSize="14" FontWeight="Bold" Margin="0,5,0,5" />

<Button Content="Видалити обране місто" Command="{Binding DeleteCityCommand}" Width="363" Height="40"

Background="#FF4C4C" Foreground="White" FontSize="14" FontWeight="Bold" Margin="0,5,0,5" />

<Button Content="Назад" Command="{Binding BackCommand}" Width="361" Height="40"

Background="#444444" Foreground="White" FontSize="14" FontWeight="Bold" Margin="0,5,0,0" />

</StackPanel>

</Grid>

</Border>

</Grid>

</Window>

**Файл UsersWindow.xaml**

<Window x:Class="Kyrsuch.UsersWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Users" Height="400" Width="600"

WindowStartupLocation="CenterScreen" WindowStyle="None" AllowsTransparency="True">

<!-- Прозорий фон для створення ефекту рамки -->

<Window.Background>

<SolidColorBrush Color="Transparent" />

</Window.Background>

<Grid>

<!-- Рамка з фоном -->

<Border Background="{DynamicResource BackgroundImage}" BorderBrush="#FF1493" BorderThickness="8" CornerRadius="20" Padding="10">

<Border.Resources>

<ImageBrush x:Key="BackgroundImage" ImageSource="pack://application:,,,/Resource/фон.jpg" Stretch="UniformToFill" />

</Border.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<!-- Заголовок -->

<TextBlock Grid.Row="0" FontSize="24" FontWeight="Bold" Text="Список користувачів"

Foreground="White" TextAlignment="Center" VerticalAlignment="Top" Margin="0,20,0,10" />

<!-- Список користувачів -->

<ListBox Grid.Row="1" ItemsSource="{Binding Users}" DisplayMemberPath="Username"

SelectedItem="{Binding SelectedUser}" Height="300"

Background="#3E3E3E" BorderBrush="#FF1493" BorderThickness="2"

Foreground="White" Padding="10" HorizontalContentAlignment="Left" />

<!-- Кнопки -->

<StackPanel Grid.Row="2" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Bottom" Margin="0,20,0,0">

<!-- Кнопка "Видалити користувача" -->

<Button Content="Видалити користувача" Command="{Binding DeleteUserCommand}" Width="216" Height="50"

Background="#FF1493" Foreground="White" FontSize="14" FontWeight="Bold"

BorderThickness="0" VerticalAlignment="Center">

<Button.Template>

<ControlTemplate TargetType="Button">

<Border Background="#FF1493" CornerRadius="25" >

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

<!-- Кнопка "Назад" -->

<Button Content="Назад" Command="{Binding BackCommand}" Width="225" Height="50"

Background="#444444" Foreground="White" FontSize="14" FontWeight="Bold"

BorderThickness="0" VerticalAlignment="Center">

<Button.Template>

<ControlTemplate TargetType="Button">

<Border Background="#444444" CornerRadius="25">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</StackPanel>

</Grid>

</Border>

</Grid>

</Window>

**Файл App.xaml**

<Application x:Class="Kyrsuch.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Kyrsuch"

