

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до курсової роботи

з дисципліни

«ОСНОВИ ПРОГРАМУВАННЯ»

на тему

«Інформаційна система альпіністського клубу»

Викладач	В.В. Хайдуров
Студентка	Ю.Д. Приймак
Група	КП-12
Варіант	19

Завдання до курсової роботи

Варіант 19. Інформаційна система альпіністського клубу

База даних хроніки сходжень в альпіністському клубі. У базі даних повинні записуватися дати початку і завершення кожного сходження, імена та адреси брали участь в ньому альпіністів, назва і висота гори, країна і район, де ця гора розташована. Дайте виразні імена таблиць і полів, в які могла б заноситися вказана інформація. Основний функціонал має складатись з процедур та функцій, які дозволили б виконати наступні дії з базою даних:

- для кожної гори показати список груп, які здійснювали сходження, в хронологічному порядку;
- надати можливість додавання нової вершини, з зазначенням назви вершини, висоти і країни розташування;
- надати можливість зміни даних про вершині, ес чи на неї не було сходження;
- показати список альпіністів, які здійснювали сходження в зазначений інтервал дат;
- надати можливість додавання нового альпініста в склад зазначеної групи;
- показати інформацію про кількості сходжень кожного альпініста на кожен гору;
- показати список сходжень (груп), які здійснювалися в зазначений користувачем період часу;
- надати можливість додавання нової групи, вказавши її назву, вершину, час початку сходження; 9) надати інформацію про те, скільки альпіністів по бували на кожній горі.

Оглавление

ВСТУП.....	4
1. Функціонування альпіністського клубу. Необхідність створення інформаційної системи роботи альпіністського клубу.....	6
2. Функціональне моделювання інформаційної системи альпіністського клубу з використанням DFD-діаграм.....	7
3. Концептуальна модель бази даних альпіністського клубу.....	8
4. Фізична модель бази даних системи альпіністського клубу.....	11
5. Опис реалізації завдань відповідно до обраного варіанту.....	13
5.1. Створення та структура таблиць бази даних інформаційної системи альпіністського клубу 13	
5.2. Формування запитів до створеної бази даних системи альпіністського клубу.....	14
5.3. Розроблення інтерфейсної частини для роботи зі створеною базою даних альпіністського клубу.....	14
ВИСНОВОК	18
ДОДАТОК А - Програмна реалізація інтерфейсної частини інформаційної системи бібліотеки та її тестування	19
ДОДАТОК Б - Основні характеристики ПК, на якому розроблена та протестована інформаційна система.....	54

ВСТУП

Актуальність досліджень. Інформаційні технології впорядковують потоки інформації на глобальному, регіональному та локальному рівнях. Вони відіграють ключову роль у формуванні техноструктури у активному впровадженні в усіх сферах життя (освіта, професійна діяльність, позаробочі хобі та громадська діяльність). Вочевидь, сучасна система альпіністського клубу має бути забезпечена спеціальним програмним забезпеченням, адже користувачу має бути зручно переглядати специфічну інформацію, використовуючи зрозумілий та приємним інтерфейсом.

Мета роботи. Метою роботи є створення багатофункціональної інформаційної системи альпіністського клубу, яка має користувацький інтерфейс та спрямована на виконання основних функцій для формування та обробки даних, необхідних для членів клубу.

Завдання дослідження. Основним завданням дослідження створення інформаційної системи, за допомогою якої можна: редагувати дані про альпіністів, груп та вершини; переглядати сходження, що відбулися чи відбудуться в зазначений період. Створена інформаційна система повинна передбачати роботу з базою даних. Повинна бути створена низка запитів, яка відображає коректність роботи такої інформаційної системи альпіністського клубу.

Порядок виконання завдання. Робота є структурованою за розділами. Перед проектуванням інформаційної системи альпіністського клубу проведено аналіз відповідної предметної області. Визначивши основні аспекти роботи розглянутих закладів, визначено основні сутності, які використовуються при створенні відповідної бази даних обліку роботи альпіністського клубу. Проведено аналіз сутностей та визначено між ними логічні зв'язки для повноцінної роботи бази даних, яка є основним структурним і функціональним елементом інформаційної системи в цілому. База даних наповнена відповідними даними, які відображають динаміку користування інформаційною системою альпіністського клубу, зокрема, база містить дані про групу, що здійснює

сходження, час початку та кінця сходження тощо. Формування таких даних відображено у кількох таблицях. Будь-які дві зв'язані таблиці мають зв'язок «один до багатьох». Наступним кроком є створення різних типів запитів до даної бази, які відображають її повноцінну роботу. Після створення бази даних створюється інтерфейс, за допомогою якого будь-який користувач має можливість виконувати обробку даних (каскадні зміни у базі, а також редагування даних «на ходу»). Отже, створена база даних та графічний інтерфейс, що призначений для неї і є основними функціональними елементами інформаційної системи.

Засоби реалізації завдань проектування. У роботі використано два основних засоби для розробки інформаційної системи альпіністського клубу. Це Microsoft SQL Server Management Studio 18, за допомогою якого виконується розробка відповідної бази інформаційної системи довідкової служби, а також, за допомогою системи створення клієнтських засобів(WPF) у Microsoft Visual Studio 2022 створено програмний інтерфейс мовою програмування C#, який дає змогу максимально спросити роботу користувача даною інформаційною системою.

1. Функціонування альпіністського клубу. Необхідність створення інформаційної системи роботи альпіністського клубу.

Будь-яка інформаційна система завжди працює з різними даними, які повинна ефективно обробляти. До обробки даних можна віднести додавання даних, видалення, оновлення даних, створення запитів звітів тощо. При цьому створення такої інформаційної системи передбачає навиків та умінь проектування бази даних та зручного й інтуїтивного інтерфейсу користувача, через який користувач інформаційної системи взаємодіє із базою даних. Завданням курсової роботи є створення інформаційної системи альпіністського клубу. Виходячи з аналізу предметної області слід зазначити, що розроблювана інформаційна система повинна мати гнучкий функціонал, зокрема повинна мати можливість:

- для кожної гори показати список груп, які здійснювали сходження, в хронологічному порядку;
- надати можливість додавання нової вершини, з зазначенням назви вершини, висоти і країни розташування;
- надати можливість зміни даних про вершині, якщо на неї не було сходження;
- показати список альпіністів, які здійснювали сходження в зазначений інтервал дат;
- надати можливість додавання нового альпініста в склад зазначеної групи;
- показати інформацію про кількість сходжень кожного альпініста на кожну гору;
- показати список сходжень (груп), які здійснювалися в зазначений користувачем період часу;
- надати можливість додавання нової групи, вказавши її назву, вершину, час початку сходження; 9) надати інформацію про те, скільки альпіністів побували на кожній горі.

2. Функціональне моделювання інформаційної системи альпіністського клубу з використанням DFD-діаграм

На рис. 2.1 показано діаграму потоків даних при роботі з даними, які містяться у базі даних інформаційної системи альпіністського клубу.

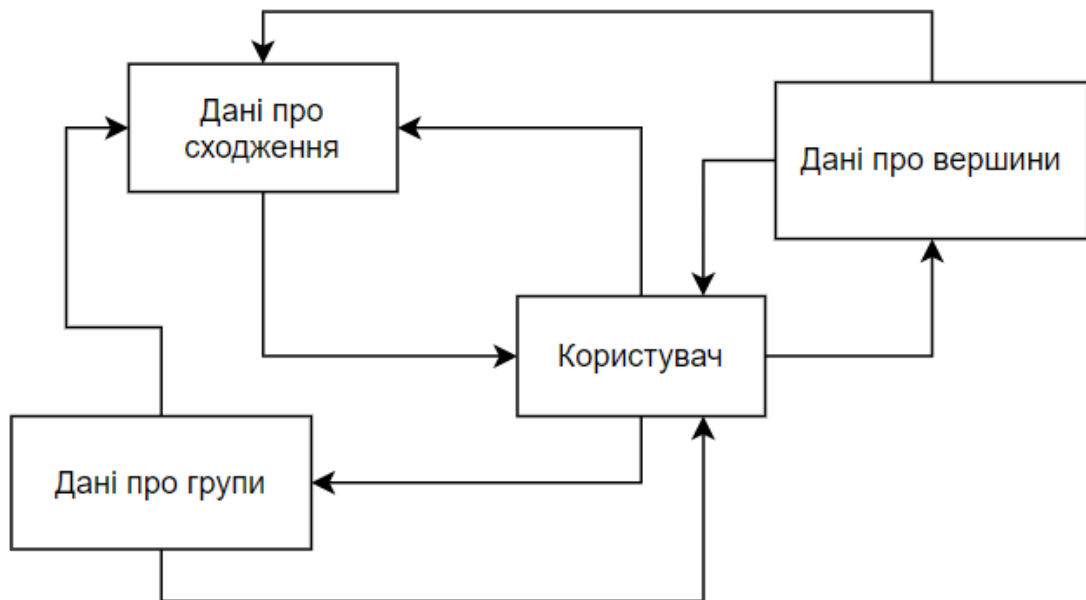


Рис.2.1. Діаграма потоків даних інформаційної системи альпіністського клубу

3. Концептуальна модель бази даних альпіністського клубу

На рис. 3.1 показана ER-діаграма, яка використана при проектуванні бази даних інформаційної системи альпіністського клубу.

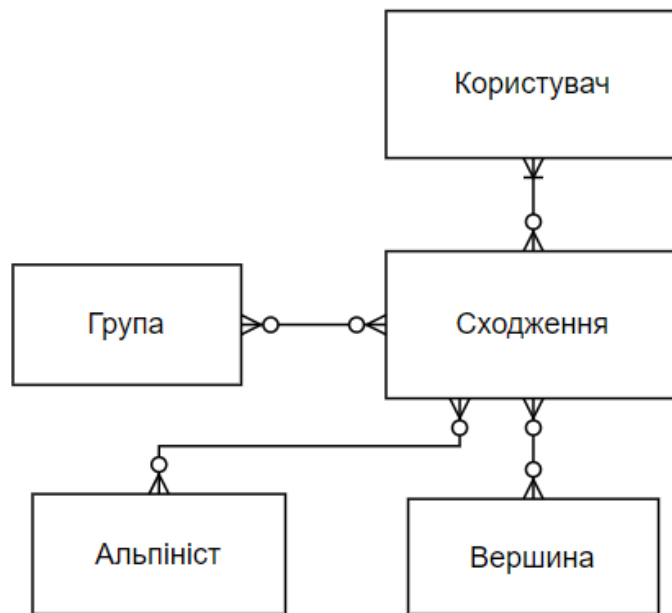


Рис. 3.1. ER-діаграма бази даних інформаційної системи альпіністського клубу

На рис. 3.2 показано концептуальну модель бази даних, що розробляється у даній курсовій роботі. На рис. 3.3 показана структура бази даних системи альпіністського клубу в середовищі Microsoft SQL Server Management Studio 18. На рис. 3.4 – діаграма розгортання. На рис. 3.5 наведена логічна модель бази даних системи альпіністського клубу.

