

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 Картотека Інтерполу

Студента 2 курсу КН-22-1 групи

Ступінь вищої освіти «Бакалавр»

(бакалавр, магістр)

Спеціальність 122 – «Комп'ютерні науки»

Освітньо-професійна програма

«Комп'ютерні науки»

Ходаковський О.В

(прізвище та ініціали)

Керівник старший викладач кафедри АІС

Бельська В. Ю.

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

Національна шкала _____

Кількість балів: _____. Оцінка: ЄКТС ____

Члени комісії

(підпис)

В. Ю. Бельська

(ініціали та прізвище)

(підпис)

І. Г. Оксанич

(ініціали та прізвище)

(підпис)

В. В. Найда

(ініціали та прізвище)

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

Кафедра автоматизації та інформаційних систем

Дисципліна «Сучасні мови об'єктно-орієнтованого програмування»

Освітній ступінь «Бакалавр»

Спеціальність 122 – «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

Курс 2 група КН-22-1 семестр 3

ЗАВДАННЯ
НА КУРСОВИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ

Ходаковського Олексія Володимировича

(прізвище, ім'я, по-батькові)

1. Тема роботи: Картотека Інтерполу
2. Термін здачі студентом роботи 1 грудня 2023 р
3. Вихідні дані до роботи:
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):
постановка завдання, аналіз технічного завдання, розробка алгоритму роботи
програми, опис моделі даних, структура програмного забезпечення,
функціональна схема та інтерфейс програми.
5. Перелік графічного матеріалу:
6. Дата видачі завдання: 1 жовтня 2023 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 _____
(підпис)

_____ **В. Ю. Бельська** _____
(ініціали та прізвище)

«1» жовтня 2023 р.

РЕФЕРАТ

Курсова робота містить:

Об'єкт розробки – WPF додаток «Картотека Інтерполу».

Мета: створення багатовіконного Wpf додатку з використанням патерну MVVM, що реалізує картотеку Інтерполу.

Під час виконання завдання, поставленого на курсову роботу, було визначено основні функціональні та не функціональні вимоги до додатку, що створюється. Побудована модель даних, що дозволить вирішити наступні задачі: розробка алгоритму реєстрації та авторизації користувача, створення віконного інтерфейсу задля графічного відображення інформації.

В якості мови програмування для створення WPF додатку використовувалася мова C# та IDE Visual Studio 2022 року.

Результатом виконання всіх етапів є застосунок, який може працювати з інформацією про злочинців та кримінальні угруповання, можливість авторизації у системі.

ОБ'ЄКТ, ІНКАПСУЛЯЦІЯ, ІНТЕРФЕЙС, КЛАС

ЗМІСТ

ВСТУП.....	2
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	3
1.1 Делегати. Стандартні делегати. Для чого використовуються.....	3
1.2 Аналіз технічного завдання на роботу.....	4
1.2.1 Функціональні вимоги	5
1.2.2 Нефункціональні вимоги	5
1.3 Опис алгоритму основних задач/підзадач у роботі	5
1.3.1 Структура програмного забезпечення	7
Висновки до розділу.....	7
2 ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	8
2.1 Структура програмного забезпечення.....	8
2.2 Опис роботи програми	14
2.3 Функціональна схема програми	15
2.4 Опис інтерфейсу програми.....	16
Висновки до розділу.....	20
ВИСНОВКИ	21
СПИСОК ЛІТЕРАТУРИ	22
Додаток А XAML розмітки вікон	23
Додаток В C# основні класи	57
Додаток Г Тестування додатку.....	73

					122 – КР.2023.01.000 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Картотека Інтерполу Пояснювальна записка			Літ.	Арк.	Аркушів
Розроб.		Ходаковський О.								
Перевір.		Бельська В. Ю							1	83
								КрНУ Кафедра АІС		
Н. контр.		Найда В. В.								

ВСТУП

Технологія WPF (Windows Presentation Foundation) є важливою складовою екосистеми платформи .NET, забезпечуючи потужну платформу для створення графічних інтерфейсів програм. Вона є підсистемою, яка дозволяє конструювати інтерфейси з високим рівнем гнучкості та стилю.

WPF стала популярною серед розробників завдяки своїй ефективності в розробці програм з графічним інтерфейсом за допомогою мови програмування C#. У порівнянні з традиційними додатками, що базуються на WinForms, де візуалізація елементів керування залежить від частин ОС Windows, таких як User32 та GDI+, WPF використовує базування на DirectX. Це призводить до ключової відмінності у рендерингу графіки: більшість роботи з візуалізації, навіть для простих кнопок або складних 3D-моделей, здійснюється на графічному процесорі відеокарти, що дає можливість використовувати апаратне прискорення графіки.

Однією з ключових особливостей WPF є використання декларативної мови розмітки XAML, що базується на XML. Це дозволяє створювати елегантний графічний інтерфейс через декларативні описи або програмно з мовами керування, такими як C#.

Платформа WPF має багато функціональних можливостей для створення різноманітних додатків: анімації, байндинг, стилі, шаблони та величезний асортимент елементів.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Делегати. Стандартні делегати. Для чого використовуються

У світі програмування делегати - це об'єкти, які вказують на методи. Вони дають можливість передавати метод як параметр іншого методу, а також зберігати його в змінній. Це потужний механізм, який забезпечує гнучкість та розширюваність коду.

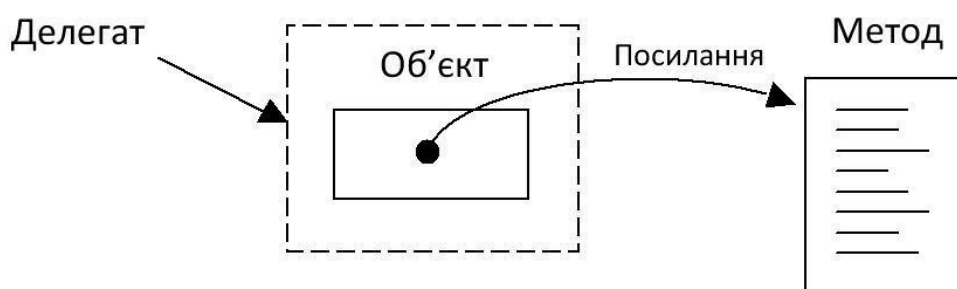


Рисунок 1.1 – Призначення делегату

Як вони працюють?

- Вказівники на методи: Делегати функціонують як вказівники на методи. Вони дозволяють передавати метод як параметр іншому методу або зберігати його в колекції.
- Типобезпечність: Вони є типобезпечними, оскільки визначають типи параметрів та повернення.
- Події: Використовуються для реалізації подій в С#. Події дозволяють класам повідомляти про події, які відбуваються у їхньому внутрішньому стані.

Для чого використовуються делегати?

- Колбеки (Callback): Вони дозволяють передавати функцію або метод в якості параметру іншої функції. Це корисно для обробки результатів асинхронних операцій або зворотного виклику в подіях.
- Реалізація подій: Вони дозволяють створювати та обробляти події, які виникають у класах.

- Розподілена робота: Делегати дозволяють виконувати виклики методів через мережу або між додатками, що дозволяє реалізувати розподілену роботу.
- Заміна анонімних методів: Делегати можуть бути використані для передачі анонімних методів, що полегшує створення простих обробників подій.
- Інверсія управління: Вони дозволяють реалізувати принцип інверсії управління, коли контроль за виконанням певних дій передається іншому об'єкту.

Передбудовані делегати в .NET:

- Action: Представляє метод, що не повертає значення, але може приймати параметри.
- Func: Представляє метод, що повертає значення.
- Predicate: Використовується для виконання перевірки на певну умову та повертає булеве значення.
- EventHandler: Використовується для реалізації подій.
- Comparison: Використовується для порівняння двох об'єктів.

Делегати і стандартні делегати в .NET - це потужний інструмент, що полегшує створення гнучких, зручних та розширюваних програм. Їх використання дозволяє покращити структуру коду, зробити його більш читабельним та легким у розширенні.

1.2 Аналіз технічного завдання на роботу

Мета курсового проєкту – створення користувацького додатку який відповідає темі курсового проєкту, відпрацювання навичок що були здобуті протягом навчального курсу.

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

					122 – КР.2023.01.000 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.1 Функціональні вимоги

1. Можливість реєстрації та авторизації.
2. Можливість перегляду списку злочинців
3. Можливість перегляду списку кримінальних угруповань
4. Можливість перегляду на головному вікні невеликої статистики такої як: кількість злочинців у базі, кількість кримінальних угруповань що знаходяться у базі та кількість агентів.
5. Можливість фільтрації по імені злочинців та фільтрації по наві угруповання.
6. Можливість серіалізації даних у JSON формат злочинця або кримінального угруповання.

1.2.2 Нефункціональні вимоги

1. Зовнішній інтерфейс користувача має бути реалізованим за допомогою створення вікон в середовищі WPF додатку.
2. Додаток має бути багатовіконним
3. Усі поля введення повинні бути захищені від некоректного введення.
4. Використання патерну MVVM та прив'язки даних (Binding).

1.3 Опис алгоритму основних задач/підзадач у роботі

Алгоритм авторизації користувача достатньо простий адже він не містить ніяких хешувань наприклад MD5 та інших алгоритмів що могли би дати більшої безпечності обліковому запису користувача, розглянемо алгоритм авторизації користувача на блок-схемі рис 1.2

					122 – КР.2023.01.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

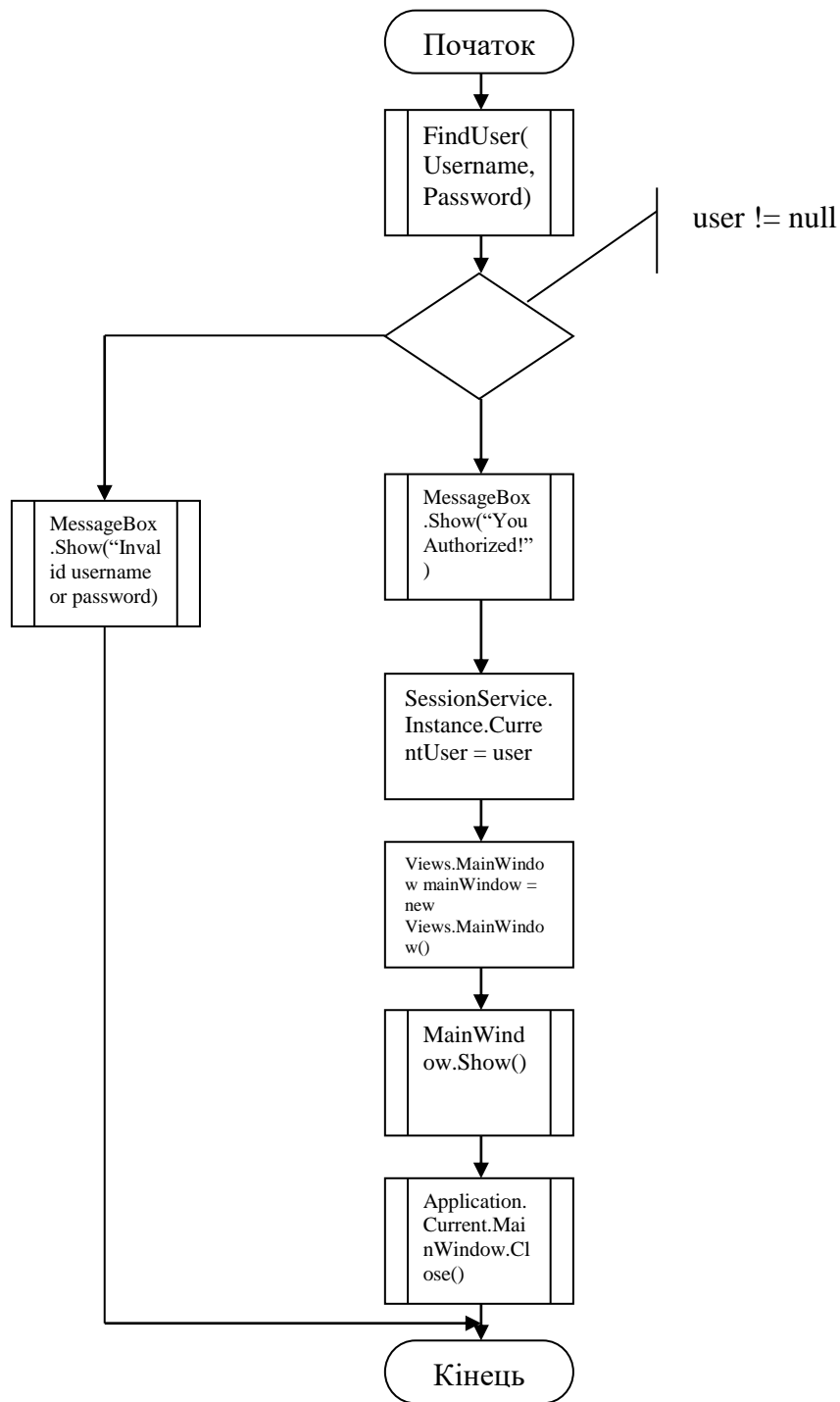


Рисунок 1.2 – Блок-схема алгоритму авторизації користувача

1.3.1 Структура програмного забезпечення

Висновки до розділу

У першому розділі ми докладно розглянули основні принципи об'єктно-орієнтованого програмування (ООП). Почали з формулювання базових принципів, які лежать в основі цього підходу до створення програм. Далі ми розробили розширену постановку задачі, яка включала в себе більш детальний опис завдання та його контексту.

					122 – КР.2023.01.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Структура програмного забезпечення

Структура програмного забезпечення - це комплекс модулів, які утворюють функціональну систему. Кожен з цих модулів призначений для виконання певних завдань програми, розв'язання основних завдань проекту та реалізації необхідних функцій.

Застосунок створений за допомогою технології Windows Presentation Foundation (WPF) на базі об'єкто-орієнтованої мови програмування C#.

На рис. 2.1 зображено оглядач рішень програмного застосунку.

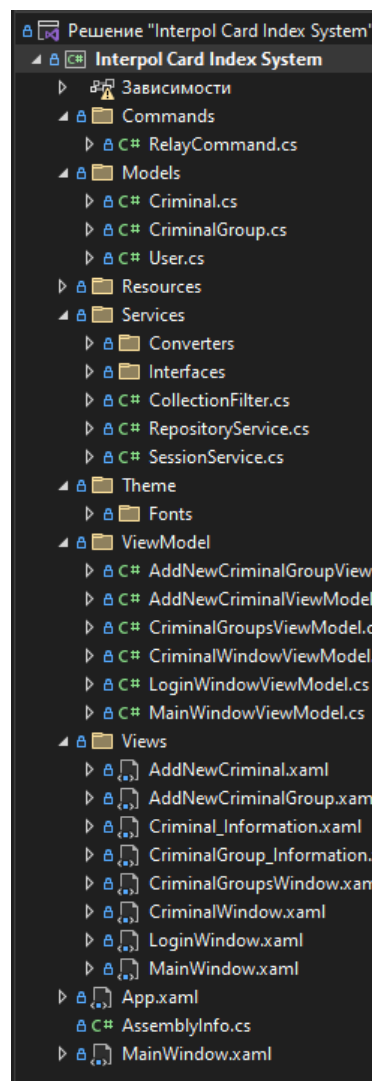


Рисунок 2.1 – Оглядач рішень

В табл.2.1 представлено опис модулів (класів) проекту курсової роботи.

					122 – КР.2023.01.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Призначення модулів програми

Назва модуля	Призначення модуля
AddNewCriminal.xaml	Клас вікна для додавання нового злочинця.
AddNewCriminalGroup.xaml	Вікно для додавання кримінального угруповання.
CriminalGroup_Information.xaml	Вікно для перегляду повної інформації про кримінальне угруповання.
CriminalGroupsWindow.xaml	Вікно для перегляду списку усіх злочинців.
Criminal_Information.xaml	Вікно для перегляду повної інформації про обраного злочинця.
LoginWindow.xaml	Вікно призначене для авторизації користувача, є стартовим вікном при ініціалізації додатку.
MainWindow.xaml	Головне вікно додатку
RepositoryService.cs	Клас, що зберігає дані усіх злочинців, кримінальних угруповань а також агентів.
SessionService.cs	Зберігає інформацію про поточну сесію користувача.
CollectionFilter.cs	Клас-шаблон що забезпечує можливість фільтрації колекції через передачу предикатів.
ICollectionFilter.cs	Інтерфейс що зберігає в собі обов'язкові до імплементації методи для класу що відповідає за фільтрацію колекцій.
RelayCommand.cs	Клас, що дозволяє створювати команди які будуть використовуватись у прив'язці.

Продовження таблиці 2.1

Назва модуля	Призначення модуля
LoginWindowViewModel.cs	Клас, що пов'язує вікно реєстрації з моделлю користувача та сесією.
MainWindowViewModel.cs	Клас, що пов'язує головне вікно програми з деякою інформацією з репозиторієм та містить декілька команд для переходу на інші вікна.
CriminalWindowViewModel.cs	Клас, що пов'язує вікно для перегляду списку злочинців з моделлю злочинців та списком що містить ці моделі.
CriminalGroupsViewModel.cs	Клас, що пов'язує вікно для перегляду списку кримінальних угруповань з моделлю кримінальних угруповань та списком що містить ці моделі.
AddNewCriminalViewModel.cs	Клас, що пов'язує діалогове вікно додавання нового злочинця до бази, містить модель нового злочинця яка потім відправляється у репозиторій через команду.
AddNewCriminalGroupViewModel.cs	Клас, що пов'язує діалогове вікно додавання нової кримінальної групи до бази, містить модель нової кримінальної групи та команду яка відправляє її у репозиторій. Також містить в собі список країн які читаються з файлу.