StartupUri="Windows/MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

Додаток Б

Тестування додатку

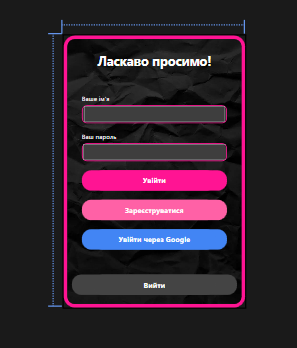


Рисунок 1 – Головне меню

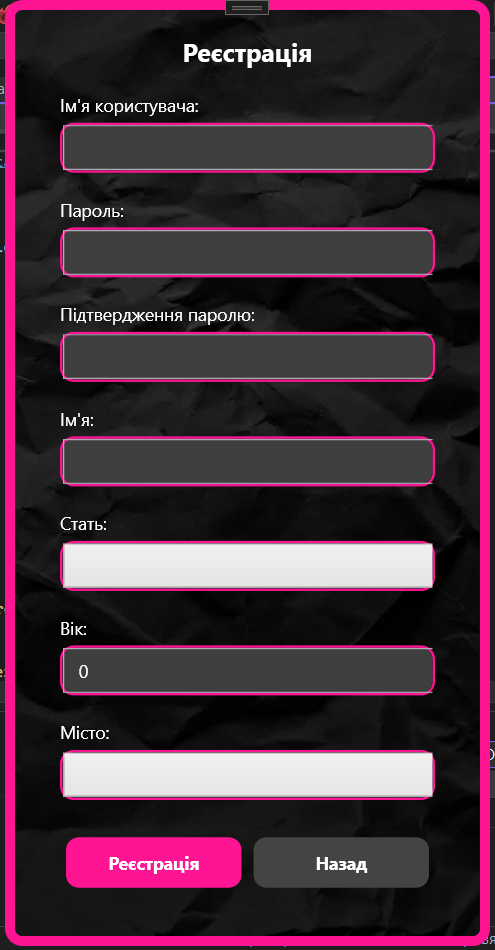


Рисунок 2 – Вікно реєстрації нового користувача

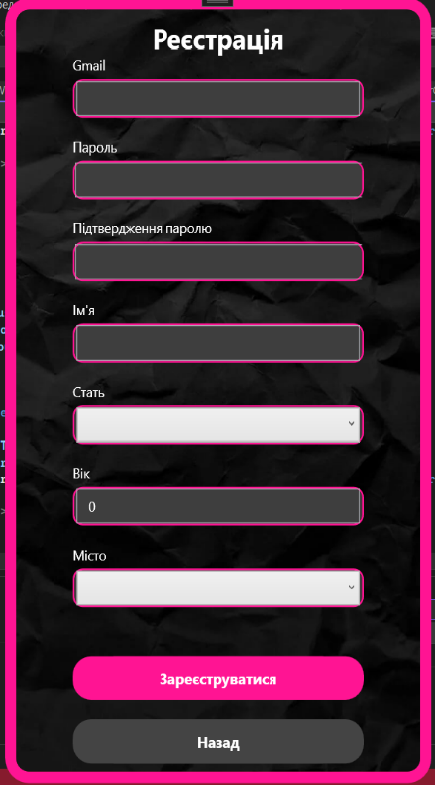