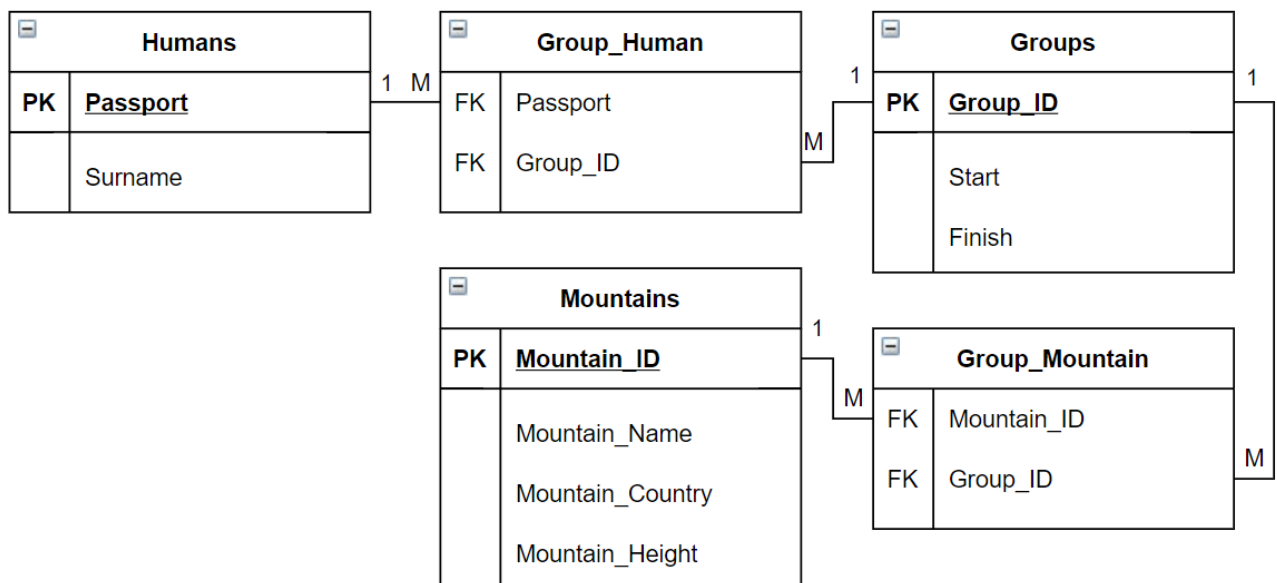


Рис. 3.2. Концептуальна модель бази даних інформаційної системи альпіністського клубу

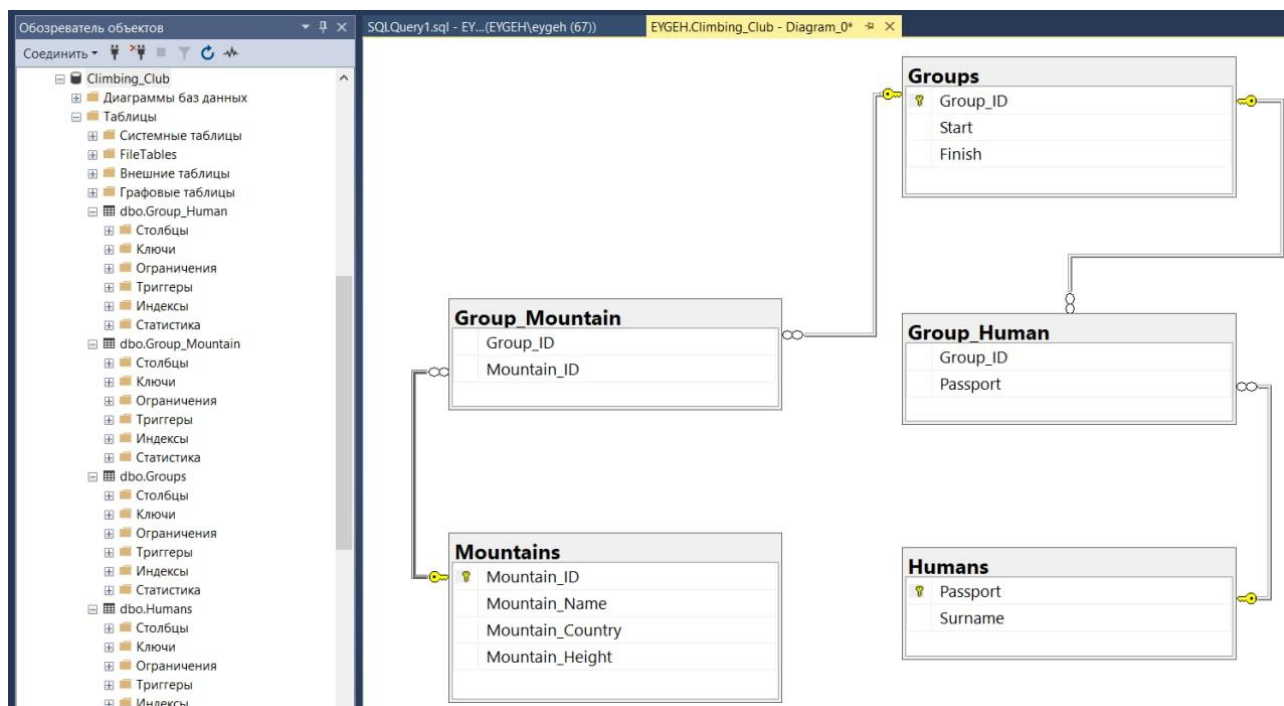


Рис. 3.3. Структура бази даних системи альпіністського клубу в середовищі Microsoft SQL Server Management Studio 18

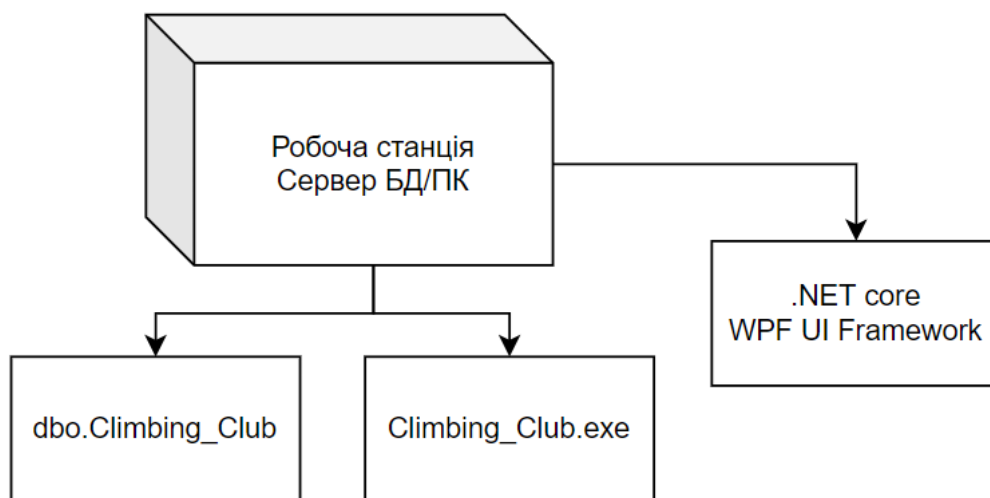


Рис. 3.4. Діаграма розгортання на апаратних засобах

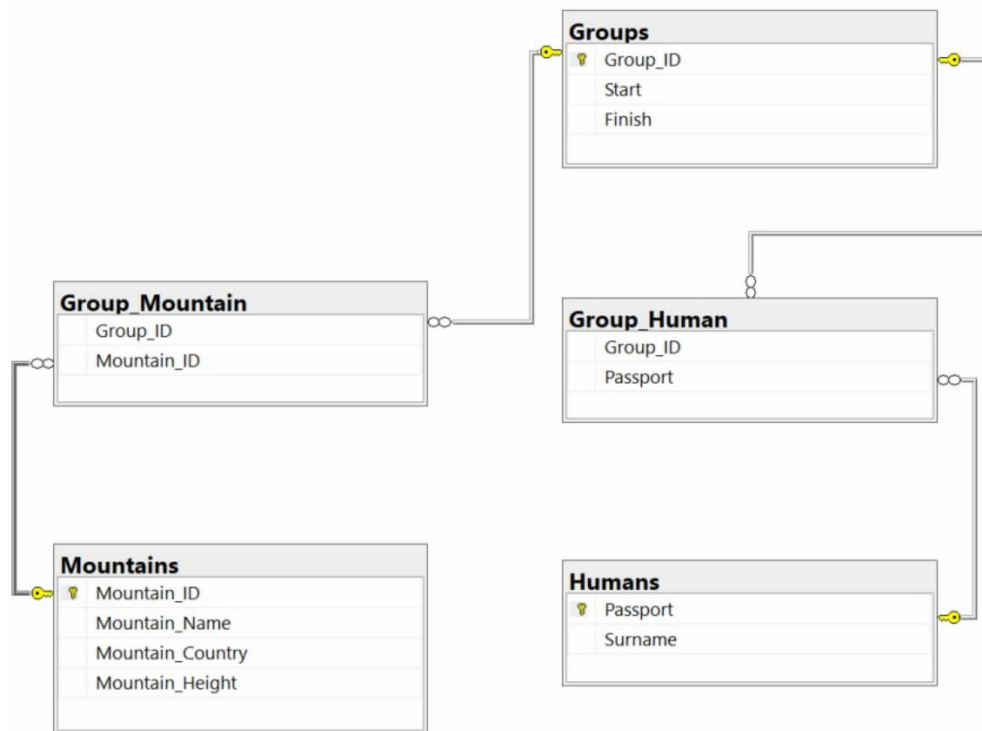


Рис. 3.4. Логічна модель бази даних системи альпіністського клубу

4. Фізична модель бази даних системи альпіністського клубу

Фізична модель бази даних - це модель даних, яка визначає, яким чином надаються дані, і містить всі деталі, необхідні СУБД для створення бази даних.