Продовження таблиці 2.1

Назва модуля	Призначення модуля
MainWindow.xaml.cs	Клас, що представляє головне вікно програми.
LoginWindow.xaml.cs	Клас, що представляє вікно авторизації користувача.
Criminal_Information.xaml.cs	Клас, що представляє вікно перегляду інформації про певного злочинця.
CriminalWindow.xaml.cs	Клас, що представляє вікно перегляду списку злочинців.
CriminalGroupsWindow.xaml.cs	Клас, що представляє вікно перегляду списку кримінальних угруповань.
CriminalGroup_Information.xaml.cs	Клас, що представляє вікно перегляду інформації про певне кримінальне угруповання.
AddNewCriminalGroup.xaml.cs	Клас, що представляє вікно додавання нового кримінального угруповання.
AddNewCriminal.xaml.cs	Клас, що представляє вікно додавання нового злочинця.
Criminal.cs	Представляє модель злочинця.
CriminalGroup.cs	Представляє модель кримінального угруповання.
User.cs	Представляє модель користувача.

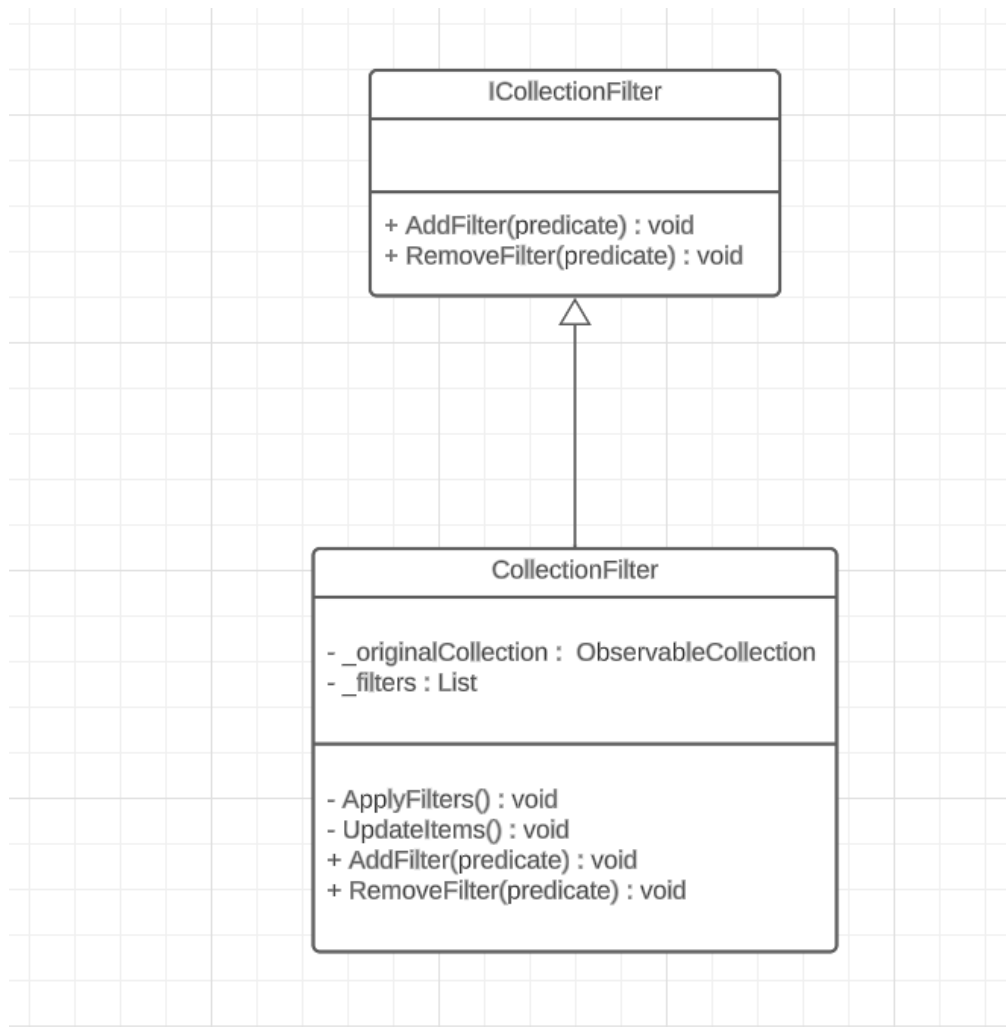


Рисунок 2.2 – UML діаграма класу CollectionFilter який наслідує інтерфейс

Під час створення програми було створено 8 вікон. Діаграма класів вікон зображена на рис. 2.2. Всі зображені класи неявно реалізують інтерфейс IComponentConnector та наслідують клас Window.

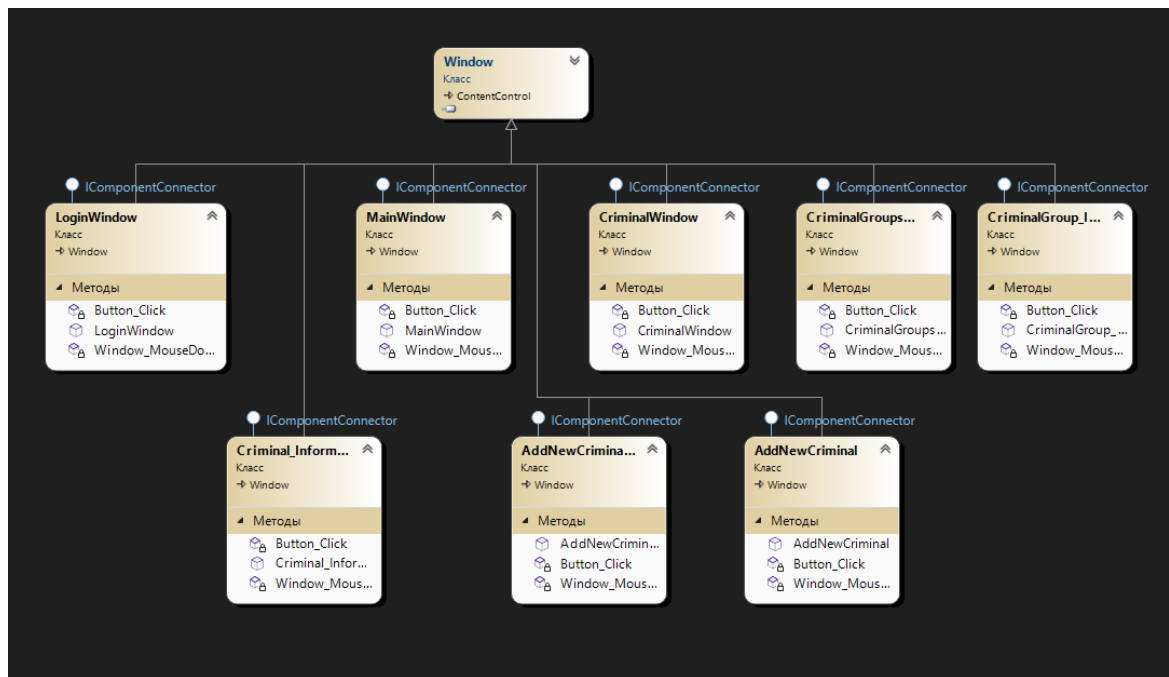


Рисунок 2.3 – Діаграма класів вікон

Для представлення злочинця, кримінального угруповання та агента, використовуються три класи які є моделями, вони зображені на рис. 2.4

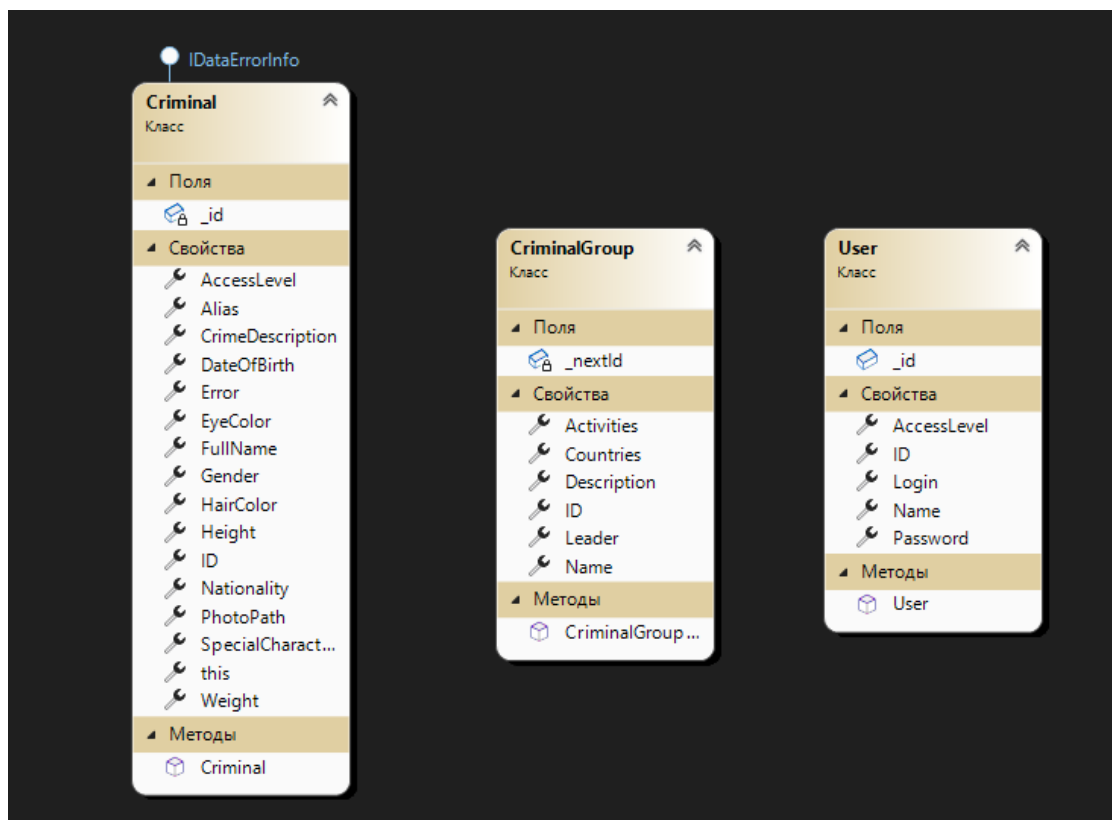


Рисунок 2.4 – Моделі даних

2.2 Опис роботи програми

Після запуску додатка відкривається вікно авторизації де при правильному введенні логіну та паролю користувача для нього з'явиться повідомлення що він увійшов до системи, після чого буде перехід на головне вікно програми. На головному вікні програми є невелика статистика по кількості злочинців та кримінальних угруповань в головній секції, також інформація про агента. На головному вікні також є бокове меню зліва де містяться такі пункти:

- «Criminals»: після натиску на цю кнопку буде перенаправлення на вікно де буде список усіх злочинців які є у базі;
- «Crime Groups»: після натиску на цю кнопку буде перенаправлення на вікно де буде список усіх кримінальних угруповань;
- «Agents»: після натиску на цей пункт буде відкрите нове вікно де буде список усіх агентів у системі.

Після того як користувач переходить на вікна де є списки злочинців або кримінальних угруповань у вигляді таблиці він може передивитись повну інформацію про них натиснувши на кнопку «Details», також на цих вікнах є кнопка «Add» яка відповідає за функцію додання нових даних про кримінальне угруповання чи злочинців.

2.3 Функціональна схема програми

Детальний опис й пояснення процесів окремих функцій розробляємої системи утворюють функціональну схему програми (рис. 2.5).

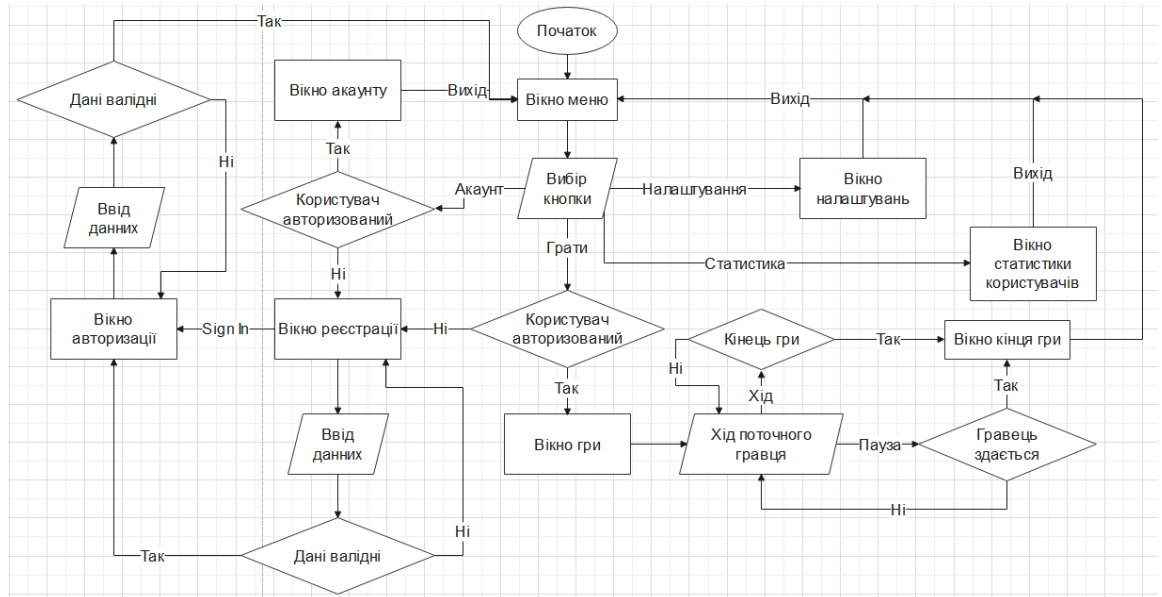


Рисунок 2.5 – Функціональна схема програмного коду

2.4 Опис інтерфейсу програми

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

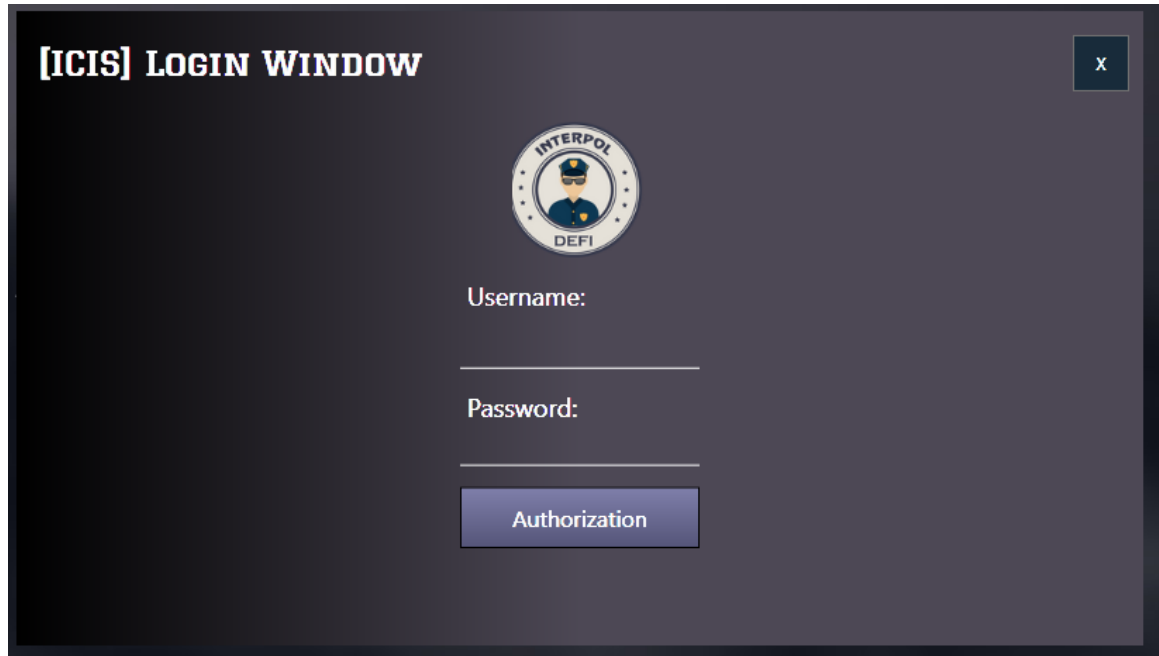


Рисунок 2.6 – Вікно авторизації користувача у систему

Вікно авторизації включає у себе 2 компонента типу Button, один компонент Image, два компонента TextBox, два компонента TextBlock та один Label.