Рисунок 3 – Вікно реєстрації Google через нових користувачів

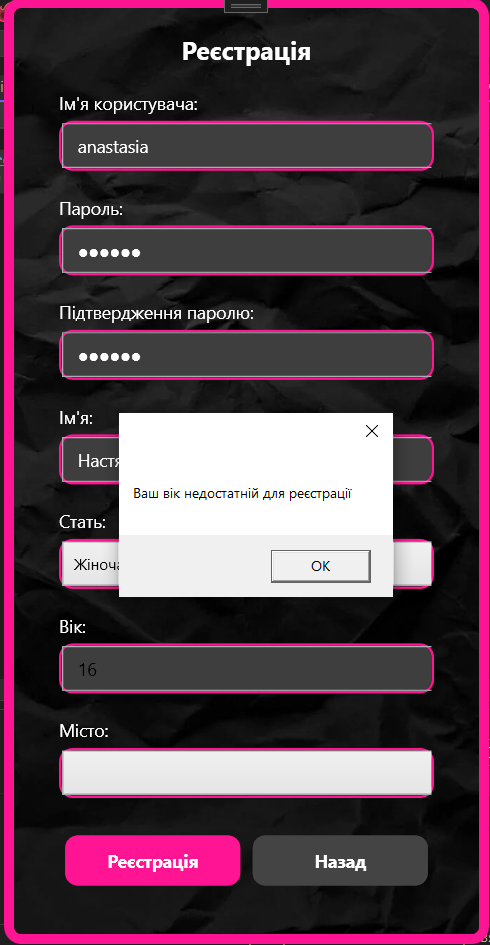


Рисунок 4 – Помилка при введені віку менше 18 років під час реєстрації

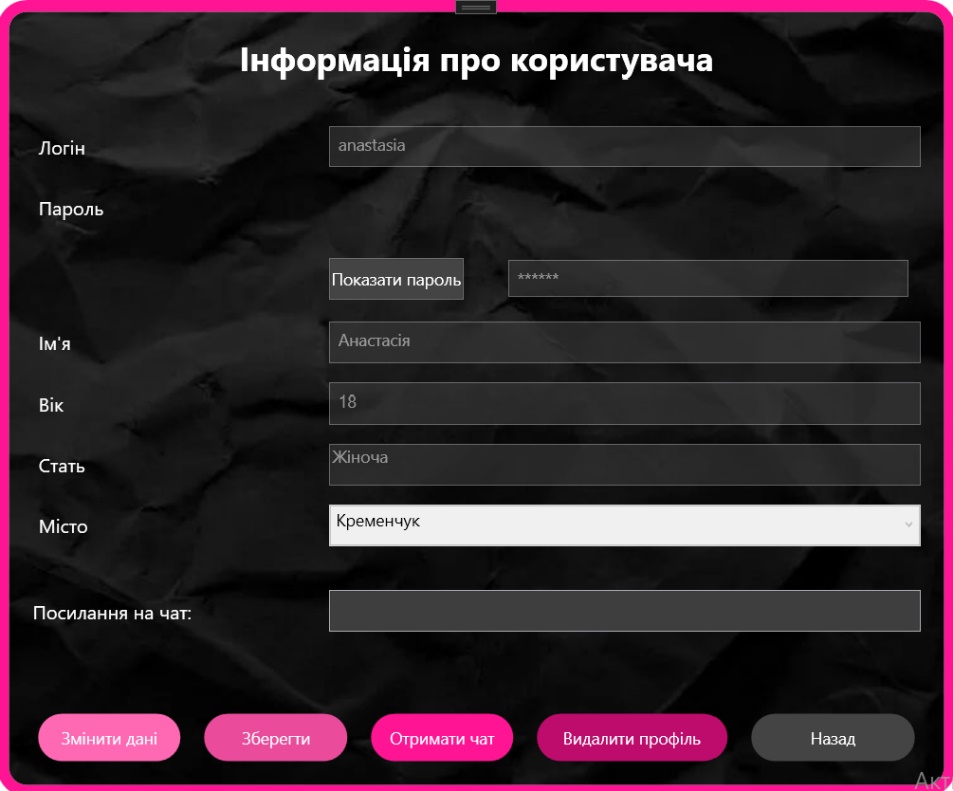


Рисунок 5 – Вікно користувача

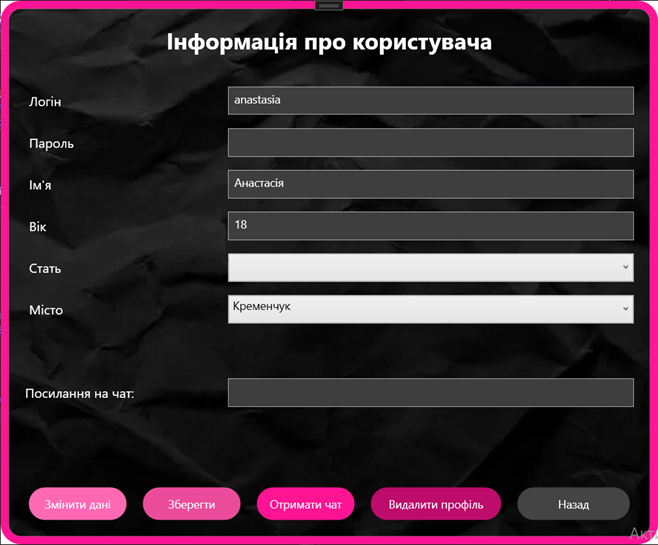