Створення таблиці «Mountains»

```
CREATE TABLE [dbo].[Mountains] (  
    [Mountain_ID] [int] NOT NULL,  
    [Mountain_Name] [varchar](20) NOT NULL,  
    [Mountain_Country] [varchar](20) NOT NULL,  
    [Mountain_Height] [int] NOT NULL,  
    CONSTRAINT [PK_Mountains] PRIMARY KEY CLUSTERED)
```

Створення таблиці «Humans»

```
CREATE TABLE [dbo].[Humans] (  
    [Passport] [int] NOT NULL,  
    [Surname] [varchar](20) NOT NULL,  
    CONSTRAINT [PK_Humans] PRIMARY KEY CLUSTERED)
```

Створення таблиці «Groups»

```
CREATE TABLE [dbo].[Groups] (  
    [Group_ID] [int] NOT NULL,  
    [Start] [smalldatetime] NOT NULL,  
    [Finish] [smalldatetime] NOT NULL,  
    CONSTRAINT [PK_Groups] PRIMARY KEY CLUSTERED)
```

Створення таблиці «Group_Human»

```
CREATE TABLE [dbo].[Group_Human] (  
    [Group_ID] [int] NOT NULL,  
    [Passport] [int] NOT NULL  
) ON [PRIMARY]
```

Створення таблиці «Group_Mountain»

```
CREATE TABLE [dbo].[Group_Mountain] (  
    [Group_ID] [int] NOT NULL,  
    [Mountain_ID] [int] NOT NULL  
) ON [PRIMARY]
```

Створена фізична модель показує зв'язки між таблицями вже спроектованої бази даних інформаційної системи альпіністського клубу.

5. Опис реалізації завдань відповідно до обраного варіанту

5.1. Створення та структура таблиць бази даних інформаційної системи альпіністського клубу

Структури таблиць розробленої бази даних інформаційної системи альпіністського клубу показані у табл. 5.1.1–5.1.5.

Таблиця 5.1.1

Структура таблиці «Mountains»

Назва стовпця	Тип	Чи може бути пустим
Mountain_ID	int	NOT NULL
Mountain_Name	varchar(50)	
Mountain_Country	varchar(50)	
Mountain_Height	int	

Таблиця 5.1.2

Структура таблиці «Groups»

Назва стовпця	Тип	Чи може бути пустим
Group_ID	int	NOT NULL
Start	smalldatetime	
Finish	smalldatetime	

Таблиця 5.1.3

Структура таблиці «Humans»

Назва стовпця	Тип	Чи може бути пустим
Passport	int	NOT NULL
Surname	varchar(50)	

Таблиця 5.1.4

Структура таблиці «Group_Mountain»

Назва стовпця	Тип	Чи може бути пустим
Mountain_ID	int	NOT NULL
Group_ID	int	

Таблиця 5.1.5

Структура таблиці «Group_Human»

Назва стовпця	Тип	Чи може бути пустим
Group_ID	int	NOT NULL
Passport	int	

Створені таблиці повністю відображають структуру бази даних, а також повністю ураховують весь необхідний функціонал при роботі з самою базою даних інформаційної системи.

5.2. Формування запитів до створеної бази даних системи альпіністського клубу

Нижче наведено основні звіти, які було створено для демонстрації роботи бази даних інформаційної системи альпіністського клубу. Усі запити, виконані в інтерфейсі, та результати їх роботи наведено у додатку А до даної курсової роботи з урахування використання інтерфейсу користувача, за допомогою якого деякі запити керуються параметрами, які вводить користувач бази даних розробленої інформаційної системи альпіністського клубу.

Запит №1. Додавання нової групи до бази даних

```
INSERT INTO Groups VALUES (6, '01.01.2019', '03.01.2019')
```

Запит №2.

```
SELECT dbo.Groups.Group_ID, dbo.Mountains.Mountain_Name,  
dbo.Groups.Start, dbo.Groups.Finish FROM dbo.Groups INNER JOIN  
dbo.Group_Mountain ON dbo.Groups.Group_ID =  
dbo.Group_Mountain.Group_ID INNER JOIN dbo.Mountains ON  
dbo.Group_Mountain.Mountain_ID = dbo.Mountains.Mountain_ID WHERE  
(dbo.Groups.Start BETWEEN ('01.01.2000') AND ('01.01.2023')) OR  
(dbo.Groups.Finish BETWEEN ('01.01.2000') AND ('01.01.2023'))
```

	Group_ID	Mountain_Name	Start	Finish
1	1	Говерла	2022-07-14 05:12:00	2022-08-14 02:12:00
2	2	Еверест	2022-03-19 19:12:00	2022-03-21 08:42:00
3	3	Говерла	2022-11-25 11:00:00	2022-11-25 22:05:00
4	4	Говерла	2022-01-01 00:00:00	2022-01-02 00:00:00
5	5	Айпетрі	2021-01-01 00:00:00	2021-01-02 00:00:00

Рис. 5.2.1 Реалізація запиту 2

5.3. Розроблення інтерфейсної частини для роботи зі створеною базою даних альпіністського клубу

Інтерфейсна частина інформаційної системи для створеної бази даних складається з кількох вікон, з якими працює користувач даної інформаційної системи. З головно вікна користувач має можливість перейти до трьох інших відповідних вікон. Усе це забезпечується цілісністю даних, яка забезпечується відповідними зв'язками у базі, що була спроектована.

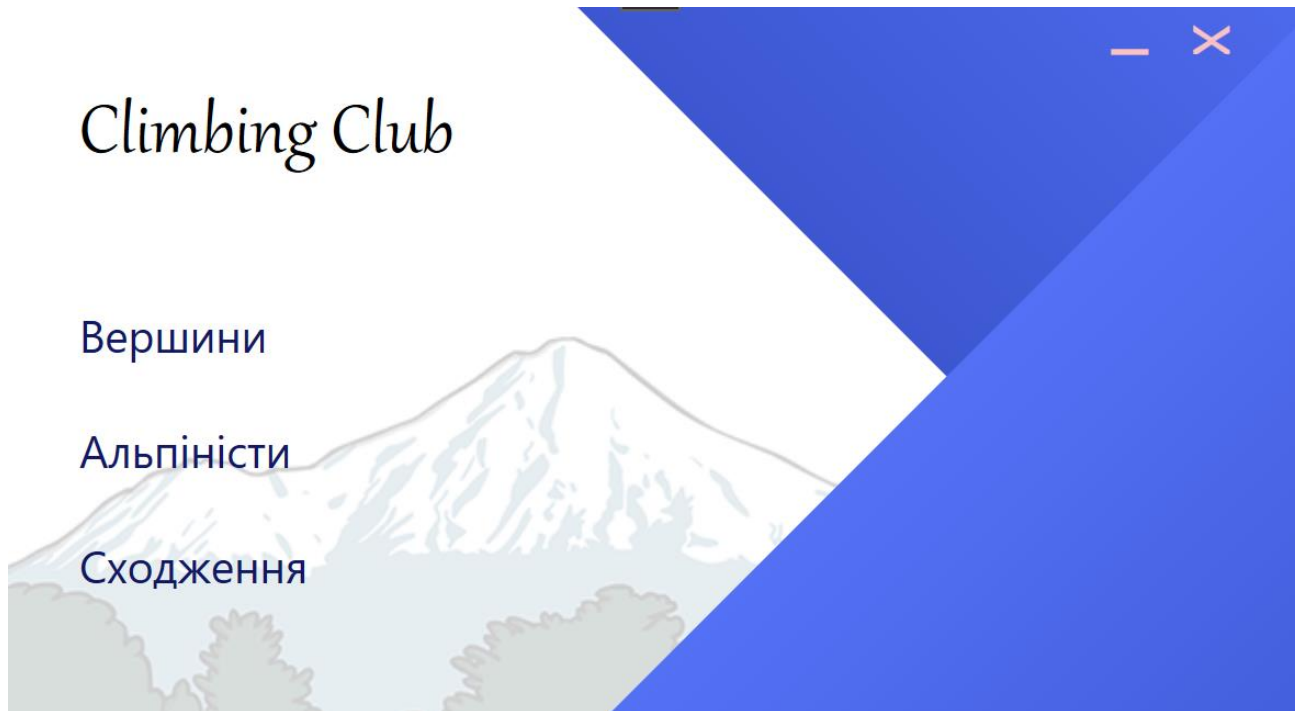


Рис 5.3.1 Головне вікно інформаційної альпіністського клубу

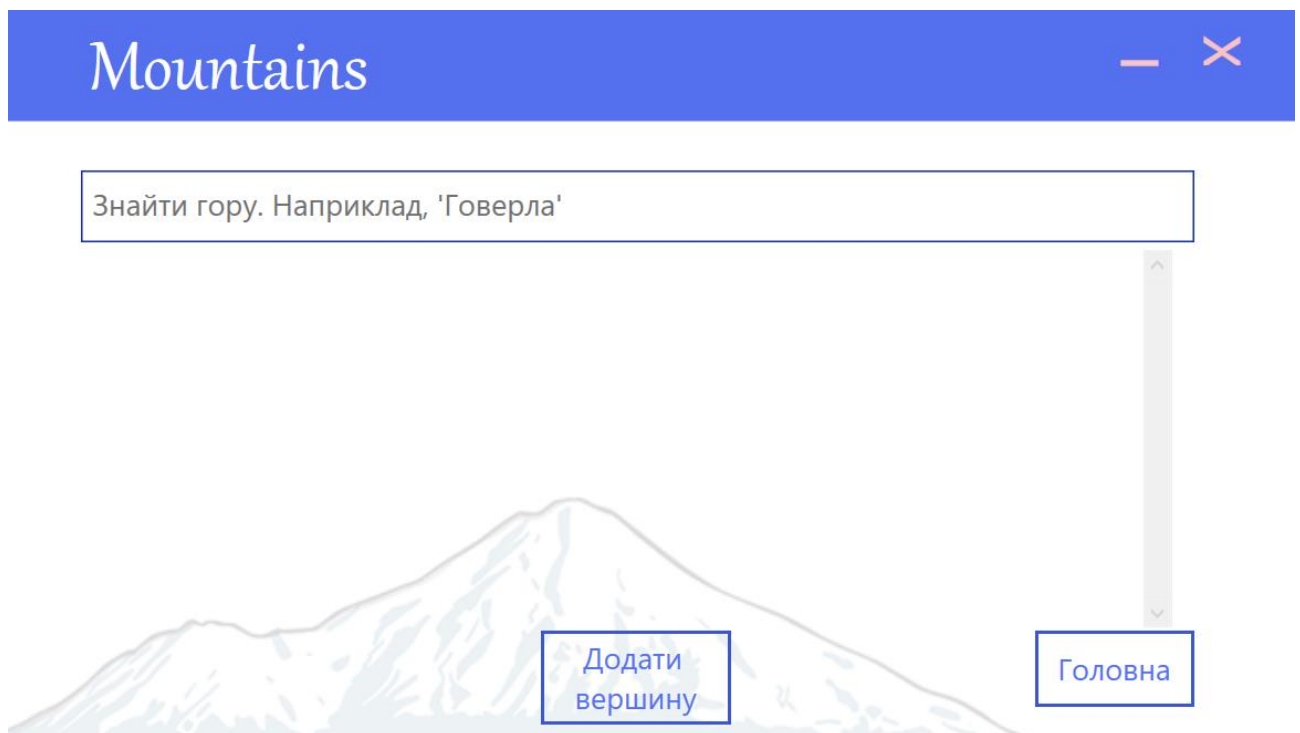


Рис 5.3.2 Вікно перегляду інформації про конкретну вершину

Додати вершину

Правильно введіть дані нової вершини,
щоб додати її.