При вірному вводі логіну та паролю користувачу буде висвічений MessageBox, як показано на рис 2.7

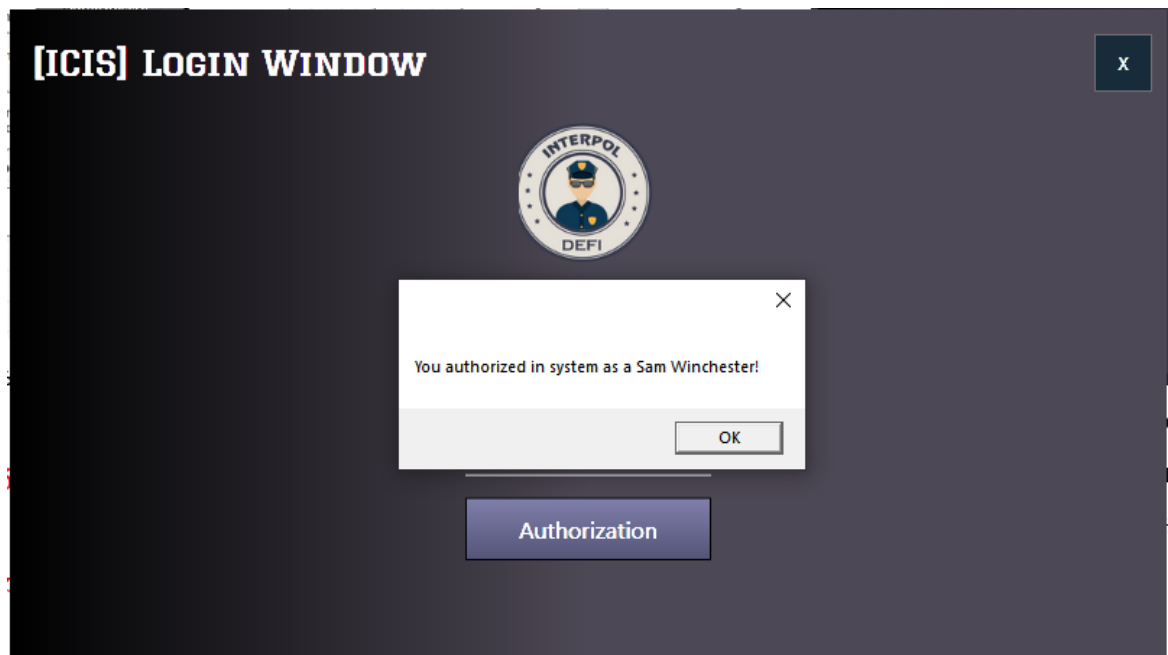


Рисунок 2.7 – MessageBox при успішній авторизації у систему

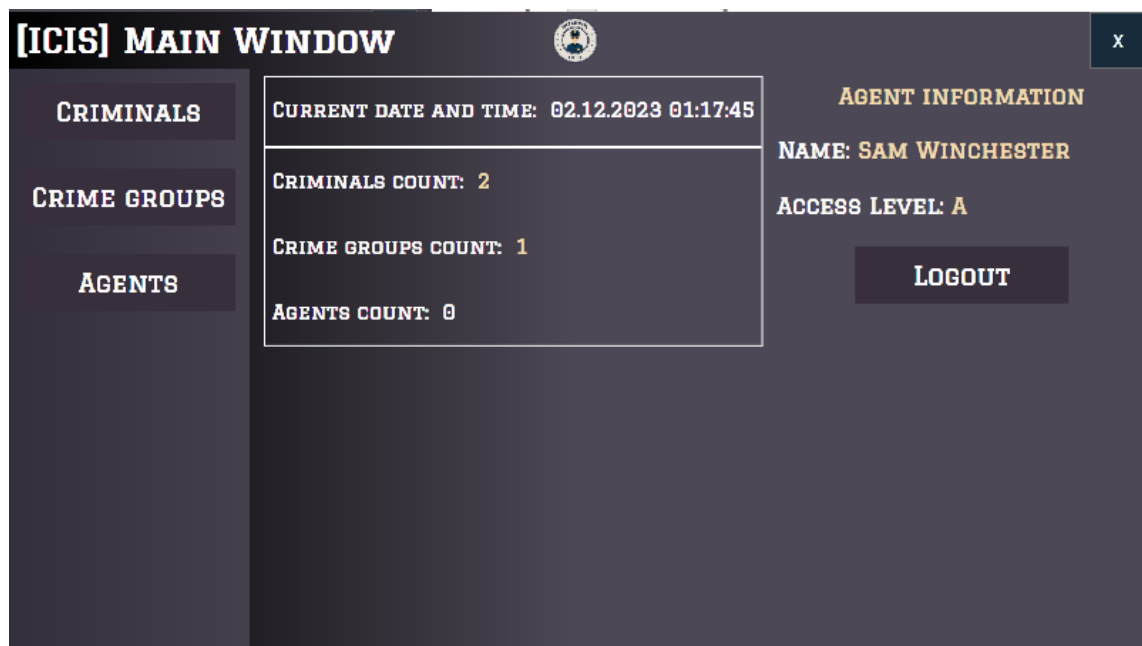


Рисунок 2.8 – Головне вікно програми

На головному вікні програми розташовано п'ять елементів керування типу Button, один елемент Image, один Label та сім елементів TextBlock.

Кнопки на цій формі відповідають за:

- Кнопка «Criminals» робить перехід на нове вікно де буде відображений список злочинців які є у базі Інтерполу;

- Кнопка «Crime Groups» робить перехід на нове вікно де відображений список кримінальних угруповань які є у базі Інтерполу;

- Кнопка «Agents» робить перехід на нове вікно де буде відображений список усіх агентів які є у системі.
- Кнопка «Logout» виконує завершення сесії користувача і виходить із системи;
- Кнопка яка розташована у правому верхньому кутку вікна відповідає за зачинення цього вікна.

Рисунок 2.9 – Вікно додавання нового злочинця.

На вікні яке дає змогу додавати злочинця розташовано 11 елементів типу TextBlock, 8 елементів TextBox, 3 елементи типу Button (одна із кнопок виступає як завантажувач фото злочинця) та один Border.

[ICIS] ADD CRIMINAL GROUP

GROUP NAME:

LEADER:

DESCRIPTION:

ACTIVITIES:

Theft
Drug-related crimes
Financial crimes

COUNTRIES:

United States
Canada
United Kingdom

CONFIRM
CANCEL

Рисунок 2.10 – Вікно додавання нового кримінального угруповання

На вікні яке зображено на рис. 2.10 розташовано шість елементів типу TextBlock, 3 елементу типу TextBox (один з яких є багаторядковим), 2 елементу типу Button та 2 елемента типу ListBox.

[ICIS] CRIMINALS
x

ADD

Name	AKA	Date of Birth	Nationality	Action
Jack Sparrow	Sparrow	October 1982	Ukrainian	Details
Ryan Gosling	Sierra-6	July 1994	USA	Details

Рисунок 2.11 – Вікно перегляду списку злочинців.

На даному вікні розташовано один елемент керування типу DataGrid, два елемента типу Button, один TextBlock та один елемент типу TextBox.

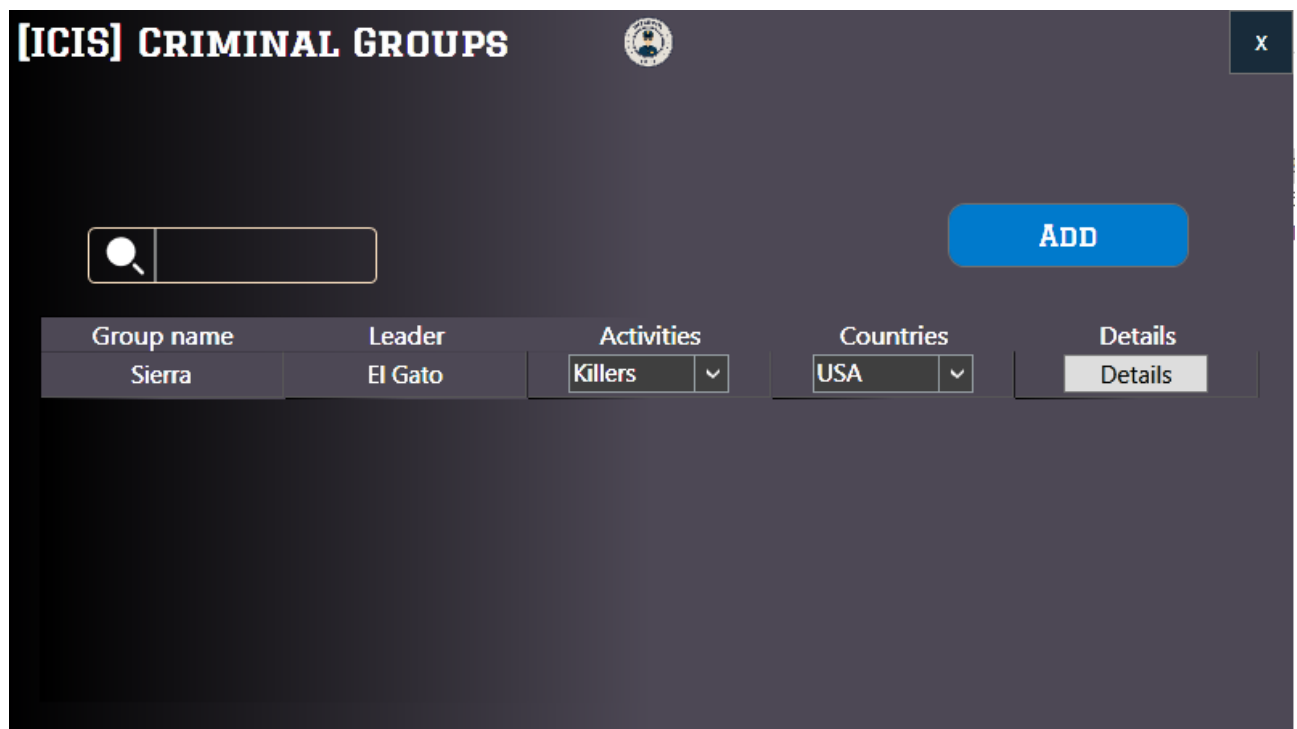


Рисунок 2.12 – Вікно перегляду списку кримінальних угруповань.

На даному вікні розташовано один елемент керування типу DataGrid, два елементи типу Button, один елемент типу TextBlock, один елемент Image та один TextBox.

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

Курсовий проєкт був спрямований на закріплення набутих навичок які були отримані протягом навчального семестру та створення WPF додатку «Картотека Інтерполу». Розглянуті базові принципи ООП, проаналізовано предметну область та розглянуто технічне завдання, реалізовано можливість авторизації, можливість додати злочинців та кримінальні угруповання до картотеки, переглянути невеличку статистику картотеки.

Для виконання даного проєкту була використана мова програмування C#, яка добре підходить для розробки програм з використанням об'єктно-орієнтованого підходу до розробки програм та технологія WPF яка дозволяє гнучко розробляти інтерфейс. Також важливим архітектурним рішенням було використання паттерну MVVM (Model-View-ViewModel).

Важливим етапом розробки стали планування й реалізація користувацького інтерфейсу, було використано функціонал валідації для перевірки правильності введених даних користувачем у інтерфейс програми.

Таким чином, у курсового проєкту продемонстровані найважливіші методи роботи з програмним середовищем WPF, його інструментами та функціями.

					122 – КР.2023.01.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ЛІТЕРАТУРИ

1. WPF и C# | Полное руководство. Інформаційний ресурс.
<https://metanit.com/sharp/wpf/>
2. Адам Натан. WPF 4. Подробное руководство. Символ-Плюс, 2011. 848 с.
3. Крис Андерсон. Основы Windows Presentation Foundation. Санкт-Петербург: Litres, 2022. 413 с.
4. Джозеф Албахари. C# 7.0 Справочник. Диалектика, 2017. 222 с.
5. Бхаргава. Грокаем алгоритмы. Питер. 2022. 288 с.

					122 – КР.2023.01.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А
ХАМЛ розмітки вікон

					122 – КР.2023.01.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Файл App.xaml

```
<Application x:Class="Interpol_Card_Index_System.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:Interpol_Card_Index_System"
    StartupUri="Views/LoginWindow.xaml">
    <Application.Resources>
        <FontFamily x:Key="GraduateFont">pack://application:,,,/Theme/Fonts/Graduate-
Regular.ttf#Graduate</FontFamily>
        <FontFamily
x:Key="DramaturgSC">pack://application:,,,/Theme/Fonts/DramaturgSC.ttf#Dramaturg</FontF
amily>
        <BitmapImage x:Key="interpol_logo"
UriSource="pack://application:,,,/Resources/Images/interpol_logo.png"/>

        <LinearGradientBrush x:Key="ApplicationBackgroundGradient" StartPoint="0, 1"
EndPoint="1, 1">
            <GradientStop Color="#000000" Offset="0"/>
            <GradientStop Color="#4D4855" Offset="0.5"/>
        </LinearGradientBrush>

        <Style x:Key="MenuButtonStyle" TargetType="Button">
            <Setter Property="Width" Value="150"/>
            <Setter Property="Height" Value="40"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="FontFamily" Value="{DynamicResource GraduateFont}"/>
            <Setter Property="FontSize" Value="18"/>
            <Setter Property="Margin" Value="0, 10"/>
            <Setter Property="Background" Value="#372F3C"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Grid>
                            <Rectangle x:Name="PART_BottomLine" Fill="{TemplateBinding
Background}" />
                            <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
                        </Grid>
                        <ControlTemplate.Triggers>
                            <Trigger Property="IsMouseOver" Value="True">
                                <Setter TargetName="PART_BottomLine" Property="Fill"
Value="#463F4E"/>
                            </Trigger>
                        </ControlTemplate.Triggers>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>
    </Application.Resources>
</Application>
```

Файл LoginWindow.xaml

```
<Window x:Class="Interpol_Card_Index_System.Views.LoginWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:Interpol_Card_Index_System.Views" xmlns:local1="clr-
namespace:Interpol_Card_Index_System.ViewModel">
```

					122 – КР.2023.01.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

```

mc:Ignorable="d"
Height="450" Width="800" ResizeMode="NoResize" WindowStyle="None"
WindowStartupLocation="CenterScreen"
MouseDown="Window_MouseDown">
<Window.Resources>
<Style x:Key="MyButtonStyle" TargetType="Button">
<Setter Property="FontSize" Value="16"/>
<Setter Property="Foreground" Value="White"/>
<Setter Property="Padding" Value="10"/>
<Setter Property="BorderBrush" Value="Black"/>
<Setter Property="BorderThickness" Value="1"/>
<Setter Property="Background">
<Setter.Value>
<LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
<GradientStop Offset="0" Color="#FF7F7FAA"/>
<!-- Более светлый сине-фиолетовый -->
<GradientStop Offset="1" Color="#FF555579"/>
<!-- Более темный сине-фиолетовый -->
</LinearGradientBrush>
</Setter.Value>
</Setter>
<Setter Property="Template">
<Setter.Value>
<ControlTemplate TargetType="Button">
<Border Background="{TemplateBinding Background}"
BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding
BorderThickness}" Padding="{TemplateBinding Padding}">
<ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
</Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>

<Style x:Key="BottomLineTextBox" TargetType="TextBox">
<Setter Property="Background" Value="Transparent"/>
<Setter Property="BorderThickness" Value="0"/>
<Setter Property="BorderBrush" Value="Transparent"/>
<Setter Property="Padding" Value="0"/>
<Setter Property="Foreground" Value="White"/>
<Setter Property="FontSize" Value="16"/>
<Setter Property="Height" Value="30"/>
<Setter Property="Template">
<Setter.Value>
<ControlTemplate TargetType="TextBox">
<Grid>
<Rectangle x:Name="PART_BottomLine"
Fill="White"
Height="1"
VerticalAlignment="Bottom"
Width="{Binding ActualWidth,
ElementName=PART_ContentHost}"/>
<ScrollViewer x:Name="PART_ContentHost"
SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}"/>
</Grid>
<ControlTemplate.Triggers>
<Trigger Property="IsFocused" Value="True">
<Trigger.EnterActions>
<BeginStoryboard>
<Storyboard>

```

```

                                <ColorAnimation
Storyboard.TargetName="PART_BottomLine"
Storyboard.TargetProperty="(Rectangle.Fill).(SolidColorBrush.Color)"
                                To="Green"
                                Duration="0:0:0.5" />
                                </Storyboard>
                                </BeginStoryboard>
                                </Trigger.EnterActions>
                                <Trigger.ExitActions>
                                <BeginStoryboard>
                                <Storyboard>
                                <ColorAnimation
Storyboard.TargetName="PART_BottomLine"
Storyboard.TargetProperty="(Rectangle.Fill).(SolidColorBrush.Color)"
                                To="White"
                                Duration="0:0:0.5" />
                                </Storyboard>
                                </BeginStoryboard>
                                </Trigger.ExitActions>
                                </Trigger>
                                </ControlTemplate.Triggers>
                                </ControlTemplate>
                                </Setter.Value>
                                </Setter>
                                </Style>
                                </Window.Resources>

                                <Window.Background>
                                <StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
                                </Window.Background>

                                <Window.DataContext>
                                <local1:LoginWindowViewModel/>
                                </Window.DataContext>

                                <Grid Margin="10">
                                <Grid.RowDefinitions>
                                <RowDefinition Height="*"/>
                                <RowDefinition Height="7*"/>
                                </Grid.RowDefinitions>

                                <Label Content="[ICIS] Login Window" Grid.Row="0" Foreground="White"
                                FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
                                FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>
                                <Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
                                Height="40" Width="40" Margin="0"
                                HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

                                <StackPanel Grid.Row="1" HorizontalAlignment="Center" Width="170" Margin="0,
5">
                                <Image VerticalAlignment="Center" HorizontalAlignment="Center"
                                Source="pack://application:,,,/Resources/Images/interpol_logo.png"
                                Width="96" Height="96" Margin="0, 10"/>

                                <StackPanel>
                                <Label Content="Username:" Foreground="White" FontSize="18"/>