Рисунок 6 – Вікно користувача при натисканні кнопки “Змінити дані “

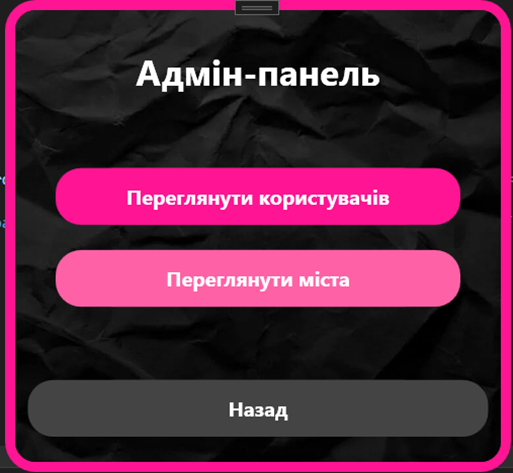


Рисунок 7 – Вікно адміністратора

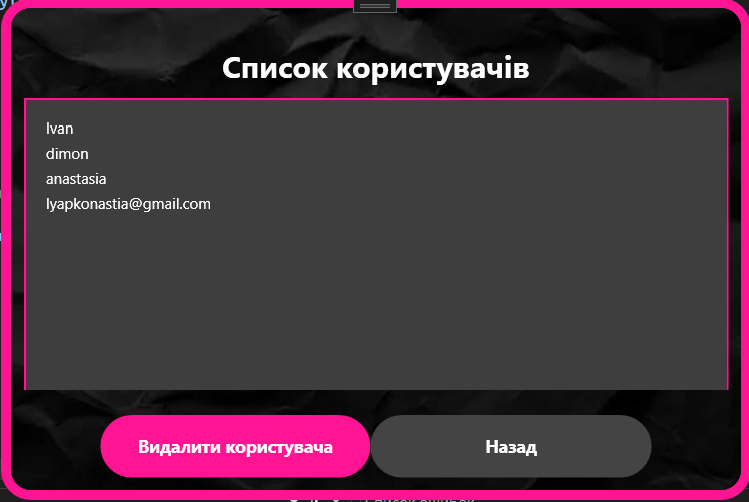


Рисунок 8 – Вікно адмістратора зі списком користувачів

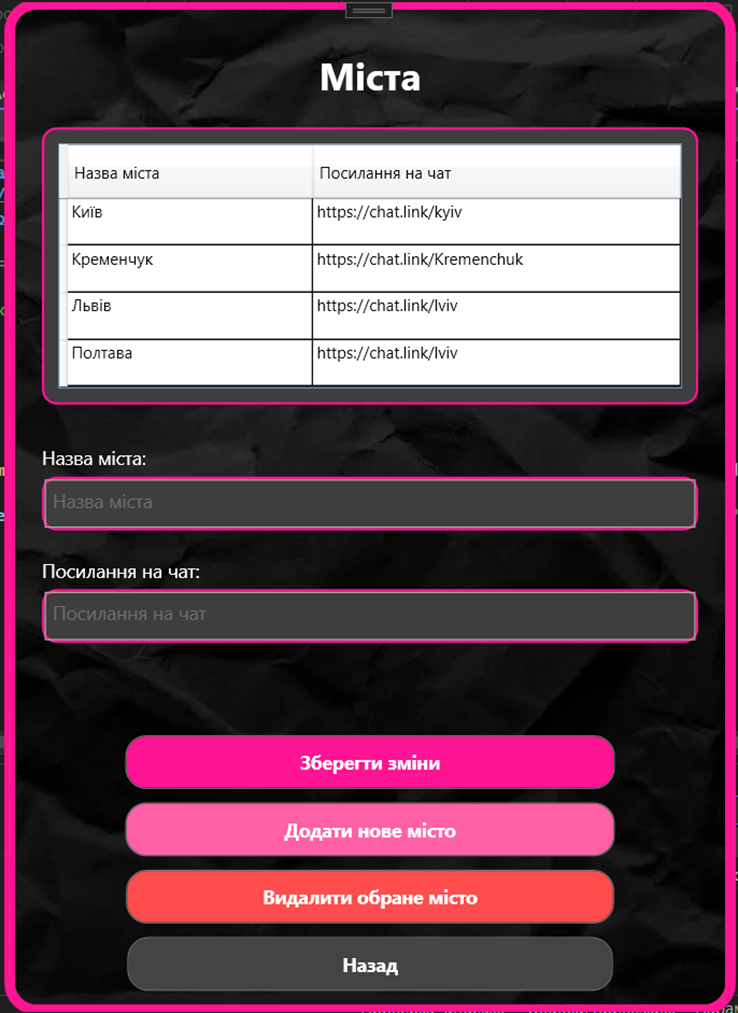


Рисунок 9 – Вікно адміністратора для редагування міст