Рис 5.3.3 Вікно додавання нової вершини до бази даних

Climbers

— ×

Введіть час підйому та спуску: -

Додати альпініста

Головна

Рис 5.3.4 Вікно перегляду інформації про альпіністів, що здійснювали або здійснять сходження у зазначений період

Додати альпініста

Рис 5.3.5 Вікно додавання нової альпініста

Рис 5.3.6 Вікно додавання інформації на сходження

Рис 5.3.7 Вікно перегляду інформації про сходження (альпіністи, вершина, час)

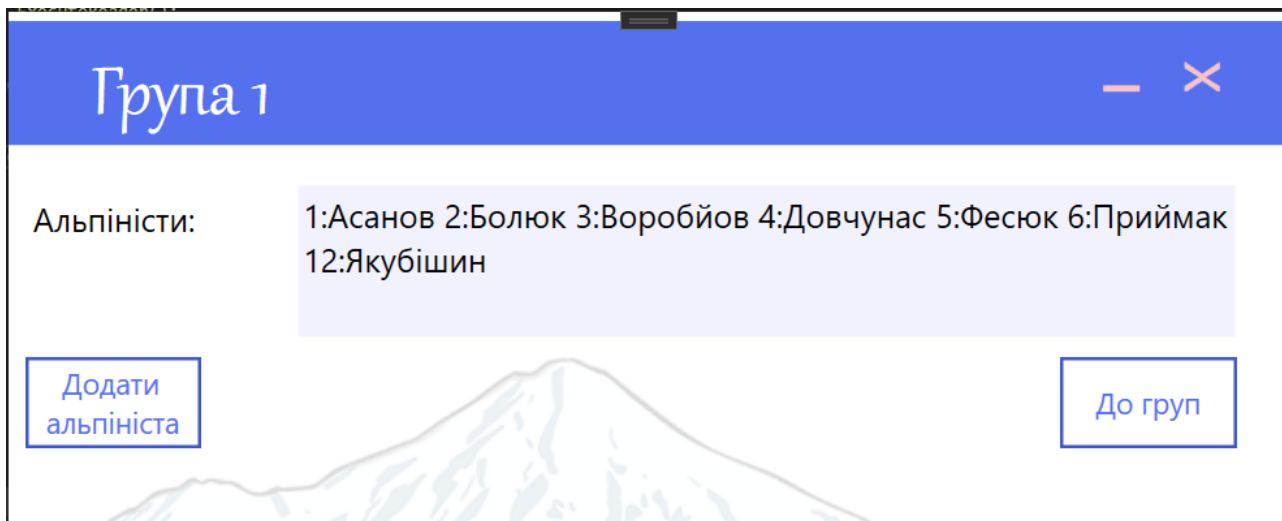


Рис 5.3.8 Вікно перегляду складу групи

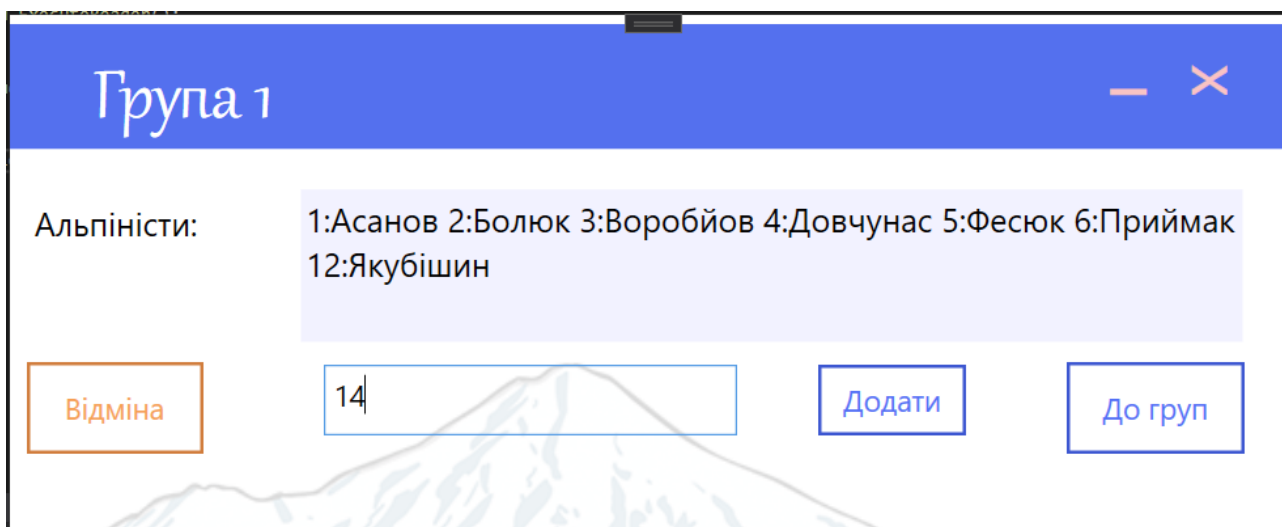


Рис 5.3.9 Вікно додавання альпіністів до складу групи

ВИСНОВОК

Курсова робота на тему «Інформаційна система альпіністського клубу» має виключно практичний зміст. Програмно реалізовано базу даних, які містить основну інформацію про альпіністів, вершини, сходження та інше. До створеної бази даних розроблено зручний динамічний інтерфейс користувача, за допомогою якого користувач може редагувати дані безпосередньо у WPF-додатку, який програмно реалізовано мовою програмування C# у середовищі Visual Studio 2019.

Для перевірки коректності роботи бази даних, а також проектованої її структури створено низка запитів, представлень та процедур. Для коректності роботи бази даних, а також для цілісності даних створені відповідні тригери, які

також забезпечують повноту функціонування бази даних створеної інформаційної системи альпіністського клубу.

Усі програмні реалізації мовою програмування C# (WPF) виведено окремо у додатки до даної курсової роботи. Усі модулі протестовані та працюють коректно.

ДОДАТОК А - Програмна реалізація інтерфейсної частини інформаційної системи бібліотеки та її тестування

Інтерфейсна XML-реалізація WPF вікон розробленої інформаційної системи.

```
<Window x:Class="Climbing_Club.MainWindow"
        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:Climbing_Club"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800"
        WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
```

```

        <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.2" Margin="-76,155,0,-198"></Image>

        <Border Style="{StaticResource TopBorderGradient}" Margin="365,-
112,-365,112" Grid.RowSpan="2"/>
        <Border Style="{StaticResource BottomBorderGradient}"
Margin="365,132,-365,-132" Grid.RowSpan="2"/>

        <Label Content="Climbing Club" HorizontalAlignment="Left"
Height="76" Margin="41,35,0,0" VerticalAlignment="Top" Width="316"
FontSize="50" FontFamily="Gabriola" FontWeight="Normal"/>

        <Label MouseLeftButtonDown="Mounatins_Click"
Style="{StaticResource MW_Button}" Margin="41,180,0,0" Content="Вершини"
Width="128"/>
        <Label MouseLeftButtonDown="Climbers_Click"
Style="{StaticResource MW_Button}" Margin="41,250,0,0"
Content="Альпіністи" Width="138"/>
        <Label MouseLeftButtonDown="Climbings_Click"
Style="{StaticResource MW_Button}" Margin="41,320,0,0"
Content="Сходження"/>
        <Label MouseLeftButtonDown="Close_Click" Style="{StaticResource
Close_Btn}" Content="X"/>
        <Label MouseLeftButtonDown="Minimazed_Click"
Style="{StaticResource Minimazed_Btn}" Content="1" />
    </Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.AddClimber"
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:Climbing_Club"
mc:Ignorable="d"
Title="AddClimber" Height="233" Width="595"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Label Content="Додати альпініста" HorizontalAlignment="Center"
Height="45" Margin="0,10,0,0" VerticalAlignment="Top" Width="220"
FontSize="25"/>

        <Label Name="Placeholder_Surname" Padding="7,7,0,0"
Content="Прізвище" HorizontalAlignment="Center" Height="42"
Margin="0,74,0,0" VerticalAlignment="Top" Width="318" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="Surname_TB" TextChanged="SurnameChanged"
Padding="4,6,0,0" HorizontalAlignment="Center" Height="42"
Margin="0,74,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="318" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>
        <Button BorderThickness="2" Content="Альпіністи"
HorizontalAlignment="Left" Margin="91,161,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"