```

```

        <TextBox x:Name="loginTextBox" Style="{StaticResource
BottomLineTextBox}"
                Text="{Binding Username, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}"
                Width="170" Height="25" Margin="0, 10" FontSize="14"/>
    </StackPanel>
    <StackPanel>
        <Label Content="Password:" Foreground="White" FontSize="18"/>
        <TextBox x:Name="passwordTextBox" Style="{StaticResource
BottomLineTextBox}"
                Text="{Binding Password, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}"
                Width="170" Height="25" FontSize="14"/>
    </StackPanel>
    <Button Content="Authorization" Margin="0, 15"
        Style="{StaticResource MyButtonStyle}" Command="{Binding
LoginCommand}">
        <Button.Triggers>
            <EventTrigger RoutedEvent="Button.MouseEnter">
                <BeginStoryboard>
                    <Storyboard>
                        <ColorAnimation Duration="0:0:0.2"
Storyboard.TargetProperty="(Button.Background).(GradientBrush.GradientStops)[0].(GradientStop.Color)"
                                To="#FF4C4C6F" />
                        <ColorAnimation Duration="0:0:0.2"
Storyboard.TargetProperty="(Button.Background).(GradientBrush.GradientStops)[1].(GradientStop.Color)"
                                To="#FF2D2D49" />
                    </Storyboard>
                </BeginStoryboard>
            </EventTrigger>
            <EventTrigger RoutedEvent="Button.MouseLeave">
                <BeginStoryboard>
                    <Storyboard>
                        <ColorAnimation Duration="0:0:0.2"
Storyboard.TargetProperty="(Button.Background).(GradientBrush.GradientStops)[0].(GradientStop.Color)"
                                To="#FF7F7FAA" />
                        <ColorAnimation Duration="0:0:0.2"
Storyboard.TargetProperty="(Button.Background).(GradientBrush.GradientStops)[1].(GradientStop.Color)"
                                To="#FF555579" />
                    </Storyboard>
                </BeginStoryboard>
            </EventTrigger>
        </Button.Triggers>
    </Button>
</StackPanel>
</Grid>
</Window>

```

Файл MainWindow.xaml

```
<Window x:Class="Interpol_Card_Index_System.Views.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

					122 – КР.2023.01.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:Interpol_Card_Index_System.Views" xmlns:local1="clr-
namespace:Interpol_Card_Index_System.ViewModel"
mc:Ignorable="d"
Height="450" Width="800" ResizeMode="NoResize" WindowStyle="None"
WindowStartupLocation="CenterScreen"
MouseDown="Window_MouseDown">
<Window.Background>
<StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
</Window.Background>

<Window.Resources>
<Style x:Key="MenuButtonStyle" TargetType="Button">
<Setter Property="Width" Value="150"/>
<Setter Property="Height" Value="40"/>
<Setter Property="Foreground" Value="White"/>
<Setter Property="FontFamily" Value="{StaticResource GraduateFont}"/>
<Setter Property="FontSize" Value="18"/>
<Setter Property="Margin" Value="0, 10"/>
<Setter Property="Background" Value="#372F3C"/>
<Setter Property="Template">
<Setter.Value>
<ControlTemplate TargetType="Button">
<Grid>
<Rectangle x:Name="PART_BottomLine" Fill="{TemplateBinding
Background}" />
<ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
</Grid>
<ControlTemplate.Triggers>
<Trigger Property="IsMouseOver" Value="True">
<Setter TargetName="PART_BottomLine" Property="Fill"
Value="#463F4E"/>
</Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
</Window.Resources>

<Window.DataContext>
<local1:MainWindowViewModel/>
</Window.DataContext>

<Grid>
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition/>
</Grid.RowDefinitions>

<Label Content="[ICIS] Main Window" Grid.Row="0" Foreground="White"
FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>
<Image Source="pack://application:,,,/Resources/Images/interpol_logo.png"
Width="32" Height="32"
HorizontalAlignment="Center"/>
<Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
Height="40" Width="40" Margin="0"

```

					122 – КР.2023.01.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

<Grid Grid.Row="1">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Column="0" Width="170">
        <StackPanel.Background>
            <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
                <GradientStop Offset="0" Color="#332F3B" />
                <GradientStop Offset="1" Color="#463F4E" />
            </LinearGradientBrush>
        </StackPanel.Background>

        <Button Content="Criminals" Style="{StaticResource MenuButtonStyle}"
Command="{Binding ShowCriminalWindow}"/>
        <Button Content="Crime groups" Style="{StaticResource MenuButtonStyle}"
Command="{Binding ShowCriminalGroupWindow}"/>
        <Button Content="Agents" Style="{StaticResource MenuButtonStyle}"/>
    </StackPanel>

    <StackPanel Grid.Column="1">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="0.7*"/>
            </Grid.ColumnDefinitions>

            <Border Margin="10, 5" BorderBrush="White" BorderThickness="1">
                <StackPanel>
                    <TextBlock Margin="5, 15" Foreground="White"
FontFamily="{StaticResource GraduateFont}" FontSize="14">
                        <Run Text="Current date and time: " />
                        <Run Text="{Binding CurrentDateAndTime}" />
                    </TextBlock>
                    <Border BorderBrush="White" BorderThickness="0, 0, 0, 2"
Height="4"/>
                    <TextBlock Margin="5, 15" Foreground="White"
FontFamily="{StaticResource GraduateFont}" FontSize="14">
                        <Run Text="Criminals count: " />
                        <Run Text="{Binding CriminalsCount, Mode=OneWay}"
Foreground="Wheat"/>
                    </TextBlock>
                    <TextBlock Margin="5, 15" Foreground="White"
FontFamily="{StaticResource GraduateFont}" FontSize="14">
                        <Run Text="Crime groups count: " />
                        <Run Text="{Binding CriminalGroupsCount, Mode=OneWay}"
Foreground="Wheat"/>
                    </TextBlock>
                    <TextBlock Margin="5, 15" Foreground="White"
FontFamily="{StaticResource GraduateFont}" FontSize="14">
                        <Run Text="Agents count: " />
                        <Run Text="0" />
                    </TextBlock>
                </StackPanel>
            </Border>
        </Grid>
    </StackPanel>
</Grid>

```

```

        <StackPanel Grid.Column="1">
            <TextBlock Text="Agent information" FontFamily="{StaticResource
GraduateFont}"
                Foreground="Wheat" FontSize="16"
HorizontalAlignment="Center" Margin="0, 10"/>
            <TextBlock FontFamily="{StaticResource GraduateFont}"
Margin="0, 10"
                Foreground="White" FontSize="16">
                <Run Text="Name:"/>
                <Run Text="{Binding CurrentUserName, Mode=OneWay}"
Foreground="Wheat"/>
            </TextBlock>
            <TextBlock FontFamily="{StaticResource GraduateFont}"
                Foreground="White" FontSize="16" Margin="0, 10">
                <Run Text="Access Level:"/>
                <Run Text="{Binding CurrentUserAccess, Mode=OneWay}"
Foreground="Wheat"/>
            </TextBlock>
            <Button Content="Logout" Style="{StaticResource
MenuButtonStyle}" Command="{Binding LogoutCommand}"/>
        </StackPanel>
    </Grid>
</Grid>
</Grid>
</Window>

```

Файл CriminalWindow.xaml

```

<Window x:Class="Interpol_Card_Index_System.Views.CriminalWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:Interpol_Card_Index_System.Views" xmlns:local1="clr-
namespace:Interpol_Card_Index_System.ViewModel"
    mc:Ignorable="d"
    Height="450" Width="800" ResizeMode="NoResize" WindowStyle="None"
WindowStartupLocation="CenterScreen"
    MouseDown="Window_MouseDown">
    <Window.Background>
        <StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
    </Window.Background>

    <Window.Resources>
        <Style x:Key="MenuButtonStyle" TargetType="Button">
            <Setter Property="Width" Value="150"/>
            <Setter Property="Height" Value="40"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="FontFamily" Value="{StaticResource GraduateFont}"/>
            <Setter Property="FontSize" Value="18"/>
            <Setter Property="Margin" Value="0, 10"/>
            <Setter Property="Background" Value="#372F3C"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Grid>
                            <Rectangle x:Name="PART_BottomLine" Fill="{TemplateBinding
Background}" />
                            <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
                        </Grid>
                    </Setter.Value>
                </ControlTemplate>
            </Setter>
        </Style>
    </Window.Resources>

```

					122 – КР.2023.01.000 ПЗ	Арх.
						30
Змн.	Арх.	№ докум.	Підпис	Дата		

```

        </Grid>
        <ControlTemplate.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter TargetName="PART_BottomLine" Property="Fill"
Value="#463F4E"/>
            </Trigger>
        </ControlTemplate.Triggers>
    </ControlTemplate>
    </Setter.Value>
</Setter>
</Style>

<Style x:Key="DataGridColumnHeaderStyle" TargetType="DataGridColumnHeader">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="HorizontalContentAlignment" Value="Center"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="DataGridColumnHeader">
                <Grid Background="#4D4855">
                    <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

<Style x:Key="DataGridRowHeaderStyle" TargetType="DataGridRowHeader">
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="HorizontalContentAlignment" Value="Center"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="DataGridRowHeader">
                <Grid Background="#4D4855">
                    <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

<Style x:Key="DataGridCellStyle" TargetType="DataGridCell">
    <Setter Property="Background" Value="#4D4855"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="BorderBrush" Value="#585858"/>
    <Setter Property="BorderThickness" Value="0,0,1,1"/>
    <Setter Property="FontSize" Value="15"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type DataGridCell}">
                <Border Background="{TemplateBinding Background}"
BorderBrush="{TemplateBinding BorderBrush}"
BorderThickness="{TemplateBinding BorderThickness}"
Padding="{TemplateBinding Padding}"
SnapsToDevicePixels="True">
                    <ContentPresenter SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
HorizontalAlignment="Center"
VerticalAlignment="Center" />
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

```

```

        </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

<!-- Style for selected cell -->
<Style x:Key="DataGridCellSelectedStyle" TargetType="DataGridCell">
    <Setter Property="Background" Value="#4D4855"/>
    <Setter Property="BorderBrush" Value="#585858"/>
    <Setter Property="BorderThickness" Value="0,0,1,1"/>
    <Setter Property="Padding" Value="10"/>
</Style>

<!-- Background cell animation -->
<Style TargetType="DataGridCell" x:Key="DataGridCellHoverStyle">
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#372F3C"/>
        </Trigger>
    </Style.Triggers>
</Style>

<Style TargetType="DataGrid" x:Key="CustomDataGridStyle">
    <Setter Property="Background" Value="{StaticResource
ApplicationBackgroundGradient}" />
    <Setter Property="BorderBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="RowBackground" Value="{StaticResource
ApplicationBackgroundGradient}" />
    <Setter Property="AlternatingRowBackground" Value="{StaticResource
ApplicationBackgroundGradient}" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="HorizontalGridLinesBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
    <Setter Property="VerticalGridLinesBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
</Style>

<Style x:Key="AddButtonStyle" TargetType="Button">
    <Setter Property="Width" Value="150"/>
    <Setter Property="Height" Value="40"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="FontFamily" Value="{StaticResource GraduateFont}"/>
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="Margin" Value="0, 10"/>
    <Setter Property="Background" Value="#007ACC"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Border x:Name="border" Background="{TemplateBinding
Background}" BorderBrush="{TemplateBinding BorderBrush}"
                BorderThickness="{TemplateBinding BorderThickness}"
                CornerRadius="10">
                    <ContentPresenter HorizontalAlignment="Center"
                    VerticalAlignment="Center"/>
                </Border>
            <ControlTemplate.Triggers>
                <Trigger Property="IsMouseOver" Value="True">
                    <Trigger.EnterActions>
                        <BeginStoryboard>

```

```

        <Storyboard>
            <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                To="#00546B" Duration="0:0:1"/>
        </Storyboard>
    </BeginStoryboard>
</Trigger.EnterActions>
<Trigger.ExitActions>
    <BeginStoryboard>
        <Storyboard>
            <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                To="#007ACC" Duration="0:0:1"/>
        </Storyboard>
    </BeginStoryboard>
</Trigger.ExitActions>
</Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>

<Style TargetType="TextBox" x:Key="SearchTextBox">
    <Setter Property="Padding" Value="15,2,2,2"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="BorderBrush" Value="Bisque"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type TextBox}">
                <Border BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}"
                    CornerRadius="4">
                    <Grid>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="Auto"/>
                            <ColumnDefinition Width="*" />
                        </Grid.ColumnDefinitions>

                        <Viewbox Width="24" Height="24" Margin="2, 0, 0, 0">
                            <Canvas Width="20" Height="20">
                                <Path Fill="White" Data="M8 15.5a7.5 7.5 0 1 0
0-15 7.5 7.5 0 0 0 15Zm11.707 2.793-4-4a1 1 0 0 0-1.414 1.414l4 4a1 1 0 0 0 1.414-
1.414Z"/>
                            </Canvas>
                        </Viewbox>

                        <Border Grid.Column="0" Width="2" Background="Gray"
Margin="40,0,0,0"/>

                        <ScrollViewer x:Name="PART_ContentHost" Grid.Column="1"
Margin="0,0,2,0" VerticalAlignment="Center" Focusable="false"/>
                    </Grid>
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

```

					122 – КР.2023.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        </Border>
        <ControlTemplate.Triggers>
            <Trigger Property="IsEnabled" Value="false">
                <Setter Property="Opacity" Value="0.5"/>
            </Trigger>
        </ControlTemplate.Triggers>
    </ControlTemplate>
</Setter.Value>
</Setter>
</Style>
</Window.Resources>

<Window.DataContext>
    <local1:CriminalWindowViewModel/>
</Window.DataContext>

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition/>
    </Grid.RowDefinitions>

    <Label Content="[ICIS] Criminals" Grid.Row="0" Foreground="White"
        FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
        FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>
    <Image Source="pack://application:,,,/Resources/Images/interpol_logo.png"
Width="32" Height="32"
        HorizontalAlignment="Center"/>
    <Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
        Height="40" Width="40" Margin="0"
        HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

    <Grid Grid.Row="1">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>

        <StackPanel Grid.Column="0" Width="170">
            <StackPanel.Background>
                <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
                    <GradientStop Offset="0" Color="#332F3B" />
                    <GradientStop Offset="1" Color="#463F4E" />
                </LinearGradientBrush>
            </StackPanel.Background>

            <Button Content="Main" Style="{StaticResource MenuButtonStyle}"/>
            <Button Content="Crime Groups" Style="{StaticResource
MenuButtonStyle}"/>
            <Button Content="Agents" Style="{StaticResource MenuButtonStyle}"/>
        </StackPanel>

        <!--Main content section here -->

        <Grid Grid.Row="1" Grid.Column="1">
            <Grid.RowDefinitions>
                <RowDefinition Height="*"/>
                <RowDefinition Height="3*"/>
            </Grid.RowDefinitions>

```



```

        xmlns:local="clr-namespace:Interpol_Card_Index_System.Views" xmlns:local1="clr-
namespace:Interpol_Card_Index_System.ViewModel"
        mc:Ignorable="d"
        Height="450" Width="800" ResizeMode="NoResize" WindowStyle="None"
WindowStartupLocation="CenterScreen"
        MouseDown="Window_MouseDown">
    <Window.Background>
        <StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
    </Window.Background>

    <Window.Resources>
        <Style x:Key="MenuButtonStyle" TargetType="Button">
            <Setter Property="Width" Value="150"/>
            <Setter Property="Height" Value="40"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="FontFamily" Value="{StaticResource GraduateFont}"/>
            <Setter Property="FontSize" Value="18"/>
            <Setter Property="Margin" Value="0, 10"/>
            <Setter Property="Background" Value="#372F3C"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Grid>
                            <Rectangle x:Name="PART_BottomLine" Fill="{TemplateBinding
Background}" />
                            <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center" />
                        </Grid>
                        <ControlTemplate.Triggers>
                            <Trigger Property="IsMouseOver" Value="True">
                                <Setter TargetName="PART_BottomLine" Property="Fill"
Value="#463F4E"/>
                            </Trigger>
                        </ControlTemplate.Triggers>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>

        <Style x:Key="DataGridColumnHeaderStyle" TargetType="DataGridColumnHeader">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="DataGridColumnHeader">
                        <Grid Background="#4D4855">
                            <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                        </Grid>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>

        <Style x:Key="DataGridRowHeaderStyle" TargetType="DataGridRowHeader">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="DataGridRowHeader">
                        <Grid Background="#4D4855">

```



```

                                <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                                </Grid>
                                </ControlTemplate>
                                </Setter.Value>
                                </Setter>
                                </Style>

                                <Style x:Key="DataGridCellStyle" TargetType="DataGridCell">
                                    <Setter Property="Background" Value="#4D4855"/>
                                    <Setter Property="Foreground" Value="White"/>
                                    <Setter Property="BorderBrush" Value="#585858"/>
                                    <Setter Property="BorderThickness" Value="0,0,1,1"/>
                                    <Setter Property="FontSize" Value="15"/>
                                    <Setter Property="Template">
                                        <Setter.Value>
                                            <ControlTemplate TargetType="{x:Type DataGridCell}">
                                                <Border Background="{TemplateBinding Background}"
BorderBrush="{TemplateBinding BorderBrush}"
BorderThickness="{TemplateBinding BorderThickness}"
Padding="{TemplateBinding Padding}"
SnapsToDevicePixels="True">
                                                    <ContentPresenter SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
HorizontalAlignment="Center"
VerticalAlignment="Center" />
                                                </Border>
                                            </ControlTemplate>
                                        </Setter.Value>
                                    </Setter>
                                </Style>

                                <!-- Style for selected cell -->
                                <Style x:Key="DataGridCellSelectedStyle" TargetType="DataGridCell">
                                    <Setter Property="Background" Value="#4D4855"/>
                                    <Setter Property="BorderBrush" Value="#585858"/>
                                    <Setter Property="BorderThickness" Value="0,0,1,1"/>
                                    <Setter Property="Padding" Value="10"/>
                                </Style>

                                <!-- Background cell animation -->
                                <Style TargetType="DataGridCell" x:Key="DataGridCellHoverStyle">
                                    <Style.Triggers>
                                        <Trigger Property="IsMouseOver" Value="True">
                                            <Setter Property="Background" Value="#372F3C"/>
                                        </Trigger>
                                    </Style.Triggers>
                                </Style>

                                <Style TargetType="DataGrid" x:Key="CustomDataGridStyle">
                                    <Setter Property="Background" Value="{StaticResource
ApplicationBackgroundGradient}" />
                                    <Setter Property="BorderBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
                                    <Setter Property="BorderThickness" Value="1" />
                                    <Setter Property="RowBackground" Value="{StaticResource
ApplicationBackgroundGradient}" />
                                    <Setter Property="AlternatingRowBackground" Value="{StaticResource
ApplicationBackgroundGradient}" />
                                    <Setter Property="Foreground" Value="White" />
                                </Style>

```

```

        <Setter Property="HorizontalGridLinesBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
        <Setter Property="VerticalGridLinesBrush" Value="{StaticResource
ApplicationBackgroundGradient}" />
    </Style>

    <Style x:Key="AddButtonStyle" TargetType="Button">
        <Setter Property="Width" Value="150"/>
        <Setter Property="Height" Value="40"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontFamily" Value="{StaticResource GraduateFont}"/>
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="Margin" Value="0, 10"/>
        <Setter Property="Background" Value="#007ACC"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border x:Name="border" Background="{TemplateBinding
Background}" BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}"
                    CornerRadius="10">
                        <ContentPresenter HorizontalAlignment="Center"
                        VerticalAlignment="Center"/>
                    </Border>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsMouseOver" Value="True">
                            <Trigger.EnterActions>
                                <BeginStoryboard>
                                    <Storyboard>
                                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                                        To="#00546B" Duration="0:0:1"/>
                                    </Storyboard>
                                </BeginStoryboard>
                            </Trigger.EnterActions>
                            <Trigger.ExitActions>
                                <BeginStoryboard>
                                    <Storyboard>
                                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                                        To="#007ACC" Duration="0:0:1"/>
                                    </Storyboard>
                                </BeginStoryboard>
                            </Trigger.ExitActions>
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

    <ControlTemplate x:Key="ComboBoxToggleButton" TargetType="{x:Type
ToggleButton}">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition Width="20" />
            </Grid.ColumnDefinitions>
        </Grid>
    </ControlTemplate>

```

```

        </Grid.ColumnDefinitions>
        <Border
            x:Name="Border"
            Grid.ColumnSpan="2"
            CornerRadius="0"
            Background="#FF3F3F3F"
            BorderBrush="#FF97A0A5"
            BorderThickness="1" />
        <Border
            Grid.Column="0"
            CornerRadius="0"
            Margin="1"
            Background="#FF3F3F3F"
            BorderBrush="#FF97A0A5"
            BorderThickness="0,0,1,0" />
        <Path
            x:Name="Arrow"
            Grid.Column="1"
            Fill="White"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Data="M0,0 L0,3 L4,6 L8,2 L8,0 L4,4 z"
            />
    </Grid>
    <ControlTemplate.Triggers>
        <Trigger Property="ToggleButton.IsMouseOver" Value="true">
            <Setter TargetName="Border" Property="Background" Value="#808080"
        />

        </Trigger>
        <Trigger Property="ToggleButton.IsChecked" Value="true">
            <Setter TargetName="Border" Property="Background" Value="#E0E0E0"
        />

        </Trigger>
        <Trigger Property="IsEnabled" Value="False">
            <Setter TargetName="Border" Property="Background" Value="#EEEEEE"
        />

            <Setter TargetName="Border" Property="BorderBrush" Value="#AAAAAA"
        />

            <Setter Property="Foreground" Value="#888888"/>
            <Setter TargetName="Arrow" Property="Fill" Value="#888888" />
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>

<ControlTemplate x:Key="ComboBoxTextBox" TargetType="{x:Type TextBox}">
    <Border x:Name="PART_ContentHost" Focusable="False"
Background="{TemplateBinding Background}" />
</ControlTemplate>

<Style x:Key="{x:Type ComboBox}" TargetType="{x:Type ComboBox}">
    <Setter Property="SnapsToDevicePixels" Value="true"/>
    <Setter Property="OverridesDefaultStyle" Value="true"/>
    <Setter Property="ScrollViewer.HorizontalScrollBarVisibility"
Value="Auto"/>
    <Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Auto"/>
    <Setter Property="ScrollViewer.CanContentScroll" Value="true"/>
    <Setter Property="MinWidth" Value="100"/>
    <Setter Property="MinHeight" Value="20"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="FontFamily" Value="{StaticResource DramatungSC}"/>
    <Setter Property="Template">

```

```

<Setter.Value>
  <ControlTemplate TargetType="{x:Type ComboBox}">
    <Grid>
      <ToggleButton
        Name="ToggleButton"
        Template="{StaticResource ComboBoxToggleButton}"
        Grid.Column="2"
        Focusable="false"
        IsChecked="{Binding
Path=IsDropDownOpen,Mode=TwoWay,RelativeSource={RelativeSource TemplatedParent}}"
        ClickMode="Press">
      </ToggleButton>
      <ContentPresenter Name="ContentSite"
IsHitTestVisible="False" Content="{TemplateBinding SelectionBoxItem}"
ContentTemplate="{TemplateBinding
SelectionBoxItemTemplate}"
ContentTemplateSelector="{TemplateBinding
ItemTemplateSelector}"
Margin="3,3,23,3"
VerticalAlignment="Center"
HorizontalAlignment="Left" />
      <TextBox x:Name="PART_EditableTextBox"
Style="{x:Null}"
Template="{StaticResource ComboBoxTextBox}"
HorizontalAlignment="Left"
VerticalAlignment="Center"
Margin="3,3,23,3"
Focusable="True"
Background="#FF3F3F3F"
Foreground="Green"
Visibility="Hidden"
IsReadOnly="{TemplateBinding IsReadOnly}"/>
      <Popup
        Name="Popup"
        Placement="Bottom"
        IsOpen="{TemplateBinding IsDropDownOpen}"
        AllowsTransparency="True"
        Focusable="False"
        PopupAnimation="Slide">

        <Grid Name="DropDown"
          SnapsToDevicePixels="True"
          MinWidth="{TemplateBinding ActualWidth}"
          MaxHeight="{TemplateBinding MaxDropDownHeight}">
          <Border
            x:Name="DropDownBorder"
            Background="#FF3F3F3F"

            BorderThickness="1"
            BorderBrush="#888888"/>

          <ScrollViewer Margin="4,6,4,6"
SnapsToDevicePixels="True">
            <StackPanel IsItemsHost="True"
KeyboardNavigation.DirectionalNavigation="Contained" />
          </ScrollViewer>
        </Grid>
      </Popup>
    </Grid>
  <ControlTemplate.Triggers>
    <Trigger Property="HasItems" Value="false">

```

```

        <Setter TargetName="DropDownBorder"
Property="MinHeight" Value="95"/>
    </Trigger>
    <Trigger Property="IsEnabled" Value="false">
        <Setter Property="Foreground" Value="#888888"/>
    </Trigger>
    <Trigger Property="IsGrouping" Value="true">
        <Setter Property="ScrollViewer.CanContentScroll"
Value="false"/>
    </Trigger>
    <Trigger SourceName="Popup"
Property="Popup.AllowsTransparency" Value="true">
        <Setter TargetName="DropDownBorder"
Property="CornerRadius" Value="0"/>
        <Setter TargetName="DropDownBorder" Property="Margin"
Value="0,2,0,0"/>
    </Trigger>
    <Trigger Property="IsEditable" Value="true">
        <Setter Property="IsTabStop" Value="false"/>
        <Setter TargetName="PART_EditableTextBox"
Property="Visibility" Value="Visible"/>
        <Setter TargetName="ContentSite" Property="Visibility"
Value="Hidden"/>
    </Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
<Style.Triggers>
</Style.Triggers>
</Style>

<!-- SimpleStyles: ComboBoxItem -->
<Style x:Key="{x:Type ComboBoxItem}" TargetType="{x:Type ComboBoxItem}">
    <Setter Property="SnapsToDevicePixels" Value="true"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="OverridesDefaultStyle" Value="true"/>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type ComboBoxItem}">
                <Border Name="Border"
                    Padding="2"
                    SnapsToDevicePixels="true">
                    <ContentPresenter />
                </Border>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsHighlighted" Value="true">
                        <Setter TargetName="Border" Property="Background"
Value="#FF4F4F4F"/>
                    </Trigger>
                    <Trigger Property="IsEnabled" Value="false">
                        <Setter Property="Foreground" Value="#888888"/>
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

<Style TargetType="TextBox" x:Key="SearchTextBox">
    <Setter Property="Padding" Value="15,2,2,2"/>

```

```

<Setter Property="Foreground" Value="White"/>
<Setter Property="FontSize" Value="16"/>
<Setter Property="BorderThickness" Value="1"/>
<Setter Property="BorderBrush" Value="Bisque"/>
<Setter Property="Template">
    <Setter.Value>
        <ControlTemplate TargetType="{x:Type TextBox}">
            <Border BorderBrush="{TemplateBinding BorderBrush}"
                BorderThickness="{TemplateBinding BorderThickness}"
                CornerRadius="4">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="Auto"/>
                        <ColumnDefinition Width="*/>
                    </Grid.ColumnDefinitions>

                    <Viewbox Width="24" Height="24" Margin="2, 0, 0, 0">
                        <Canvas Width="20" Height="20">
                            <Path Fill="White" Data="M8 15.5a7.5 7.5 0 1 0
0-15 7.5 7.5 0 0 0 15Zm11.707 2.793-4-4a1 1 0 0 0-1.414 1.414l4 4a1 1 0 0 0 1.414-
1.414Z"/>
                        </Canvas>
                    </Viewbox>

                    <Border Grid.Column="0" Width="2" Background="Gray"
Margin="40,0,0,0"/>

                    <ScrollViewer x:Name="PART_ContentHost" Grid.Column="1"
Margin="0,0,2,0" VerticalAlignment="Center" Focusable="false"/>

                </Grid>
            </Border>
            <ControlTemplate.Triggers>
                <Trigger Property="IsEnabled" Value="false">
                    <Setter Property="Opacity" Value="0.5"/>
                </Trigger>
            </ControlTemplate.Triggers>
        </ControlTemplate>
    </Setter.Value>
</Setter>
</Style>
</Window.Resources>

<Window.DataContext>
    <local1:CriminalGroupsViewModel/>
</Window.DataContext>

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition/>
    </Grid.RowDefinitions>

    <Label Content="[ICIS] Criminal Groups" Grid.Row="0" Foreground="White"
        FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
        FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left"/>
    <Image Source="pack://application:,,,/Resources/Images/interpol_logo.png"
Width="32" Height="32"
        HorizontalAlignment="Center"/>
    <Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
        Height="40" Width="40" Margin="0"

```

					122 – КР.2023.01.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

<Grid Grid.Row="1">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Column="0" Width="170">
        <StackPanel.Background>
            <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
                <GradientStop Offset="0" Color="#332F3B" />
                <GradientStop Offset="1" Color="#463F4E" />
            </LinearGradientBrush>
        </StackPanel.Background>

        <Button Content="Main" Style="{StaticResource MenuButtonStyle}"/>
        <Button Content="Criminals" Style="{StaticResource MenuButtonStyle}"/>
        <Button Content="Agents" Style="{StaticResource MenuButtonStyle}"/>
    </StackPanel>

    <!--Main content section here -->

    <Grid Grid.Row="1" Grid.Column="1">
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="2*" />
        </Grid.RowDefinitions>

        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <TextBox Width="180" Height="35" VerticalAlignment="Bottom"
HorizontalAlignment="Center"
            Style="{StaticResource SearchTextBox}" Margin="0, 10"
            Text="{Binding GroupNameFilterText,
UpdateSourceTrigger=PropertyChanged}"/>

        <Button Grid.Column="2" Style="{StaticResource AddButtonStyle}"
            Command="{Binding AddCrimeGroup}"
            Content="Add" FontSize="18" FontFamily="{StaticResource
GraduateFont}"
            Height="40" Margin="20" VerticalAlignment="Bottom"/>

        <DataGrid Style="{StaticResource CustomDataGridStyle}"
            ItemsSource="{Binding FilteredCriminalGroups}"
            CellStyle="{StaticResource DataGridCellStyle}"
            ColumnHeaderStyle="{StaticResource DataGridColumnHeaderStyle}"
            RowHeaderStyle="{StaticResource DataGridRowHeaderStyle}"
            Grid.Row="1" RowHeight="30"
            Grid.ColumnSpan="3" Margin="10" AutoGenerateColumns="False"
CanUserAddRows="False">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Group name" Width="*"
Binding="{Binding Name}"/>

```

```

Leader}"/>
<DataGridTextColumn Header="Leader" Width="*" Binding="{Binding
Leader}"/>

<DataGridTemplateColumn Header="Activities" Width="*">
    <DataGridTemplateColumn.CellTemplate>
        <DataTemplate>
            <ComboBox ItemsSource="{Binding Activities}"
SelectedIndex="0"/>
        </DataTemplate>
    </DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Countries" Width="*">
    <DataGridTemplateColumn.CellTemplate>
        <DataTemplate>
            <ComboBox ItemsSource="{Binding Countries}"
SelectedIndex="0"/>
        </DataTemplate>
    </DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Details" Width="*">
    <DataGridTemplateColumn.CellTemplate>
        <DataTemplate>
            <Button Content="Details" MinWidth="90"
Command="{Binding Path=DataContext.ViewDetailsCommand,
RelativeSource= {RelativeSource FindAncestor,
AncestorType={x:Type DataGrid}}}"
CommandParameter="{Binding}"/>
        </DataTemplate>
    </DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>

</DataGrid.Columns>
</DataGrid>
</Grid>
</Grid>
</Grid>
</Window>

```

Файл Criminal_Information.xaml

```

<Window x:Class="Interpol_Card_Index_System.Views.Criminal_Information"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:Interpol_Card_Index_System.Views"
mc:Ignorable="d"
Height="450" Width="600" ResizeMode="NoResize" WindowStyle="None"
WindowStartupLocation="CenterScreen"
MouseDown="Window_MouseDown">
    <Window.Background>
        <StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
    </Window.Background>

    <Window.Resources>
        <Style TargetType="TextBox">
            <Setter Property="Background" Value="#2C3E50"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="BorderThickness" Value="1"/>
        </Style>
    </Window.Resources>

```



```

        <Setter Property="BorderBrush" Value="#34495E"/>
        <Setter Property="Padding" Value="5"/>
        <Setter Property="Margin" Value="10,5"/>
        <Setter Property="FontFamily" Value="{StaticResource DramaturgSC}"/>
        <Setter Property="FontSize" Value="14"/>
    </Style>
</Window.Resources>

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition/>
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0" Foreground="White" Margin="10, 0"
        FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
        FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left">
        <Run Text="Criminal #"/>
        <Run Text="{Binding ID, Mode=OneWay}"/>
    </TextBlock>

    <Image Source="pack://application:,,,/Resources/Images/interpol_logo.png"
Width="32" Height="32"
        HorizontalAlignment="Center"/>
    <Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
        Height="40" Width="40" Margin="0"
        HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

    <StackPanel Grid.Row="1" Margin="10">
        <DockPanel HorizontalAlignment="Center">
            <Image Height="128" Width="128" Stretch="Uniform" IsEnabled="False"
                Source="{Binding PhotoPath}"/>
        </DockPanel>

        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition/>
                <ColumnDefinition/>
            </Grid.ColumnDefinitions>