```

```

Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMountain_Click"/>
    <Button x:Name="AddClimber_Btn" BorderThickness="2"
Content="Додати" HorizontalAlignment="Left" Margin="409,161,0,0"
VerticalAlignment="Top" Height="46" Width="96" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="AddClimber_Btn_Click" Visibility="Hidden"/>
    <TextBlock x:Name="IncorrectInput_TB" HorizontalAlignment="Left"
Height="42" Margin="356,165,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="229" Foreground="#FFFF7979"><Run
Text="Правильно введіть дані "/><Run Text="нової"/><Run Text=" вершини,
щоб "/><Run Text="додати "/><Run Language="uk-ua"
Text="ii."/></TextBlock>

</Grid>
</Window>

<Window x:Class="Climbing_Club.AddClimbing"
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:Climbing_Club"
mc:Ignorable="d"
Title="AddClimbing" Height="233" Width="595"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Label Content="Додати групи" HorizontalAlignment="Center"
Height="45" Margin="0,10,0,0" VerticalAlignment="Top" Width="162"
FontSize="25"/>
        <Label Name="Placeholder_Mountain" Padding="7,7,0,0"
Content="Назва гори" HorizontalAlignment="Left" Height="42"
Margin="29,55,0,0" VerticalAlignment="Top" Width="253" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="Mountain_TB"
TextChanged="Mountain_TB_TextChanged" Padding="4,6,0,0"
HorizontalAlignment="Left" Height="42" Margin="29,55,0,0"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="253"
BorderBrush="#FF122BA3" Background="#145458FF" FontSize="20"/>
        <Button BorderThickness="2" Content="Групи"
HorizontalAlignment="Left" Margin="29,165,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToGroup_Click"/>
        <Label x:Name="Placeholder_Start" Padding="7,7,0,0" Content="Час
початку сходження" HorizontalAlignment="Left" Height="42"
Margin="29,114,0,0" VerticalAlignment="Top" Width="265" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="Start_TB" TextChanged="Start_TB_TextChanged"
Padding="4,6,0,0" HorizontalAlignment="Left" Height="42"
Margin="28,114,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="265" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>
        <Label x:Name="Placeholder_Finish" Padding="7,7,0,0" Content="Час
завершення сходження" HorizontalAlignment="Left" Height="42"

```

```

Margin="298,114,0,0" VerticalAlignment="Top" Width="268" FontSize="20"
Foreground="#FF707070"/>
    <TextBox x:Name="Finish_TB" TextChanged="Finish_TB_TextChanged"
Padding="4,6,0,0" HorizontalAlignment="Left" Height="42"
Margin="298,114,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="268" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>
    <TextBlock x:Name="IncorrectInput_TB" HorizontalAlignment="Left"
Height="42" Margin="356,165,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="229" Foreground="#FFFF7979"><Run
Text="Правильно введіть дані "/><Run Text="нової"/><Run Text=" вершини,
щоб "/><Run Text="додати "/><Run Language="uk-ua"
Text="ii."/></TextBlock>
    <Button x:Name="AddClimbing_Btn" BorderThickness="2"
Content="Додати" HorizontalAlignment="Left" Margin="409,161,0,0"
VerticalAlignment="Top" Height="46" Width="96" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="AddClimbing_Btn_Click" Visibility="Hidden"/>

</Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.AddMountain"
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:Climbing_Club"
mc:Ignorable="d"
Title="AddMountain" Height="233" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Label Content="Додати вершину" HorizontalAlignment="Center"
Height="45" Margin="0,10,0,0" VerticalAlignment="Top" Width="204"
FontSize="25"/>
        <Label Name="Placeholder_Name" Padding="7,7,0,0" Content="Назва
гори" HorizontalAlignment="Left" Height="42" Margin="28,74,0,0"
VerticalAlignment="Top" Width="244" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="MountainName_TB" TextChanged="InfoChanged"
Padding="4,6,0,0" HorizontalAlignment="Left" Height="42"
Margin="28,74,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="244" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>

        <Label Name="Placeholder_Height" Padding="7,7,0,0"
Content="Висота гори" HorizontalAlignment="Left" Height="42"
Margin="555,74,0,0" VerticalAlignment="Top" Width="212" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="MountainHeight_TB" TextChanged="InfoChanged"
Padding="4,6,0,0" HorizontalAlignment="Left" Height="42"
Margin="555,74,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="212" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>
    </Grid>
</Window>

```

```

        <Label Name="Placeholder_Country" Padding="7,7,0,0"
Content="Країна розташування" HorizontalAlignment="Left" Height="42"
Margin="298,74,0,0" VerticalAlignment="Top" Width="212" FontSize="20"
Foreground="#FF707070"/>
        <TextBox x:Name="MountainCountry_TB" TextChanged="InfoChanged"
Padding="4,6,0,0" HorizontalAlignment="Left" Height="42"
Margin="298,74,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="212" BorderBrush="#FF122BA3" Background="#145458FF"
FontSize="20"/>
        <Button BorderThickness="2" Content="Вершини"
HorizontalAlignment="Left" Margin="127,163,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMountain_Click"/>
        <Button Name="AddMountain_Btn" Visibility="Hidden"
BorderThickness="2" Content="Додати" HorizontalAlignment="Left"
Margin="517,163,0,0" VerticalAlignment="Top" Height="46" Width="96"
Background="{x:Null}" BorderBrush="#FF3D57D1" Foreground="#FF5471F6"
FontSize="18" FontWeight="Normal" Click="AddMountain_Click"/>
        <TextBlock x:Name="IncorrectInput_TB" HorizontalAlignment="Left"
Height="42" Margin="460,165,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="229" Foreground="#FFFF7979"><Run
Text="Правильно введіть дані "/><Run Text="нової"/><Run Text=" вершини,
щоб "/><Run Text="додати "/><Run Language="uk-ua"
Text="її."/></TextBlock>

    </Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.ClimberAbout"
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:Climbing_Club"
mc:Ignorable="d"
Title="ClimberAbout" Height="324" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Grid x:Name="Header" Margin="0,0,0,235">
            <Border Background="#FF5470EE" HorizontalAlignment="Center"
Height="89" VerticalAlignment="Center" Width="800"/>
            <Label MouseLeftButtonDown="Close_Click"
Style="{StaticResource Close_Btn}" Content="X" Margin="758,19,0,0"/>
            <Label MouseLeftButtonDown="Minimized_Click"
Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
            <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.15" Margin="-76,151,0,-568"/>
            <Label Name="Surname_LB" Content="Прізвище"
HorizontalAlignment="Left" Height="76" Margin="46 ,0,0,0"
VerticalAlignment="Top" Width="316" FontSize="50" FontFamily="Gabriola"
FontWeight="Normal" Foreground="White"/>
        </Grid>
        <Grid Name="Content" Margin="0,89,0,0">

```

```

        <Label Content="Сходження:" HorizontalAlignment="Left"
Height="40" Margin="10,30,0,0" VerticalAlignment="Top" Width="145"
FontSize="20"/>
        <ScrollViewer Margin="135,30,10,83">
            <StackPanel Name="Mountains">

                </StackPanel>
            </ScrollViewer>
            <Button BorderThickness="2" Content="До
&#xD;&#xA;альпіністів" HorizontalAlignment="Left" Margin="638,157,0,0"
VerticalAlignment="Top" Height="56" Width="109" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="Climbings_Click"/>
        </Grid>
    </Grid>
</Window>

<Window x:Class="Climbing_Club.Climbers"
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:Climbing_Club"
mc:Ignorable="d"
Title="Climbers" Height="450" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Border Background="#FF5470EE" HorizontalAlignment="Center"
Height="78" VerticalAlignment="Top" Width="800"/>
        <Label MouseLeftButtonDown="Close_Click" Style="{StaticResource
Close_Btn}" Content="X" Margin="758,19,0,0"/>
        <Label MouseLeftButtonDown="Minimized_Click"
Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
        <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.15" Margin="-76,247,0,-290"/>
        <Label Content="Climbers" HorizontalAlignment="Left" Height="76"
Margin="46,0,0,0" VerticalAlignment="Top" Width="316" FontSize="50"
FontFamily="Gabriola" FontWeight="Normal" Foreground="White"/>

        <Grid Name="Content" Margin="0,78,0,0">
            <Button BorderThickness="2" Content="Головна"
HorizontalAlignment="Left" Margin="621,305,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMain_Click"/>

            <Label Padding="7,8,0,0" Content="Введіть час підйому та
спуску:" HorizontalAlignment="Left" Height="43" Margin="30,31,0,0"
VerticalAlignment="Top" Width="266" FontSize="18"
Foreground="#FF020086"/>
            <Label Name="Placeholder_Start" Padding="7,8,0,0"
Content="01.01.2022 6:00" HorizontalAlignment="Left" Height="43"
Margin="317,31,0,0" VerticalAlignment="Top" Width="142" FontSize="18"
Foreground="#FF707070"/>
        </Grid>
    </Grid>
</Window>

```



```

        <TextBox x:Name="Start_TB" TextChanged="Time_Changed"
        Padding="3,5,0,0" HorizontalAlignment="Left" Height="43"
        Margin="317,31,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
        Width="142" BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="18"/>
        <Label Padding="7,8,0,0" Content="-"
        HorizontalAlignment="Left" Height="43" Margin="464,31,0,0"
        VerticalAlignment="Top" Width="22" FontSize="18" Foreground="#FF020086"/>
        <Label x:Name="Placeholder_Finish" Padding="7,8,0,0"
        Content="01.02.2022 6:00" HorizontalAlignment="Left" Height="43"
        Margin="491,31,0,0" VerticalAlignment="Top" Width="142" FontSize="18"
        Foreground="#FF707070"/>
        <TextBox x:Name="Finish_TB" TextChanged="Time_Changed"
        Padding="3,5,0,0" HorizontalAlignment="Left" Height="43"
        Margin="491,31,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
        Width="142" BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="18"/>
        <ScrollViewer Margin="30,79,83,72">
            <StackPanel x:Name="Surnames_SP">
                <Label Content="Прізвище" FontSize="23"
                Foreground="#FF000383" Cursor="Hand"></Label>
            </StackPanel>
        </ScrollViewer>
        <Button BorderThickness="2" Content="
Додати&#xD;&#xA;альпініста" HorizontalAlignment="Left"
        Margin="47,305,0,0" VerticalAlignment="Top" Height="51" Width="104"
        Background="{x:Null}" BorderBrush="#FF3D57D1" Foreground="#FF5471F6"
        FontSize="18" FontWeight="Normal" Click="AddClimber"/>
    </Grid>
</Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.Climbings"
        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:Climbing_Club"
        mc:Ignorable="d"
        Title="Climbings" Height="458" Width="800"
        WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Border Background="#FF5470EE" HorizontalAlignment="Center"
        Height="78" VerticalAlignment="Top" Width="800"/>
        <Label MouseLeftButtonDown="Close_Click" Style="{StaticResource
        Close_Btn}" Content="X" Margin="758,19,0,0"/>
        <Label MouseLeftButtonDown="Minimized_Click"
        Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
        <Image
        Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
        wiki_ver_1.png" Opacity="0.15" Margin="-76,247,0,-290"/>
        <Label Content="Climbings" HorizontalAlignment="Left" Height="76"
        Margin="46,0,0,0" VerticalAlignment="Top" Width="316" FontSize="50"
        FontFamily="Gabriola" FontWeight="Normal" Foreground="White"/>
    </Grid>

```

```

<Grid Name="Content" Margin="0,78,0,0">

```

```

        <Button BorderThickness="2" Content="Головна"
HorizontalAlignment="Left" Margin="621,305,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMain_Click"/>
        <Label Padding="7,8,0,0" Content="Введіть час підйому та
спуску:" HorizontalAlignment="Left" Height="43" Margin="30,31,0,0"
VerticalAlignment="Top" Width="266" FontSize="18"
Foreground="#FF020086"/>
        <Label Name="Placeholder_Start" Padding="7,8,0,0"
Content="01.01.2022 6:00" HorizontalAlignment="Left" Height="43"
Margin="317,31,0,0" VerticalAlignment="Top" Width="142" FontSize="18"
Foreground="#FF707070"/>
        <TextBox x:Name="Start_TB" TextChanged="Time_Changed"
Padding="3,5,0,0" HorizontalAlignment="Left" Height="43"
Margin="317,31,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="142" BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="18"/>
        <Label Padding="7,8,0,0" Content="-"
HorizontalAlignment="Left" Height="43" Margin="464,31,0,0"
VerticalAlignment="Top" Width="22" FontSize="18" Foreground="#FF020086"/>
        <Label x:Name="Placeholder_Finish" Padding="7,8,0,0"
Content="01.02.2022 6:00" HorizontalAlignment="Left" Height="43"
Margin="491,31,0,0" VerticalAlignment="Top" Width="142" FontSize="18"
Foreground="#FF707070"/>
        <TextBox x:Name="Finish_TB" TextChanged="Time_Changed"
Padding="3,5,0,0" HorizontalAlignment="Left" Height="43"
Margin="491,31,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="142" BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="18"/>
        <ScrollViewer Margin="30,79,83,72">
            <StackPanel x:Name="Groups_SP">
                <Label Content="Група 1 - Говерла, 01.01.2022 6:00 -
01.02.2022 6:00" FontSize="23" Foreground="#FF000383" Cursor="Hand"
MouseLeftButtonDown="GroupAbout_Click"></Label>
            </StackPanel>
        </ScrollViewer>
    </Grid>
    <Button x:Name="ToAdd_Btn" BorderThickness="2" Content="Додати
&#xA;сходження" HorizontalAlignment="Left" Margin="25,384,0,0"
VerticalAlignment="Top" Height="56" Width="108" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="ToAddClimbing_Click"/>
</Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.GroupAbout"
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:Climbing_Club"
mc:Ignorable="d"
Title="GroupAbout" Height="324" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Grid x:Name="Header" Margin="0,0,0,235">

```

```

        <Border Background="#FF5470EE" HorizontalAlignment="Center"
Height="89" VerticalAlignment="Center" Width="800"/>
        <Label MouseLeftButtonDown="Close_Click"
Style="{StaticResource Close_Btn}" Content="X" Margin="758,19,0,0"/>
        <Label MouseLeftButtonDown="Minimized_Click"
Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
        <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.15" Margin="-76,151,0,-568"/>
        <Label Name="Group_Name_LB" Content="Група 1"
HorizontalAlignment="Left" Height="76" Margin="46 ,0,0,0"
VerticalAlignment="Top" Width="316" FontSize="50" FontFamily="Gabriola"
FontWeight="Normal" Foreground="White"/>
    </Grid>
    <Grid Name="Content" Margin="0,73,0,0">
        <Label Content="Альпіністи:" HorizontalAlignment="Left"
Height="40" Margin="10,30,0,0" VerticalAlignment="Top" Width="163"
FontSize="20"/>
        <TextBlock x:Name="Climbers_TB" Padding="3,5,0,0"
HorizontalAlignment="Left" Height="93" Margin="178,28,0,0"
TextWrapping="Wrap" Text="3:Воробйов 1:Асанов 5:Приймак"
VerticalAlignment="Top" Width="576" Background="#145458FF"
FontSize="20"/>
    </Grid>
    <Button Name="ToAdd_Btn" BorderThickness="2" Content=" Додати
&#xD;&#xA;альпініста" HorizontalAlignment="Left" Margin="10,206,0,0"
VerticalAlignment="Top" Height="56" Width="108" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="ToAddClimber_Click"/>
    <Button BorderThickness="2" Content="До груп"
HorizontalAlignment="Left" Margin="646,206,0,0" VerticalAlignment="Top"
Height="56" Width="109" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="Climbings_Click"/>
    <Grid Name="AddClimber_Grid" Margin="182,198,154,10"
Visibility="Hidden">
        <Label x:Name="Placeholder_NewClimber" Padding="7,8,0,0"
Content="Паспорт (Наприклад 1)" HorizontalAlignment="Left" Height="43"
Margin="10,10,0,0" VerticalAlignment="Top" Width="253" FontSize="18"
Foreground="#FF707070"/>
        <TextBox Name="NewClData" TextChanged="ClimberToAdd_Changed"
Padding="3,5,0,0" HorizontalAlignment="Left" Height="43"
Margin="10,10,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="253" BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="18"/>
        <Button Name="AddCl_Btn" BorderThickness="2" Content="Додати"
HorizontalAlignment="Left" Margin="312,10,0,0" VerticalAlignment="Top"
Height="43" Width="91" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="AddClimber_Click" IsEnabled="False"/>
        <Label Name="InputIncorrect" Content="*Правильно введіть дані
для додавання" HorizontalAlignment="Left" Margin="10,72,0,0"
VerticalAlignment="Top" Width="393" Height="34" Foreground="#FFFF5E5E"
FontSize="16"/>
    </Grid>
</Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.Mountain_About"
        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:Climbing_Club"
        mc:Ignorable="d"
        Title="Mountain_About" Height="455" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Grid x:Name="Header" Margin="0,0,0,377">
            <Border Background="#FF5470EE" HorizontalAlignment="Center"
Height="78" VerticalAlignment="Top" Width="800"/>
            <Label MouseLeftButtonDown="Close_Click"
Style="{StaticResource Close_Btn}" Content="X" Margin="758,19,0,0"/>
            <Label MouseLeftButtonDown="Minimized_Click"
Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
            <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.15" Margin="-76,247,0,-664"/>
            <Label Name="Mountain_Name_LB" Content="Говерла"
HorizontalAlignment="Left" Height="76" Margin="46,0,0,0"
VerticalAlignment="Top" Width="316" FontSize="50" FontFamily="Gabriola"
FontWeight="Normal" Foreground="White"/>
        </Grid>
        <Grid x:Name="Content" Margin="0,78,0,0">
            <Label Content="Крайна:" HorizontalAlignment="Left"
Margin="38,25,0,0" VerticalAlignment="Top" Height="48" Width="98"
FontSize="25" Background="{x:Null}" Foreground="#FF0A2D9E"/>
            <TextBox x:Name="MountainCountry_TB"
TextChanged="InfoChanged" Padding="3,5,0,0" HorizontalAlignment="Left"
Height="42" Margin="141,28,0,0" TextWrapping="Wrap" Text="Україна"
VerticalAlignment="Top" Width="181" BorderBrush="#FF122BA3"
Background="#145458FF" FontSize="20"/>
            <Label Content="Висота:" HorizontalAlignment="Center"
Margin="0,25,0,0" VerticalAlignment="Top" Height="48" Width="98"
FontSize="25" Background="{x:Null}" Foreground="#FF0A2D9E"/>
            <TextBox x:Name="MountainHeight_TB" TextChanged="InfoChanged"
Padding="3,5,0,0" HorizontalAlignment="Left" Height="42"
Margin="449,28,0,0" TextWrapping="Wrap" Text="2061"
VerticalAlignment="Top" Width="77" BorderBrush="#FF122BA3"
Background="#145458FF" FontSize="20"/>
            <Button BorderThickness="2" Content="Вершини"
HorizontalAlignment="Left" Margin="621,305,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMountain_Click"/>
            <ScrollViewer Margin="38,131,83,77">
                <StackPanel x:Name="Climbings_SP">
                    <Label Content="Триває 1: Час: 24.03.22; 12:00 -
24.03.22; 12:00" FontSize="23" Foreground="#FF158300" ></Label>
                    <Label Content="Триває 1: Час: 24.03.22; 12:00 -
24.03.22; 12:00" FontSize="23" Foreground="#FF830000" ></Label>
                </StackPanel>
            </ScrollViewer>
            <Button x:Name="ChangeInf_Button" BorderThickness="2"
Content="Змінити" HorizontalAlignment="Left" Margin="621,24,0,0"
VerticalAlignment="Top" Height="46" Width="96" Background="{x:Null}"

```

```

BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="ChangeInf_Button_Click"/>
    <TextBlock Name="IncorrectInput_TB" Visibility="Hidden"
HorizontalAlignment="Left" Height="42" Margin="531,28,0,0"
TextWrapping="Wrap" VerticalAlignment="Top" Width="229"
Foreground="#FFFF7979"><Run Language="uk-ua" Text="Правильно введіть нові
дані вершини, щоб змінити їх."/></TextBlock>
    <Label Content="Сходження:" HorizontalAlignment="Left"
Margin="38,78,0,0" VerticalAlignment="Top" Height="48" Width="151"
FontSize="25" Background="{x:Null}" Foreground="#FF0A2D9E"/>
    <Rectangle HorizontalAlignment="Left" Height="24"
Margin="228,91,0,0" VerticalAlignment="Top" Width="26" Fill="#FF158300"/>
    <Label Content="- Завершене" HorizontalAlignment="Left"
Margin="259,78,0,0" VerticalAlignment="Top" Height="48" Width="159"
FontSize="25" Background="{x:Null}" Foreground="#FF0A2D9E"/>
    <Rectangle HorizontalAlignment="Left" Height="24"
Margin="438,91,0,0" VerticalAlignment="Top" Width="26" Fill="#FF830000"/>
    <Label Content="- Не завершене" HorizontalAlignment="Left"
Margin="469,78,0,0" VerticalAlignment="Top" Height="48" Width="200"
FontSize="25" Background="{x:Null}" Foreground="#FF0A2D9E"/>
</Grid>
</Grid>
</Window>