            <StackPanel Grid.Column="0" Margin="5, 0">
                <TextBlock Text="{Binding FullName, StringFormat=Full Name: {0}}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
                <TextBlock Text="{Binding Alias, StringFormat=Alias: {0}}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
                <TextBlock Text="{Binding DateOfBirth, StringFormat=Date of Birth:
{0}}" Foreground="Wheat" FontSize="16"/>
                <TextBlock Text="{Binding Nationality, StringFormat=Nationality:
{0}}" Foreground="Wheat" Background="Transparent" FontSize="16"/>
                <TextBox Text="{Binding SpecialCharacteristics}" Margin="0, 25"
MinHeight="80" AcceptsReturn="True"/>
            </StackPanel>

            <StackPanel Grid.Column="1" Margin="10, 0">
                <TextBlock Text="{Binding Gender, StringFormat=Gender: {0}}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
                <TextBlock Text="{Binding HairColor, StringFormat=Hair Color: {0}}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
                <TextBlock Text="{Binding EyeColor}" Foreground="Wheat"
FontSize="16"/>
            </StackPanel>
        </Grid>
    </StackPanel>

```

```

        <TextBlock Text="{Binding Height, StringFormat={}{0} m.}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
        <TextBlock Text="{Binding Weight, StringFormat={}{0} kg.}"
Foreground="Wheat" Background="Transparent" FontSize="16"/>
        <TextBox Text="{Binding CrimeDescription}" Margin="0, 5"
MinHeight="80" AcceptsReturn="True"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">
    <Button Width="60" Height="50"
        Style="{StaticResource MenuButtonStyle}"
        FontSize="14" Foreground="Black">
        <Button.Content>
            <Viewbox Width="32" Height="32">
                <Canvas Width="20" Height="20">
                    <Path Fill="Gray" Data="M4.5 11H4v1h.5a.5.5 0 0
0 0-1ZM7 5V.13a2.96 2.96 0 0 0-1.293.749L2.879 3.707A2.96 2.96 0 0 0 2.13 5H7Zm3.375
6H10v3h.375a.624.624 0 0 0 .625-.625v-1.75a.624.624 0 0 0-.625-.625Z"/>
                    <Path Fill="Gray" Data="M19 7h-1V2a1.97 1.97 0
0 0-1.933-2H9v5a2 2 0 0 1-2 2H1a1 1 0 0 0-1 1v9a1 1 0 0 0 1 1h1a1.969 1.969 0 0 0 1.933
2h12.134c1.1 0 1.7-1.236 1.856-1.614a.988.988 0 0 0 .07-.386H19a1 1 0 0 0 1-1V8a1 1 0 0
0-1-1ZM4.5 14H4v1a1 1 0 1 1-2 0v-5a1 1 0 0 1 1-1h1.5a2.5 2.5 0 1 1 0 5Zm8.5-.625A2.63
2.63 0 0 1 10.375 16H9a1 1 0 0 1-1-1v-5a1 1 0 0 1 1-1h1.375A2.63 2.63 0 0 1 13
11.625v1.75ZM17 12a1 1 0 0 1 0 2h-1v1a1 1 0 0 1-2 0v-5a1 1 0 0 1 1-1h2a1 1 0 1 1 0 2h-
1v1h1Z"/>
                </Canvas>
            </Viewbox>
        </Button.Content>
    </Button>
    <Button Width="60" Height="50"
        Margin="10, 0, 0, 0"
        Style="{StaticResource MenuButtonStyle}"
        FontSize="14" Foreground="Black">
        <Button.Content>
            <Viewbox Width="32" Height="32">
                <Canvas Width="20" Height="20">
                    <Path Fill="Gray" Data="M 3 0 L 3 12 L 4 12 L 4
1 L 12 1 L 12 4 L 12 5 L 16 5 L 16 12 L 17 12 L 17 5 L 17 4 L 13 0 L 12 0 L 3 0 z M 13
1.3535156 L 15.646484 4 L 13 4 L 13 1.3535156 z M 4 13 L 4 16 L 3.9921875 16.130859 L
3.9667969 16.257812 L 3.9238281 16.382812 L 3.8652344 16.5 L 3.7929688 16.607422 L
3.7070312 16.707031 L 3.609375 16.792969 L 3.5 16.865234 L 3.3828125 16.923828 L
3.2597656 16.964844 L 3.1308594 16.992188 L 3 17 L 2.8691406 16.992188 L 2.7402344
16.964844 L 2.6171875 16.923828 L 2.5 16.865234 L 2.390625 16.792969 L 2.2929688
16.707031 L 2.2070312 16.607422 L 2.1347656 16.5 L 2.0761719 16.382812 L 2.0332031
16.257812 L 2.0078125 16.130859 L 2 16 L 1 16 L 1.0078125 16.183594 L 1.0351562
16.367188 L 1.0761719 16.546875 L 1.1347656 16.722656 L 1.2089844 16.890625 L 1.3007812
17.052734 L 1.4042969 17.205078 L 1.5214844 17.347656 L 1.6523438 17.478516 L 1.7949219
17.595703 L 1.9472656 17.699219 L 2.1074219 17.791016 L 2.2773438 17.865234 L 2.453125
17.923828 L 2.6328125 17.964844 L 2.8164062 17.992188 L 3 18 L 3.1835938 17.992188 L
3.3671875 17.964844 L 3.546875 17.923828 L 3.7226562 17.865234 L 3.8925781 17.791016 L
4.0527344 17.699219 L 4.2050781 17.595703 L 4.3476562 17.478516 L 4.4785156 17.347656 L
4.5957031 17.205078 L 4.6992188 17.052734 L 4.7910156 16.890625 L 4.8652344 16.722656 L
4.9238281 16.546875 L 4.9648438 16.367188 L 4.9921875 16.183594 L 5 16 L 5 13 L 4 13 z
M 7.5 13 A 1.5 1.4999999 0 0 0 6 14.5 A 1.5 1.4999999 0 0 0 7.5 16 L 8.5 16 A 0.5 0.5 0
0 1 9 16.5 A 0.5 0.5 0 0 1 8.5 17 L 6 17 L 6 18 L 8.5 18 A 1.5 1.4999999 0 0 0 10 16.5
A 1.5 1.4999999 0 0 0 8.5 15 L 7.5 15 A 0.5 0.5 0 0 1 7 14.5 A 0.5 0.5 0 0 1 7.5 14 L
10 14 L 10 13 L 7.5 13 z M 13 13 L 12.816406 13.007812 L 12.632812 13.033203 L
12.453125 13.076172 L 12.277344 13.134766 L 12.107422 13.208984 L 11.947266 13.298828 L
11.794922 13.404297 L 11.652344 13.521484 L 11.521484 13.652344 L 11.404297 13.794922 L
11.300781 13.947266 L 11.208984 14.107422 L 11.134766 14.277344 L 11.076172 14.451172 L
11.035156 14.632812 L 11.007812 14.816406 L 11 15 L 11 16 L 11.007812 16.183594 L
11.035156 16.367188 L 11.076172 16.546875 L 11.134766 16.722656 L 11.208984 16.890625 L

```

					122 – КР.2023.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

        </Canvas>
    </Viewbox>
</Button.Content>
</Button>
</StackPanel>
</StackPanel>
</Grid>
</StackPanel>
</Grid>
</Window>

```

```
<Window x:Class="Interpol_Card_Index_System.Views.CriminalGroup_Information"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Interpol_Card_Index_System.Views"
        mc:Ignorable="d"
        Height="350" Width="600" ResizeMode="NoResize" WindowStyle="None"
        WindowStartupLocation="CenterScreen"
        MouseDown="Window_MouseDown">
    <Window.Background>
        <StaticResourceExtension ResourceKey="ApplicationBackgroundGradient"/>
    </Window.Background>

    <Window.Resources>
        <Style TargetType="TextBlock">
            <Setter Property="Margin" Value="0, 10"/>
        </Style>
    </Window.Resources>

    <Grid>
```

					122 – КР.2023.01.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition/>
</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Foreground="White" Margin="10, 0"
    FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
    FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Left">
    <Run Text="Crime Group #"/>
    <Run Text="{Binding ID, Mode=OneWay}"/>
</TextBlock>

<Image Source="pack://application:,,,/Resources/Images/interpol_logo.png"
Width="32" Height="32"
    HorizontalAlignment="Center"/>
<Button Content="X" Grid.Row="0" Background="#182B3A" Foreground="White"
    Height="40" Width="40" Margin="0"
    HorizontalAlignment="Right" VerticalAlignment="Center"
Click="Button_Click"/>

<StackPanel Grid.Row="1" Margin="10">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>

        <StackPanel Grid.Column="0" Margin="5, 0">
            <TextBlock Text="{Binding Name, StringFormat=Name: {0}}"
Foreground="Wheat" Background="Transparent"
                FontSize="16" FontFamily="{StaticResource
GraduateFont}"/>
            <TextBlock Text="{Binding Leader, StringFormat=Leader: {0}}"
Foreground="Wheat" Background="Transparent"
                FontSize="16" FontFamily="{StaticResource
GraduateFont}"/>
            <Separator/>
            <TextBlock Text="Activities: " FontFamily="{StaticResource
GraduateFont}" Foreground="Wheat" FontSize="18"/>
            <ComboBox ItemsSource="{Binding Activities}" SelectedIndex="0"
MinWidth="120"
                Foreground="Black" FontSize="16" Margin="0, 10"/>
            <TextBlock Text="Countries: " Foreground="Wheat"
                FontSize="18" FontFamily="{StaticResource
GraduateFont}"/>
            <ComboBox ItemsSource="{Binding Countries}" SelectedIndex="0"
MinWidth="120"
                Foreground="Black" FontSize="16" Margin="0, 10"/>
        </StackPanel>

        <StackPanel Grid.Column="1" Margin="10, 0">
            <TextBox Text="{Binding Description}" Grid.ColumnSpan="2"
Margin="0, 25" MinHeight="160" AcceptsReturn="True"/>

            <StackPanel Orientation="Horizontal" HorizontalAlignment="Right">
                <Button Width="60" Height="50"
                    Style="{StaticResource MenuButtonStyle}"
                    FontSize="14" Foreground="Black">
                    <Button.Content>
                        <Viewbox Width="32" Height="32">
                            <Canvas Width="20" Height="20">

```

```

<Path Fill="Gray" Data="M4.5 11H4v1h.5a.5.5 0 0
0 0-1ZM7 5V.13a2.96 2.96 0 0 0-1.293.749L2.879 3.707A2.96 2.96 0 0 0 2.13 5H7Zm3.375
6H10v3h.375a.624.624 0 0 0 .625-.625v-1.75a.624.624 0 0 0-.625-.625Z"/>
<Path Fill="Gray" Data="M19 7h-1V2a1.97 1.97 0
0 0-1.933-2H9v5a2 2 0 0 1-2 2H1a1 1 0 0 0-1 1v9a1 1 0 0 0 1 1h1a1.969 1.969 0 0 0 1.933
2h12.134c1.1 0 1.7-1.236 1.856-1.614a.988.988 0 0 0 .07-.386H19a1 1 0 0 0 1-1V8a1 1 0 0
0-1-1ZM4.5 14H4v1a1 1 0 1 1-2 0v-5a1 1 0 0 1 1-1h1.5a2.5 2.5 0 1 1 0 5Zm8.5-.625A2.63
2.63 0 0 1 10.375 16H9a1 1 0 0 1-1-1v-5a1 1 0 0 1 1-1h1.375A2.63 2.63 0 0 1 13
11.625v1.75ZM17 12a1 1 0 0 1 0 2h-1v1a1 1 0 0 1-2 0v-5a1 1 0 0 1 1-1h2a1 1 0 1 1 0 2h-
1v1h1Z"/>
</Canvas>
</Viewbox>
</Button.Content>
</Button>
<Button Width="60" Height="50"
Margin="10, 0, 0, 0"
Style="{StaticResource MenuButtonStyle}"
FontSize="14" Foreground="Black">
<Button.Content>
<Viewbox Width="32" Height="32">
<Canvas Width="20" Height="20">
<Path Fill="Gray" Data="M 3 0 L 3 12 L 4 12 L 4
1 L 12 1 L 12 4 L 12 5 L 16 5 L 16 12 L 17 12 L 17 5 L 17 4 L 13 0 L 12 0 L 3 0 z M 13
1.3535156 L 15.646484 4 L 13 4 L 13 1.3535156 z M 4 13 L 4 16 L 3.9921875 16.130859 L
3.9667969 16.257812 L 3.9238281 16.382812 L 3.8652344 16.5 L 3.7929688 16.607422 L
3.7070312 16.707031 L 3.609375 16.792969 L 3.5 16.865234 L 3.3828125 16.923828 L
3.2597656 16.964844 L 3.1308594 16.992188 L 3 17 L 2.8691406 16.992188 L 2.7402344
16.964844 L 2.6171875 16.923828 L 2.5 16.865234 L 2.390625 16.792969 L 2.2929688
16.707031 L 2.2070312 16.607422 L 2.1347656 16.5 L 2.0761719 16.382812 L 2.0332031
16.257812 L 2.0078125 16.130859 L 2 16 L 1 16 L 1.0078125 16.183594 L 1.0351562
16.367188 L 1.0761719 16.546875 L 1.1347656 16.722656 L 1.2089844 16.890625 L 1.3007812
17.052734 L 1.4042969 17.205078 L 1.5214844 17.347656 L 1.6523438 17.478516 L 1.7949219
17.595703 L 1.9472656 17.699219 L 2.1074219 17.791016 L 2.2773438 17.865234 L 2.453125
17.923828 L 2.6328125 17.964844 L 2.8164062 17.992188 L 3 18 L 3.1835938 17.992188 L
3.3671875 17.964844 L 3.546875 17.923828 L 3.7226562 17.865234 L 3.8925781 17.791016 L
4.0527344 17.699219 L 4.2050781 17.595703 L 4.3476562 17.478516 L 4.4785156 17.347656 L
4.5957031 17.205078 L 4.6992188 17.052734 L 4.7910156 16.890625 L 4.8652344 16.722656 L
4.9238281 16.546875 L 4.9648438 16.367188 L 4.9921875 16.183594 L 5 16 L 5 13 L 4 13 z
M 7.5 13 A 1.5 1.4999999 0 0 0 6 14.5 A 1.5 1.4999999 0 0 0 7.5 16 L 8.5 16 A 0.5 0.5 0
0 1 9 16.5 A 0.5 0.5 0 0 1 8.5 17 L 6 17 L 6 18 L 8.5 18 A 1.5 1.4999999 0 0 0 10 16.5
A 1.5 1.4999999 0 0 0 8.5 15 L 7.5 15 A 0.5 0.5 0 0 1 7 14.5 A 0.5 0.5 0 0 1 7.5 14 L
10 14 L 10 13 L 7.5 13 z M 13 13 L 12.816406 13.007812 L 12.632812 13.033203 L
12.453125 13.076172 L 12.277344 13.134766 L 12.107422 13.208984 L 11.947266 13.298828 L
11.794922 13.404297 L 11.652344 13.521484 L 11.521484 13.652344 L 11.404297 13.794922 L
11.300781 13.947266 L 11.208984 14.107422 L 11.134766 14.277344 L 11.076172 14.451172 L
11.035156 14.632812 L 11.007812 14.816406 L 11 15 L 11 16 L 11.007812 16.183594 L
11.035156 16.367188 L 11.076172 16.546875 L 11.134766 16.722656 L 11.208984 16.890625 L
11.300781 17.052734 L 11.404297 17.205078 L 11.521484 17.347656 L 11.652344 17.478516 L
11.794922 17.595703 L 11.947266 17.699219 L 12.107422 17.791016 L 12.277344 17.865234 L
12.453125 17.923828 L 12.632812 17.964844 L 12.816406 17.992188 L 13 18 L 13.183594
17.992188 L 13.367188 17.964844 L 13.546875 17.923828 L 13.722656 17.865234 L 13.892578
17.791016 L 14.052734 17.699219 L 14.205078 17.595703 L 14.347656 17.478516 L 14.478516
17.347656 L 14.595703 17.205078 L 14.699219 17.052734 L 14.791016 16.890625 L 14.865234
16.722656 L 14.923828 16.546875 L 14.964844 16.367188 L 14.992188 16.183594 L 15 16 L
15 15 L 14.992188 14.816406 L 14.964844 14.632812 L 14.923828 14.451172 L 14.865234
14.277344 L 14.791016 14.107422 L 14.699219 13.947266 L 14.595703 13.794922 L 14.478516
13.652344 L 14.347656 13.521484 L 14.205078 13.404297 L 14.052734 13.298828 L 13.892578
13.208984 L 13.722656 13.134766 L 13.546875 13.076172 L 13.367188 13.033203 L 13.183594
13.007812 L 13 13 z M 16.513672 13 A 0.50005 0.50005 0 0 0 16 13.5 L 16 18 L 17 18 L 17
15.001953 L 19.099609 17.800781 A 0.50005 0.50005 0 0 0 20 17.5 L 20 13 L 19 13 L 19
15.998047 L 16.900391 13.199219 A 0.50005 0.50005 0 0 0 16.513672 13 z M 13 14 L

```

					122 – КР.2023.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49


```

        Path=(Validation.Errors)[0].ErrorContent}" />
    </Trigger>
</Style.Triggers>
</Style>