```

```

<Window x:Class="Climbing_Club.Mountains"
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:Climbing_Club"
mc:Ignorable="d"
Title="Mountains" Height="455" Width="800"
WindowStartupLocation="CenterScreen" WindowStyle="None">
    <Grid>
        <Grid x:Name="Header" Margin="0,0,0,377">
            <Border Background="#FF5470EE" HorizontalAlignment="Center"
Height="78" VerticalAlignment="Top" Width="800"/>
            <Label MouseLeftButtonDown="Close_Click"
Style="{StaticResource Close_Btn}" Content="X" Margin="758,19,0,0"/>
            <Label MouseLeftButtonDown="Minimized_Click"
Style="{StaticResource Minimized_Btn}" Content="1" Margin="708,32,0,0" />
            <Image
Source="https://images.twinkl.co.uk/tr/raw/upload/u/ux/mountain-
wiki_ver_1.png" Opacity="0.15" Margin="-76,247,0,-664"/>
            <Label Content="Mountains" HorizontalAlignment="Left"
Height="76" Margin="46,0,0,0" VerticalAlignment="Top" Width="316"
FontSize="50" FontFamily="Gabriola" FontWeight="Normal"
Foreground="White"/>
        </Grid>

        <Grid x:Name="Content" Margin="0,78,0,0">
            <Label Name="Placeholder_MountainSearch" Padding="7,8,0,0"
Content="Знайти гору. Наприклад, 'Говерла'" HorizontalAlignment="Left"
Height="43" Margin="46,28,0,0" VerticalAlignment="Top" Width="392"
FontSize="18" Foreground="#FF707070"/>

```

```

        <TextBox x:Name="MountainName_TB"
TextChanged="MountainName_Changed" Padding="3,5,0,0"
HorizontalAlignment="Left" Height="43" Margin="46,28,0,0"
TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="671"
BorderBrush="#FF122BA3" Background="{x:Null}" FontSize="20"/>
        <Button BorderThickness="2" Content="Головна"
HorizontalAlignment="Left" Margin="621,305,0,0" VerticalAlignment="Top"
Height="46" Width="96" Background="{x:Null}" BorderBrush="#FF3D57D1"
Foreground="#FF5471F6" FontSize="18" FontWeight="Normal"
Click="ToMain_Click"/>
        <ScrollViewer Margin="46,76,83,72">
            <StackPanel x:Name="Mountain_Search_SP">
                <Label Content="Говерла, Висота - 999 м"
FontSize="23" Foreground="#FF000383"
MouseLeftButtonDown="MountainInfo_Click" Cursor="Hand"></Label>
            </StackPanel>
        </ScrollViewer>
        <Button BorderThickness="2" Content="
Додати&#xD;&#xA;вершину" HorizontalAlignment="Left" Margin="323,305,0,0"
VerticalAlignment="Top" Height="57" Width="115" Background="{x:Null}"
BorderBrush="#FF3D57D1" Foreground="#FF5471F6" FontSize="18"
FontWeight="Normal" Click="Add_Mountain"/>
    </Grid>
</Grid>
</Window>

```

Програмна реалізація вікон розробленої інформаційної системи

До вікна [MainWindow](#):

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}

```

```

e)         private void Minimized_Click(object sender, MouseEventArgs
           {
               this.WindowState = WindowState.Minimized;
           }

           private void Close_Click(object sender, MouseEventArgs e)
           {
               this.Close();
           }

           private void Mounatins_Click(object sender, MouseEventArgs
e)         {
               Mountains mw = new Mountains();
               mw.Show();
               this.Close();
           }
           private void Climbers_Click(object sender, MouseEventArgs
e)         {
               Climbers cw = new Climbers();
               cw.Show();
               this.Close();
           }
           private void Climblings_Click(object sender, MouseEventArgs
e)         {
               Climblings cw = new Climblings();
               cw.Show();
               this.Close();
           }
       }
   }
}

```

До вікна [AddClimber](#)

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для AddClimber.xaml
    /// </summary>

```

```

public partial class AddClimber : Window
{
    public AddClimber()
    {
        InitializeComponent();
    }
    DataBase db = new DataBase();
    string surname = "";
    private void SurnameChanged(object sender, TextChangedEventArgs
e)
    {
        if (Surname_TB.Text.Length > 0)
Placeholder_Surname.Visibility = Visibility.Hidden;
        else Placeholder_Surname.Visibility = Visibility.Visible;

        if (IsSurnameExist(Surname_TB.Text))
        {
            surname = Surname_TB.Text;
            IncorrectInput_TB.Visibility = Visibility.Hidden;
            AddClimber_Btn.Visibility = Visibility.Visible;
        }
        else
        {
            IncorrectInput_TB.Visibility = Visibility.Visible;
            AddClimber_Btn.Visibility = Visibility.Hidden;
        }
    }

    private bool IsSurnameExist(string surname)
    {
        foreach (char c in surname)
        {
            try
            {
                int a = Convert.ToInt32(c.ToString());
                return false;
            }
            catch { }
        }
        if (surname.Length <= 2 || surname.Length >= 20)
        {
            return false;
        }
        return true;
    }

    private void AddClimber_Btn_Click(object sender, RoutedEventArgs
e)
    {
        int climbID = -1;
        var getClimbID = db.ReadDB("SELECT * FROM Humans");
        while (getClimbID.Read())
        {
            climbID = getClimbID.GetInt32(0) + 1;
        }
        getClimbID.Close(); db.CloseConnection();
        db.ExecuteNonQuery($"INSERT INTO Humans VALUES
({climbID}, '{surname}')");
        Climbers cw = new Climbers();

```



```

        cw.Show();
        this.Close();
    }

    private void ToMountain_Click(object sender, RoutedEventArgs e)
    {
        Climbers cw = new Climbers();
        cw.Show();
        this.Close();
    }
}

```

До вікна `AddClimbing`

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для AddClimbing.xaml
    /// </summary>
    public partial class AddClimbing : Window
    {
        public AddClimbing()
        {
            InitializeComponent();
            var getMountains = db.ReadDB("SELECT * FROM Mountains");
            while (getMountains.Read())
            {
                mountains.Add(getMountains.GetString(1));
            }
            getMountains.Close(); db.CloseConnection();
            mountains = NormalizeArr(mountains);
        }
        DataBase db = new DataBase();
        List<string> mountains = new List<string>();
        bool mExist = false, stExist = false, ftExist = false;

        DateTime start = DateTime.Parse("01.01.1000");
        DateTime finish = DateTime.Parse("01.01.3000");
        private List<string> NormalizeArr(List<string> arr)
        {
            string alphabet = "абвггдеежзиііклмнопрстуфхцщъя";
            List<string> newArr = new List<string>();

```

```

        foreach (string elem in arr)
        {
            string normalEl = "";
            foreach (char c in elem)
            {
                if (alphabet.Contains(c.ToString().ToLower()))
                {
                    normalEl += c;
                }
            }
            newArr.Add(normalEl);
        }
        return newArr;
    }
    private void Mountain_TB_TextChanged(object sender,
    TextChangedEventArgs e)
    {
        string m = (sender as TextBox).Text;
        if (m.Length > 0)
        {
            Placeholder_Mountain.Visibility = Visibility.Hidden;
            if (mountains.Contains(m))
            {
                mExist = true;
            }
            else mExist = false;
        }
        else Placeholder_Mountain.Visibility = Visibility.Visible;
        if (mExist && stExist && ftExist)
        {
            CanAdd(true);
        }
        else CanAdd(false);
    }
    private void CanAdd (bool check)
    {
        if (check)
        {
            IncorrectInput_TB.Visibility = Visibility.Hidden;
            AddClimbing_Btn.Visibility = Visibility.Visible;
        }
        else
        {
            IncorrectInput_TB.Visibility = Visibility.Visible;
            AddClimbing_Btn.Visibility = Visibility.Hidden;
        }
    }

    private void Start_TB_TextChanged(object sender,
    TextChangedEventArgs e)
    {
        TextBox textBox = (TextBox)sender;
        if (textBox.Text.Length > 0)
        {
            Placeholder_Start.Visibility = Visibility.Hidden;
        }
        else Placeholder_Start.Visibility = Visibility.Visible;
        DateTime dt = new DateTime();
        try

```

```

        {
            dt = DateTime.Parse(textBox.Text);
        }
        catch
        {
            stExist = false;
            CanAdd(false);
            return;
        }
        if (dt > DateTime.Parse("01.01.1902") && dt <
DateTime.Parse("01.01.2076") && dt < finish)
        {
            start = dt;
            stExist = true;
        }
        else
        {
            stExist = false;
            CanAdd(false);
            return;
        }
        if (mExist && stExist && ftExist)
        {
            CanAdd(true);
        }
        else CanAdd(false);
    }

    private void Finish_TB_TextChanged(object sender,
TextChangedEventArgs e)
    {
        TextBox textBox = (TextBox)sender;
        if (textBox.Text.Length > 0)
        {
            Placeholder_Finish.Visibility = Visibility.Hidden;
        }
        else Placeholder_Finish.Visibility = Visibility.Visible;
        DateTime dt = new DateTime();
        try
        {
            dt = DateTime.Parse(textBox.Text);
        }
        catch
        {
            ftExist = false;
            CanAdd(false);
            return;
        }
        if (dt > DateTime.Parse("01.01.1902") && dt <
DateTime.Parse("01.01.2076") && dt > start)
        {
            finish = dt;
            ftExist = true;
        }
        else
        {
            ftExist = false;
            CanAdd(false);
            return;
        }
    }

```

```

        }
        if (mExist && stExist && ftExist)
        {
            CanAdd(true);
        }
        else CanAdd(false);
    }
    private void AddClimbing_Btn_Click(object sender, RoutedEventArgs
e)
    {
        var getID = db.ReadDB("SELECT * FROM Groups ORDER BY
Group_ID");
        int id = -1;
        while (getID.Read())
        {
            id = getID.GetInt32(0) + 1;
        }
        getID.Close(); db.CloseConnection();
        db.ExecuteQuery($"INSERT INTO Groups
VALUES({id}, '{start}', '{finish}')");
        string mName = Mountain_TB.Text;
        var getMID = db.ReadDB($"SELECT * FROM Mountains WHERE
Mountain_Name = '{mName}'");
        getMID.Read();
        int mountID = getMID.GetInt32(0);
        getMID.Close(); db.CloseConnection();

        db.ExecuteQuery($"INSERT INTO Group_Mountain
VALUES({id},{mountID})");

        Climblings cw = new Climblings();
        cw.Show();
        this.Close();
    }

    private void ToGroup_Click(object sender, RoutedEventArgs e)
    {
        Climblings cw = new Climblings();
        cw.Show();
        this.Close();
    }
}

```

До вікна [AddMountain](#)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;