<Style TargetType="RadioButton">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="RadioButton">
                <Grid VerticalAlignment="Center" HorizontalAlignment="Left">
                    <Border BorderBrush="#372F3C" BorderThickness="1"
Width="20" Height="20">
                        <Grid>
                            <Rectangle x:Name="Square" Fill="White"/>
                            <Path x:Name="Check" Data="M0,5 L4,9 L10,0"
Stroke="Black" StrokeThickness="2" Visibility="Collapsed"/>
                        </Grid>
                    </Border>
                    <ContentPresenter VerticalAlignment="Center"
HorizontalAlignment="Right" Margin="65,0,0,0" />
                </Grid>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsChecked" Value="True">
                        <Setter TargetName="Square" Property="Fill"
Value="Wheat"/>
                        <Setter TargetName="Check" Property="Visibility"
Value="Hidden"/>
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
</Window.Resources>

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition/>
    </Grid.RowDefinitions>

    <Label Content="[ICIS] Add criminal" Grid.Row="0" Foreground="White"
        FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
        FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Center"/>
    <Separator Grid.Row="0" VerticalAlignment="Bottom"/>

    <Grid Grid.Row="1">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <Grid.RowDefinitions>
            <RowDefinition Height="4*" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <StackPanel>
            <StackPanel>
                <TextBlock Text="Fullname:" />
                <TextBox>

```

```

        <TextBox.Text>
            <Binding Path="NewCriminal.FullName">
                <Binding.ValidationRules>
                    <DataErrorValidationRule />
                </Binding.ValidationRules>
            </Binding>
        </TextBox.Text>
    </TextBox>
</StackPanel>

<StackPanel>
    <TextBlock Text="Alias:" />
    <TextBox>
        <TextBox.Text>
            <Binding Path="NewCriminal.Alias">
                <Binding.ValidationRules>
                    <DataErrorValidationRule />
                </Binding.ValidationRules>
            </Binding>
        </TextBox.Text>
    </TextBox>
</StackPanel>

<StackPanel>
    <TextBlock Text="Birthday:" />
    <DatePicker SelectedDate="{Binding SelectedBirthday, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}" />
</StackPanel>

<StackPanel>
    <TextBlock Text="Nationality:" />
    <TextBox>
        <TextBox.Text>
            <Binding Path="NewCriminal.Nationality">
                <Binding.ValidationRules>
                    <DataErrorValidationRule />
                </Binding.ValidationRules>
            </Binding>
        </TextBox.Text>
    </TextBox>
</StackPanel>

<StackPanel>
    <TextBlock Text="Gender:" />

    <StackPanel Orientation="Horizontal">
        <RadioButton Content="Male" Foreground="White"
GroupName="gender"
                IsChecked="{Binding SelectedGender, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged,
                Converter={StaticResource RadioButtonToBooleanConverter},
ConverterParameter=Male}" />
        <RadioButton Content="Female" Foreground="White"
GroupName="gender"
                IsChecked="{Binding SelectedGender, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged,
                Converter={StaticResource RadioButtonToBooleanConverter},
ConverterParameter=Female}" />
    </StackPanel>
</StackPanel>

```



```

        <StackPanel>
            <TextBlock Text="Hair color:"/>
            <TextBox>
                <TextBox.Text>
                    <Binding Path="NewCriminal.HairColor">
                        <Binding.ValidationRules>
                            <DataErrorValidationRule />
                        </Binding.ValidationRules>
                    </Binding>
                </TextBox.Text>
            </TextBox>
        </StackPanel>
    </StackPanel>

    <StackPanel Grid.Column="1">
        <StackPanel VerticalAlignment="Center">
            <TextBlock Text="Photo:"/>
            <Border Width="128" Height="128" BorderBrush="White"
BorderThickness="1">
                <Button Background="Transparent" Command="{Binding
OpenImageCommand}" />
            </Border>
        </StackPanel>

        <StackPanel>
            <TextBlock Text="Eye color:"/>
            <TextBox>
                <TextBox.Text>
                    <Binding Path="NewCriminal.EyeColor">
                        <Binding.ValidationRules>
                            <DataErrorValidationRule />
                        </Binding.ValidationRules>
                    </Binding>
                </TextBox.Text>
            </TextBox>
        </StackPanel>

        <StackPanel>
            <TextBlock Text="Height:"/>
            <TextBox>
                <TextBox.Text>
                    <Binding Path="NewCriminal.Height">
                        <Binding.ValidationRules>
                            <DataErrorValidationRule />
                        </Binding.ValidationRules>
                    </Binding>
                </TextBox.Text>
            </TextBox>
        </StackPanel>

        <StackPanel>
            <TextBlock Text="Weight:"/>
            <TextBox>
                <TextBox.Text>
                    <Binding Path="NewCriminal.Weight">
                        <Binding.ValidationRules>
                            <DataErrorValidationRule />
                        </Binding.ValidationRules>
                    </Binding>
                </TextBox.Text>
            </TextBox>
        </StackPanel>
    
```



```

        <Border Background="{TemplateBinding Background}"
            BorderBrush="{TemplateBinding BorderBrush}"
            BorderThickness="{TemplateBinding BorderThickness}"
            Padding="{TemplateBinding Padding}">
            <ScrollView>
                <ItemsPresenter/>
            </ScrollView>
        </Border>
    </ControlTemplate>
</Setter.Value>
</Setter>
</Style>
</Window.Resources>

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition/>
    </Grid.RowDefinitions>

    <Label Content="[ICIS] Add Criminal Group" Grid.Row="0" Foreground="White"
        FontFamily="{StaticResource GraduateFont}" FontWeight="Bold"
        FontSize="24" VerticalAlignment="Center" HorizontalAlignment="Center"/>
    <Separator Grid.Row="0" VerticalAlignment="Bottom"/>

    <Grid Grid.Row="1">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>

        <Grid.RowDefinitions>
            <RowDefinition Height="4*"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <StackPanel>
            <StackPanel>
                <TextBlock Text="Group name:"/>
                <TextBox Text="{Binding NewCriminalGroup.Name,
UpdateSourceTrigger=PropertyChanged}"/>
            </StackPanel>

            <StackPanel>
                <TextBlock Text="Leader:"/>
                <TextBox Text="{Binding NewCriminalGroup.Leader,
UpdateSourceTrigger=PropertyChanged}"/>
            </StackPanel>

            <StackPanel>
                <TextBlock Text="Description:"/>
                <TextBox Text="{Binding NewCriminalGroup.Description,
UpdateSourceTrigger=PropertyChanged}"
                    Height="120" AcceptsReturn="True"/>
            </StackPanel>
        </StackPanel>

        <StackPanel Grid.Column="1">
            <StackPanel>
                <TextBlock Text="Activities:"/>

```

```

        <ListBox x:Name="activitiesListView" SelectionMode="Multiple"
ItemsSource="{Binding AllActivities}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="SelectionChanged">
                <i:InvokeCommandAction Command="{Binding
SelectionActivitiesChangedCommand}" CommandParameter="{Binding SelectedItems,
ElementName=activitiesListView}" />
            </i:EventTrigger>
        </i:Interaction.Triggers>
    </ListBox>
</StackPanel>

    <StackPanel>
        <TextBlock Text="Countries:"/>
        <ListBox x:Name="countriesListBox" SelectionMode="Multiple"
ItemsSource="{Binding AllCountries}">
        <i:Interaction.Triggers>
            <i:EventTrigger EventName="SelectionChanged">
                <i:InvokeCommandAction Command="{Binding
SelectionCountriesChangedCommand}" CommandParameter="{Binding SelectedItems,
ElementName=countriesListBox}" />
            </i:EventTrigger>
        </i:Interaction.Triggers>
    </ListBox>
    </StackPanel>
</StackPanel>

    <StackPanel Grid.Row="1" Grid.ColumnSpan="2" Orientation="Horizontal"
        VerticalAlignment="Center" HorizontalAlignment="Center">
        <Button Content="Confirm" Style="{StaticResource MenuButtonStyle}"
Command="{Binding ConfirmCommand}" Margin="10, 0"/>
        <Button Content="Cancel" Style="{StaticResource MenuButtonStyle}"
Click="Button_Click"/>
    </StackPanel>
</Grid>
</Grid>
</Window>

```

Додаток В
С# основні класи

					122 – КР.2023.01.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас RepositoryService.cs

```
public class RepositoryService
{
    private static RepositoryService _instance;
    public static RepositoryService Instance => _instance ??= new
RepositoryService();

    private ObservableCollection<User> _usersList;
    private ObservableCollection<Criminal> _criminalsList;
    private ObservableCollection<CriminalGroup> _crimeGroups;

    public ObservableCollection<User> Users
    {
        get
        {
            return _usersList;
        }
    }

    public ObservableCollection<Criminal> Criminals
    {
        get
        {
            return _criminalsList;
        }
    }

    public ObservableCollection<CriminalGroup> CriminalGroups
    {
        get
        {
            return _crimeGroups;
        }
    }

    private RepositoryService() {
        _usersList = new ObservableCollection<User>()
        {
            new User() {Name = "Alex Winchester", Login = "Test123", Password =
"qwerty123", AccessLevel = 'B'},
            new User() {Name = "Sam Winchester", Login = "Test2546", Password =
"qwerty1234", AccessLevel = 'A'}
        };

        _crimeGroups = new ObservableCollection<CriminalGroup>()
        {
            new CriminalGroup("Sierra", "El Gato", new List<string>() {"Killers",
"Drugs"}, new List<string>() {"USA", "Spain" }, "Lorem ipsum...")
        };

        _criminalsList = new ObservableCollection<Criminal>
        {
            new Criminal() {FullName = "Jack Sparrow", Alias = "Sparrow",
DateOfBirth = DateTime.Parse("12.10.1982"), Nationality = "Ukrainian", Weight = 96,
AccessLevel = 'B', PhotoPath = "/Views/girl.jpg"},
            new Criminal() {FullName = "Ryan Gosling", Alias = "Sierra-6",
DateOfBirth = DateTime.Parse("24.07.1994"), Nationality = "USA", Weight = 84,
AccessLevel = 'A', PhotoPath = Path.Combine(Environment.CurrentDirectory,
"Criminal_Photos", "sierra.jpg")}
        };
    }
}
```

					122 – КР.2023.01.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public void AddUser(User userInstance)
    {
        _usersList.Add(userInstance);
    }

    public void AddCriminal(Criminal criminalInstance)
    {
        _criminalsList.Add(criminalInstance);
    }

    public void AddCriminalGroup(CriminalGroup criminalGroupInstance)
    {
        _crimeGroups.Add(criminalGroupInstance);
    }
}

```

Клас SessionService.cs

```

public class SessionService
{
    private static SessionService _instance;
    public static SessionService Instance => _instance ??= new SessionService();

    private User _currentUser;
    public User CurrentUser
    {
        get => _currentUser;
        set => _currentUser = value;
    }

    private SessionService() { }

    public void ResetSession()
    {
        _currentUser = null;
    }
}

```

Клас CollectionFilter.cs

```

public class CollectionFilter<T> : ObservableCollection<T>, ICollectionFilter<T>
{
    private ObservableCollection<T> _originalCollection;
    private List<Func<T, bool>> _filters;

    public CollectionFilter(ObservableCollection<T> collection)
    {
        _originalCollection = collection;
        _filters = new List<Func<T, bool>>();

        UpdateItems(_originalCollection);
    }

    public void AddFilter(Func<T, bool> predicate)
    {
        _filters.Add(predicate);
        ApplyFilters();
    }

    public void RemoveFilter(Func<T, bool> predicate)
    {
        _filters.Remove(predicate);
        ApplyFilters();
    }
}

```

```

    }

    private void ApplyFilters()
    {
        Clear();

        foreach (var item in _originalCollection.Where(item => _filters.All(filter
=> filter(item))))
        {
            Add(item);
        }
    }

    private void UpdateItems(IEnumerable<T> items)
    {
        foreach (var item in items)
        {
            Add(item);
        }
    }
}

```

Інтерфейс ICollectionFilter.cs

```

public interface ICollectionFilter<T>
{
    void AddFilter(Func<T, bool> predicate);
    void RemoveFilter(Func<T, bool> predicate);
}

```

Клас LoginWindowViewModel.cs

```

public class LoginWindowViewModel : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler? PropertyChanged;

    private string _username;
    public string Username
    {
        get
        {
            return _username;
        }
        set
        {
            _username = value;
            OnPropertyChanged();
        }
    }

    private string _password;
    public string Password
    {
        get
        {
            return _password;
        }
        set
        {
            _password = value;
            OnPropertyChanged();
        }
    }
}

```

					122 – КР.2023.01.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public RelayCommand LoginCommand { get; private set; }

public LoginWindowViewModel()
{
    LoginCommand = new RelayCommand(Login, CanLogin);
}

private bool CanLogin()
{
    return !string.IsNullOrEmpty(Username) &&
!string.IsNullOrEmpty>Password);
}

private void Login(object parameter)
{
    User user = FindUser(Username, Password);

    if (user != null)
    {
        MessageBox.Show($"You authorized in system as a {user.Name}!");
        SessionService.Instance.CurrentUser = user;

        Views.MainWindow mainWindow = new Views.MainWindow();
        mainWindow.Show();

        Application.Current.MainWindow.Close();
    }
    else
    {
        MessageBox.Show("Invalid username or password.");
    }
}

private User FindUser(string username, string password)
{
    return RepositoryService.Instance.Users.ToList().Find(u => u.Login == username
&& u.Password == password);
}

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

Клас MainWindowViewModel.cs

```

public class MainWindowViewModel: INotifyPropertyChanged
{
    public event PropertyChangedEventHandler? PropertyChanged;
    private DispatcherTimer _timer;

    public RelayCommand LogoutCommand { get; private set; }
    public RelayCommand ShowCriminalGroupWindow { get; private set; }
    public RelayCommand ShowCriminalWindow { get; private set; }

    public MainWindowViewModel()
    {
        LogoutCommand = new RelayCommand(Logout, IsUserAuthorized);
        ShowCriminalGroupWindow = new RelayCommand(CrimeWindowOpen, IsUserAuthorized);
        ShowCriminalWindow = new RelayCommand(CriminalWindowOpen, IsUserAuthorized);

        _timer = new DispatcherTimer();
        _timer.Interval = TimeSpan.FromSeconds(1);
        _timer.Tick += Timer_Tick;
    }
}

```

					122 – КР.2023.01.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        _timer.Start();
    }

    private string _currentDateAndTime;
    public string CurrentDateAndTime
    {
        get
        {
            return _currentDateAndTime;
        }
        set
        {
            _currentDateAndTime = value;
            OnPropertyChanged();
        }
    }

    public string CurrentUserName => SessionService.Instance.CurrentUser.Name;
    public char CurrentUserAccess => SessionService.Instance.CurrentUser.AccessLevel;

    public int CriminalsCount => RepositoryService.Instance.Criminals.Count;
    public int CriminalGroupsCount => RepositoryService.Instance.CriminalGroups.Count;

    private void Timer_Tick(object sender, EventArgs e)
    {
        UpdateCurrentDateTime();
    }

    private void UpdateCurrentDateTime()
    {
        CurrentDateAndTime = DateTime.Now.ToString();
    }

    private void CrimeWindowOpen(object parameter)
    {
        Views.CriminalGroupsWindow criminalGroupsWindow = new
Views.CriminalGroupsWindow();
        criminalGroupsWindow.Show();
    }

    private void CriminalWindowOpen(object parameter)
    {
        Views.CriminalWindow criminalWindow = new Views.CriminalWindow();
        criminalWindow.Show();
    }

    private void Logout(object parameter)
    {
        SessionService.Instance.ResetSession();
        Views.LoginWindow loginWindow = new Views.LoginWindow();
        Application.Current.MainWindow.Close();

        loginWindow.Show();
    }

    private bool IsUserAuthorized()
    {
        return SessionService.Instance.CurrentUser != null ? true : false;
    }

    protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }

```

					122 – КР.2023.01.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}

Клас CriminalWindowViewModel.cs
public class CriminalWindowViewModel : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    private CollectionFilter<Criminal> _filteredCriminals;
    private string _nameFilterText;

    public ObservableCollection<Criminal> Criminals
    {
        get { return RepositoryService.Instance.Criminals; }
    }

    public CollectionFilter<Criminal> FilteredCriminals
    {
        get => _filteredCriminals;
        set
        {
            _filteredCriminals = value;
            OnPropertyChanged(nameof(FilteredCriminals));
        }
    }

    public string NameFilterText
    {
        get => _nameFilterText;
        set
        {
            _nameFilterText = value;
            OnPropertyChanged(nameof(NameFilterText));
            FilterCommand.Execute(null);
        }
    }

    public ICommand AddCriminalCommand { get; private set; }
    public ICommand ViewDetailsCommand { get; private set; }
    public ICommand FilterCommand { get; private set; }

    public CriminalWindowViewModel()
    {
        _filteredCriminals = new CollectionFilter<Criminal>(Criminals);

        AddCriminalCommand = new RelayCommand(AddCriminal);
        ViewDetailsCommand = new RelayCommand(ViewDetails);

        FilterCommand = new RelayCommand(Filter);
    }

    private void ViewDetails(object parameter)
    {
        if (parameter is Criminal criminal)
        {
            if (SessionService.Instance.CurrentUser.AccessLevel > criminal.AccessLevel)
            {
                MessageBox.Show("You can't get access to this information!");
                return;
            }

            Criminal_Information criminal_InfoWindow = new
Criminal_Information(criminal);

            criminal_InfoWindow.Show();
        }
    }
}

```

					122 – КР.2023.01.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}

private void AddCriminal(object parameter)
{
    AddNewCriminalViewModel targetView = new AddNewCriminalViewModel();
    AddNewCriminal addNewCriminalWindow = new AddNewCriminal();
    addNewCriminalWindow.DataContext = targetView;

    addNewCriminalWindow.ShowDialog();

    if(targetView.IsResult)
        RepositoryService.Instance.AddCriminal(targetView.NewCriminal);
}

private void Filter(object parameter)
{
    _filteredCriminals.AddFilter(criminal =>
criminal.FullName.Contains(NameFilterText));

    OnPropertyChanged(nameof(NameFilterText));
}

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

Клас CriminalGroupsViewModel.cs

```

public class CriminalGroupsViewModel
{
    public event PropertyChangedEventHandler PropertyChanged;

    private CollectionFilter<CriminalGroup> _filteredCriminalGroups;
    private string _groupNameFilterText;

    public ObservableCollection<CriminalGroup> CriminalGroups
    {
        get { return RepositoryService.Instance.CriminalGroups; }
    }

    public CollectionFilter<CriminalGroup> FilteredCriminalGroups
    {
        get => _filteredCriminalGroups;
        set
        {
            _filteredCriminalGroups = value;
            OnPropertyChanged(nameof(FilteredCriminalGroups));
        }
    }

    public string GroupNameFilterText
    {
        get => _groupNameFilterText;
        set
        {
            _groupNameFilterText = value;
            OnPropertyChanged(nameof(GroupNameFilterText));
            FilterCommand.Execute(null);
        }
    }

    public ICommand AddCrimeGroup { get; private set; }
}

```

					122 – КР.2023.01.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public ICommand ViewDetailsCommand { get; private set; }
public ICommand FilterCommand { get; private set; }

public CriminalGroupsViewModel()
{
    _filteredCriminalGroups = new CollectionFilter<CriminalGroup>(CriminalGroups);

    AddCrimeGroup = new RelayCommand(AddCriminalGroup);
    ViewDetailsCommand = new RelayCommand(ViewDetails);
    FilterCommand = new RelayCommand(Filter);
}

private void ViewDetails(object parameter)
{
    if(parameter is CriminalGroup criminalGroup)
    {
        CriminalGroup_Information criminalGroupView = new
CriminalGroup_Information(criminalGroup);

        criminalGroupView.Show();
    }
}

private void AddCriminalGroup(object parameter)
{
    AddNewCriminalGroupViewModel targetView = new AddNewCriminalGroupViewModel();
    AddNewCriminalGroup addNewCriminalGroupWindow = new AddNewCriminalGroup();
    addNewCriminalGroupWindow.DataContext = targetView;

    addNewCriminalGroupWindow.ShowDialog();

    if (targetView.IsResult)
        RepositoryService.Instance.AddCriminalGroup(targetView.NewCriminalGroup);
}

private void Filter(object parameter)
{
    _filteredCriminalGroups.AddFilter(criminalGroup =>
criminalGroup.Name.Contains(GroupNameFilterText));

    OnPropertyChanged(nameof(GroupNameFilterText));
}

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

Клас AddNewCriminalViewModel.cs

```

public class AddNewCriminalViewModel : INotifyPropertyChanged
{
    private Criminal _newCriminal;
    private string _selectedGender;
    private DateTime _birthday;
    private bool _isResult;

    public event PropertyChangedEventHandler? PropertyChanged;

    public Criminal NewCriminal
    {
        get { return _newCriminal; }
        set
        {

```

```

        _newCriminal = value;
        OnPropertyChanged(nameof(NewCriminal));
    }
}

public DateTime SelectedBirthDay
{
    get { return _birthday; }
    set
    {
        _birthday = value;
        OnPropertyChanged(nameof(SelectedBirthDay));
    }
}

public string SelectedGender
{
    get { return _selectedGender; }
    set
    {
        _selectedGender = value;
        OnPropertyChanged(nameof(SelectedGender));
    }
}

public bool IsResult
{
    get => _isResult;
}

public ICommand ConfirmCommand { get; private set; }
public ICommand OpenImageCommand { get; private set; }

public AddNewCriminalViewModel()
{
    SelectedBirthDay = DateTime.Now;
    _isResult = false;

    NewCriminal = new Criminal();
    ConfirmCommand = new RelayCommand(Confirm);
    OpenImageCommand = new RelayCommand(OpenImage);
}

private void Confirm(object parameter)
{
    if (IsValidModel(NewCriminal))
    {
        NewCriminal.Gender = SelectedGender;
        NewCriminal.AccessLevel = 'D';
        _isResult = true;
    }
    else
    {
        MessageBox.Show("Not all data is valid!");
    }
}

private void OpenImage(object obj)
{
    var openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.gif;*.bmp";

    if (openFileDialog.ShowDialog() == true)
    {

```

```

        string imagePath = openFileDialog.FileName;
        string fileExtension = Path.GetExtension(imagePath);

        Guid uniqueFileID = Guid.NewGuid();
        string targetPath = Path.Combine(Environment.CurrentDirectory,
"Criminal_Images/", uniqueFileID.ToString() + fileExtension);
        File.Copy(imagePath, targetPath, true);

        NewCriminal.PhotoPath = targetPath;
    }
}

private void CloseWindow(Window window)
{
    window?.Close();
}

private bool IsModelValid(Criminal criminal)
{
    var properties = typeof(Criminal).GetProperties();

    foreach (var property in properties)
    {
        var columnName = property.Name;
        var validationResult = criminal[columnName];

        if (!string.IsNullOrEmpty(validationResult))
        {
            return false;
        }
    }
    return true;
}

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

Клас AddNewCriminalGroupViewModel.cs

```

public class AddNewCriminalGroupViewModel : INotifyPropertyChanged
{
    private ObservableCollection<string> _allCountries;
    private ObservableCollection<string> _allActivities;
    private ObservableCollection<string> _selectedActivities;
    private ObservableCollection<string> _selectedCountries;
    private CriminalGroup _newCriminalGroup;
    private bool _isResult;

    public event PropertyChangedEventHandler? PropertyChanged;

    public CriminalGroup NewCriminalGroup
    {
        get { return _newCriminalGroup; }
        set
        {
            _newCriminalGroup = value;
            OnPropertyChanged(nameof(NewCriminalGroup));
        }
    }
    public ObservableCollection<string> AllActivities
    {
        get { return _allActivities; }
    }
}

```

					122 – КР.2023.01.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        set
        {
            _allActivities = value;
            OnPropertyChanged(nameof(AllActivities));
        }
    }

    public ObservableCollection<string> AllCountries
    {
        get { return _allCountries; }
        set
        {
            _allCountries = value;
            OnPropertyChanged(nameof(AllCountries));
        }
    }

    public ObservableCollection<string> SelectedActivities
    {
        get { return _selectedActivities; }
        set
        {
            _selectedActivities = value;
            OnPropertyChanged(nameof(SelectedActivities));
        }
    }

    public ObservableCollection<string> SelectedCountries
    {
        get { return _selectedCountries; }
        set
        {
            _selectedCountries = value;
            OnPropertyChanged(nameof(SelectedCountries));
        }
    }

    public bool IsResult
    {
        get => _isResult;
    }

    public ICommand ConfirmCommand { get; private set; }
    public ICommand SelectionActivitiesChangedCommand { get; private set; }
    public ICommand SelectionCountriesChangedCommand { get; private set; }

    public AddNewCriminalGroupViewModel()
    {
        _newCriminalGroup = new CriminalGroup();
        _allActivities = new ObservableCollection<string>();
        _allCountries = LoadCountriesFromFile(Path.Combine(Environment.CurrentDirectory,
"countries.txt"));
        _isResult = false;

        _selectedActivities = new ObservableCollection<string>();
        _selectedCountries = new ObservableCollection<string>();

        _allActivities.Add("Theft");
        _allActivities.Add("Drug-related crimes");
        _allActivities.Add("Financial crimes");
        _allActivities.Add("Cybercrimes");
        _allActivities.Add("Terrorism");

        SelectionActivitiesChangedCommand = new RelayCommand(OnSelectionChanged);
    }

```



```

        SelectionCountriesChangedCommand = new
RelayCommand(OnSelectionCountriesChanged);
        ConfirmCommand = new RelayCommand(Confirm);
    }

    private void Confirm(object parameter)
    {
        NewCriminalGroup.Activities = SelectedActivities.ToList();
        NewCriminalGroup.Countries = SelectedCountries.ToList();
        _isResult = true;
    }

    private ObservableCollection<string> LoadCountriesFromFile(string filePath)
    {
        ObservableCollection<string> countries = new ObservableCollection<string>();

        try
        {
            using (StreamReader reader = new StreamReader(filePath))
            {
                string line;
                while ((line = reader.ReadLine()) != null)
                {
                    countries.Add(line);
                }
            }
        }
        catch (Exception ex)
        {
        }

        return countries;
    }

    private void OnSelectionChanged(object selectedItems)
    {
        if (selectedItems is IList items)
        {
            SelectedActivities.Clear();

            foreach (var item in items)
            {
                if (item is string str)
                {
                    SelectedActivities.Add(str);
                }
            }
        }
    }

    private void OnSelectionCountriesChanged(object selectedItems)
    {
        if (selectedItems is IList items)
        {
            SelectedCountries.Clear();

            foreach (var item in items)
            {
                if (item is string str)
                {
                    SelectedCountries.Add(str);
                }
            }
        }
    }

```

```

    }
}

protected virtual void OnPropertyChanged([CallerMemberName] string propertyName =
null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}
}

```

Клас RelayCommand.cs

```

public class RelayCommand : ICommand
{
    private readonly Action<object> execute;
    private readonly Func<bool> canExecute;

    public event EventHandler CanExecuteChanged
    {
        add { CommandManager.RequerySuggested += value; }
        remove { CommandManager.RequerySuggested -= value; }
    }

    public RelayCommand(Action<object> execute, Func<bool> canExecute = null)
    {
        this.execute = execute ?? throw new ArgumentNullException(nameof(execute));
        this.canExecute = canExecute;
    }

    public bool CanExecute(object parameter)
    {
        return canExecute == null || canExecute();
    }

    public void Execute(object parameter)
    {
        execute(parameter);
    }
}

```

Клас Criminal.cs

```

public class Criminal : IDataErrorInfo
{
    private static int _id = 1;

    public int ID { get; private set; }
    public string FullName { get; set; }
    public string Alias { get; set; }
    public DateTime DateOfBirth { get; set; }
    public string Nationality { get; set; }
    public string Gender { get; set; }
    public string HairColor { get; set; }
    public string EyeColor { get; set; }
    public double Height { get; set; }
    public double Weight { get; set; }
    public string SpecialCharacteristics { get; set; }
    public string CrimeDescription { get; set; }
    public string PhotoPath { get; set; }
    public char AccessLevel { get; set; }

    public string this[string columnName]
    {
        get
        {
            string result = null;
            switch (columnName)

```

					122 – КР.2023.01.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            case nameof(FullName):
            {
                if (string.IsNullOrEmpty(FullName))
                    result = "Full Name is required.";

                if (FullName.Length < 8 || FullName.Length > 128)
                    result = "Full Name length must be more than 8 symbols or
less than 128 symbols.";
                break;
            }
            case nameof(Alias):
            {
                if (string.IsNullOrEmpty(Alias))
                    result = "Alias is required.";

                if (Alias.Length < 2 || Alias.Length > 32)
                    result = "Full Name length must be more than 2 symbols or
less than 32 symbols.";
                break;
            }
            case nameof(Nationality):
            {
                if (string.IsNullOrEmpty(Nationality))
                    result = "Nationality is required.";
                break;
            }
        }
        return result;
    }
}

public string Error => null;

public Criminal()
{
    ID = _id++;
    FullName = String.Empty;
    Alias = String.Empty;
    PhotoPath = String.Empty;
    DateOfBirth = DateTime.Now;
}
}

```

Клас CriminalGroup.cs

```

public class CriminalGroup
{
    private static int _nextId = 1;

    #region Property
    public int ID { get; }
    public string Name { get; set; }
    public string Leader { get; set; }
    public string Description { get; set; }
    public List<string> Activities { get; set; }
    public List<string> Countries { get; set; }
    #endregion

    public CriminalGroup()
    {
        ID = _nextId++;
    }

    public CriminalGroup(string name, string leader, List<string> activities,
List<string> countries)

```

					122 – КР.2023.01.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        ID = _nextId++;
        Name = name;
        Leader = leader;
        Activities = activities;
        Countries = countries;
    }

    public CriminalGroup(string name, string leader, List<string> activities,
List<string> countries, string description)
    {
        ID = _nextId++;
        Name = name;
        Leader = leader;
        Activities = activities;
        Countries = countries;
        Description = description;
    }
}

```

Клас User.cs

```

public class User
{
    public static int _id = 1;

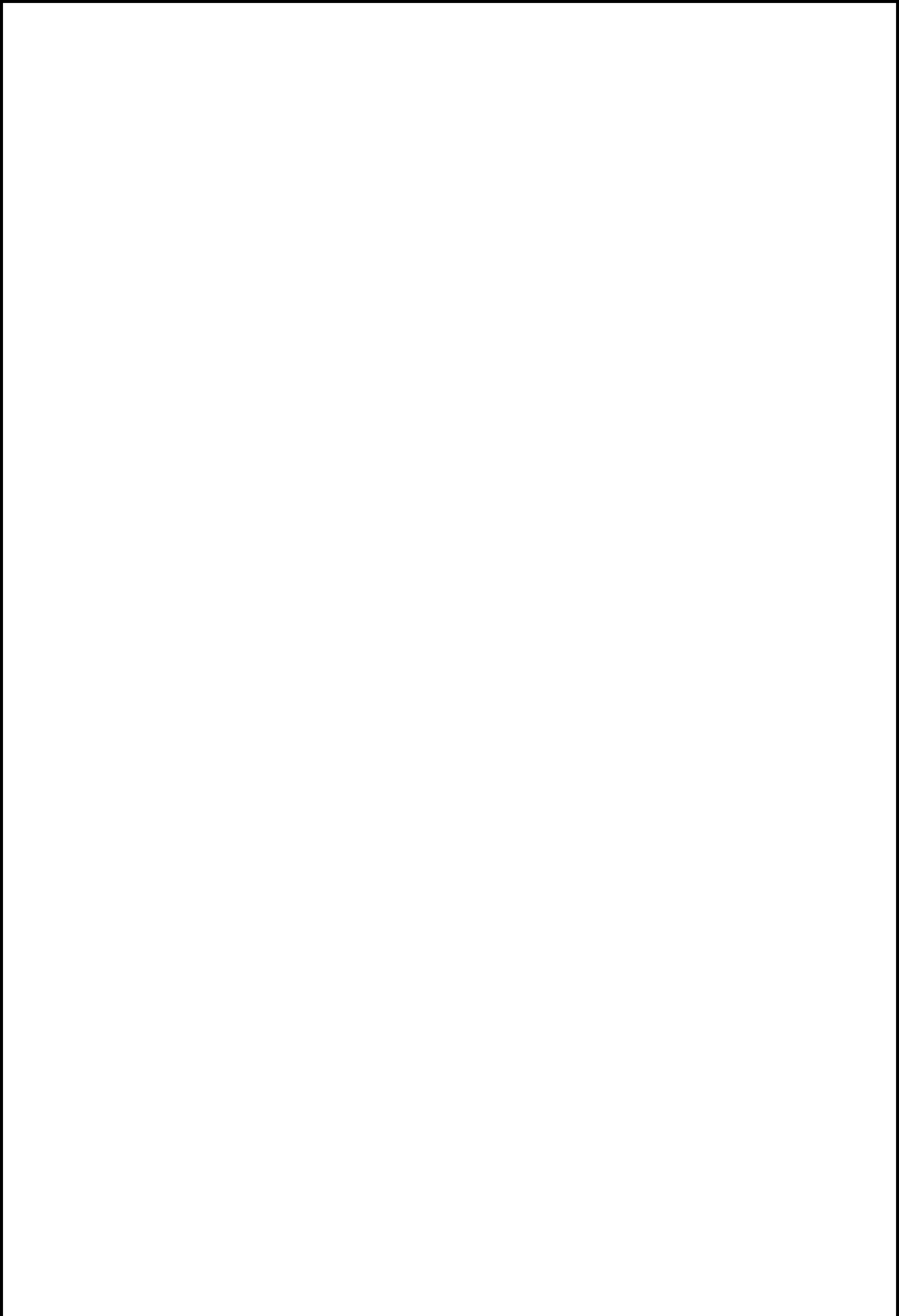
    public int ID { get; private set; }
    public string Name { get; set; }
    public string Login { get; set; }
    public string Password { get; set; }
    public char AccessLevel { get; set; }

    public User()
    {
        ID = _id++;
        Name = String.Empty;
        Login = String.Empty;
        Password = String.Empty;
        AccessLevel = 'D';
    }
}

```

Додаток Г
Тестування додатку

					122 – КР.2023.01.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		



					122 – КР.2023.01.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		