```

```

using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаимодействия для AddMountain.xaml
    /// </summary>
    public partial class AddMountain : Window
    {
        public AddMountain()
        {
            InitializeComponent();
        }
        DataBase db = new DataBase();
        private void IsMountainExist(bool check)
        {
            if (check)
            {
                IncorrectInput_TB.Visibility = Visibility.Hidden;
                AddMountain_Btn.Visibility = Visibility.Visible;
            }
            else
            {
                IncorrectInput_TB.Visibility = Visibility.Visible;
                AddMountain_Btn.Visibility = Visibility.Hidden;
            }
        }
        private void AddMountain_Click(object sender, RoutedEventArgs e)
        {
            int mountID = -1;
            var getID = db.ReadDB("SELECT * FROM Mountains ORDER BY
Mountain_ID");
            while (getID.Read())
            {
                mountID = getID.GetInt32(0) + 1;
            }
            getID.Close(); db.CloseConnection();
            db.ExecuteNonQuery($"INSERT INTO Mountains VALUES
({mountID}, '{MountainName_TB.Text}', '{MountainCountry_TB.Text}',
{MountainHeight_TB.Text})");
            Mountains mw = new Mountains();
            mw.Show();
            this.Close();
        }

        private void InfoChanged(object sender, TextChangedEventArgs e)
        {
            if (MountainHeight_TB.Text.Length > 0)
                Placeholder_Height.Visibility = Visibility.Hidden;
            else Placeholder_Height.Visibility = Visibility.Visible;

            if (MountainCountry_TB.Text.Length > 0)
                Placeholder_Country.Visibility = Visibility.Hidden;
            else Placeholder_Country.Visibility = Visibility.Visible;

            if (MountainName_TB.Text.Length > 0)
                Placeholder_Name.Visibility = Visibility.Hidden;
            else Placeholder_Name.Visibility = Visibility.Visible;
        }
    }
}

```

```

        int height = 0;
        try
        {
            height = Convert.ToInt32(MountainHeight_TB.Text);
        }
        catch
        {
            IsMountainExist(false); return;
        }
        if (IsTitleExist(MountainName_TB.Text) &&
IsTitleExist(MountainCountry_TB.Text)
            && height > 1 && height < 10000)
        {
            if (!IsMountainInDB(MountainName_TB.Text))
            {
                IsMountainExist(true); return;
            }
        }
        IsMountainExist(false);
    }
    private bool IsTitleExist(string str)
    {
        foreach (char c in str)
        {
            try
            {
                int a = Convert.ToInt32(c.ToString());
                return false;
            }
            catch { }
        }
        if (str.Length <= 2 || str.Length >= 20)
        {
            return false;
        }
        return true;
    }
    private bool IsMountainInDB(string mountain)
    {
        var checkMountain = db.ReadDB($"SELECT * FROM Mountains WHERE
Mountain_Name = '{mountain}'");
        while (checkMountain.Read())
        {
            checkMountain.Close(); db.CloseConnection();
            return true;
        }
        checkMountain.Close(); db.CloseConnection();
        return false;
    }

    private void ToMountain_Click(object sender, RoutedEventArgs e)
    {
        Mountains mw = new Mountains();
        mw.Show();
        this.Close();
    }
}
}

```

До вікна `ClimberAbout`

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для ClimberAbout.xaml
    /// </summary>
    public partial class ClimberAbout : Window
    {
        public ClimberAbout()
        {
            InitializeComponent();
        }
        DataBase db = new DataBase();
        private void Close_Click(object sender, MouseButtonEventArgs e)
        {
            this.Close();
        }
        private void Minimazed_Click(object sender, MouseButtonEventArgs e)
        {
            this.WindowState = WindowState.Minimized;
        }
        public void GetInfo(string data)
        {
            string surname = data.Split(':')[1];
            int id = int.Parse(data.Split(':')[0]);
            Surname_LB.Content = surname;
            Dictionary<string, int> climbings = new Dictionary<string,
int>();

            var getClimbings = db.ReadDB("SELECT dbo.Humans.Passport,
dbo.Group_Human.Group_ID, dbo.Group_Mountain.Group_ID AS Expr1,
dbo.Mountains.Mountain_Name FROM      dbo.Mountains INNER JOIN
dbo.Group_Mountain ON dbo.Mountains.Mountain_ID =
dbo.Group_Mountain.Mountain_ID CROSS JOIN dbo.Humans INNER JOIN
dbo.Group_Human ON dbo.Humans.Passport = dbo.Group_Human.Passport WHERE
dbo.Group_Human.Group_ID = dbo.Group_Mountain.Group_ID");
            while (getClimbings.Read())
            {
                string mountain = getClimbings.GetString(3);
                if (getClimbings.GetInt32(0) != id) continue;
            }
        }
    }
}
```

```

        if (climblings.ContainsKey(mountain))
        {
            climblings[mountain]++;
        }
        else climblings.Add(mountain, 1);
    }
    getClimbings.Close(); db.CloseConnection();
    foreach (var climbing in climblings)
    {
        Label lbl = new Label
        {
            FontSize = 23, Foreground = new
SolidColorBrush(Color.FromArgb(255,0,3,131)),
            Content = $"{climbing.Key} - {climbing.Value} разів"
        };
        Mountains.Children.Add(lbl);
    }
}

private void Climblings_Click(object sender, RoutedEventArgs e)
{
    Climbers cw = new Climbers();
    cw.Show();
    this.Close();
}
}
}

```

До вікна Climbers

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для Climbers.xaml
    /// </summary>
    public partial class Climbers : Window
    {
        public Climbers()
        {
            InitializeComponent();
            Surnames_SP.Children.Clear();
        }
        DateTime start = DateTime.Parse("01.01.1902 1:00");
    }
}

```



```

DateTime finish = DateTime.Parse("01.01.1902 2:00");
DataBase db = new DataBase();
private void Minimazed_Click(object sender, MouseEventArgs
e)
{
    this.WindowState = WindowState.Minimized;
}
private void Close_Click(object sender, MouseEventArgs e)
{
    this.Close();
}
private void ToMain_Click(object sender, RoutedEventArgs e)
{
    MainWindow mw = new MainWindow();
    mw.Show();
    this.Close();
}

private void Time_Changed(object sender, TextChangedEventArgs e)
{
    if (Start_TB.Text.Length > 0) Placeholder_Start.Visibility =
Visibility.Hidden;
    else Placeholder_Start.Visibility = Visibility.Visible;

    if (Finish_TB.Text.Length > 0) Placeholder_Finish.Visibility
= Visibility.Hidden;
    else Placeholder_Finish.Visibility = Visibility.Hidden;

    if (Start_TB.Text.Length < 8 || Finish_TB.Text.Length < 8)
return;

    try
    {
        start = DateTime.Parse(Start_TB.Text);
        finish = DateTime.Parse(Finish_TB.Text);
    }
    catch
    {
        return;
    }

    if (start < DateTime.Parse("01.01.1901") || start >
DateTime.Parse("01.01.2076")
        || finish < DateTime.Parse("01.01.1901") || finish >
DateTime.Parse("01.01.2076"))
        return;

    PrintClimbers();
}
private void PrintClimbers()
{
    List<string> surnames = new List<string>();
    List<int> ids = new List<int>();
    var getClimbers = db.ReadDB($"SELECT dbo.Groups.Start,
dbo.Groups.Finish, dbo.Humans.Surname, dbo.Humans.Passport FROM
dbo.Groups INNER JOIN dbo.Group_Human ON dbo.Groups.Group_ID =
dbo.Group_Human.Group_ID INNER JOIN dbo.Humans ON
dbo.Group_Human.Passport = dbo.Humans.Passport WHERE (dbo.Groups.Start
BETWEEN ('{start}')) AND ('{finish}')) OR (dbo.Groups.Finish BETWEEN
('{start}') AND ('{start}'))");

```

```

try
{
    while (getClimbers.Read())
    {
        surnames.Add(getClimbers.GetString(2));
        ids.Add(getClimbers.GetInt32(3));
    }
}
catch
{
    getClimbers.Close(); db.CloseConnection();
}
getClimbers.Close(); db.CloseConnection();
surnames = surnames.Distinct().ToList();
ids = ids.Distinct().ToList();
Surnames_SP.Children.Clear();
for (int i = 0; i < surnames.Count; i++)
{
    Label lbl = new Label
    {
        Content = $"{ids[i]}:{surnames[i]}",
        FontSize = 23,
        Foreground = new SolidColorBrush(Color.FromArgb(255,
0, 3, 131)),
    };
    lbl.MouseLeftButtonDown += ClimberAbout;
    Surnames_SP.Children.Add(lbl);
}

private void ClimberAbout(object sender, MouseButtonEventArgs e)
{
    ClimberAbout caw = new ClimberAbout();
    caw.GetInfo((sender as Label).Content.ToString());
    caw.Show();
    this.Close();
}

private void AddClimber(object sender, RoutedEventArgs e)
{
    AddClimber acw = new AddClimber();
    acw.Show();
    this.Close();
}
}
}

```

До вікна `Climbings`

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Media;

```

```

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаимодействия для Climblings.xaml
    /// </summary>
    public partial class Climblings : Window
    {
        public Climblings()
        {
            InitializeComponent();
            Groups_SP.Children.Clear();
        }
        DataBase db = new DataBase();
        private void Minimazed_Click(object sender, MouseButtonEventArgs
e)
        {
            this.WindowState = WindowState.Minimized;
        }

        DateTime start = DateTime.Parse("01.01.1902 1:00");
        DateTime finish = DateTime.Parse("01.01.1902 2:00");
        private void Close_Click(object sender, MouseButtonEventArgs e)
        {
            this.Close();
        }
        private void ToMain_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mw = new MainWindow();
            mw.Show();
            this.Close();
        }

        private string NormalizeStr(string str)
        {
            string alphabet = "абвггдеежзиіїйклмнопрстуфхцщщя";

            string normalEl = "";
            foreach (char c in str)
            {
                if (alphabet.Contains(c.ToString().ToLower()))
                {
                    normalEl += c;
                }
            }
            return normalEl;
        }
        private void Time_Changed(object sender, TextChangedEventArgs e)
        {
            if (Start_TB.Text.Length > 0) Placeholder_Start.Visibility =
Visibility.Hidden;
            else Placeholder_Start.Visibility = Visibility.Visible;

            if (Finish_TB.Text.Length > 0) Placeholder_Finish.Visibility
= Visibility.Hidden;
            else Placeholder_Finish.Visibility = Visibility.Hidden;

            if (Start_TB.Text.Length < 8 || Finish_TB.Text.Length < 8)
return;
            try

```

```

        {
            start = DateTime.Parse(Start_TB.Text);
            finish = DateTime.Parse(Finish_TB.Text);
        }
        catch
        {
            return;
        }

        if (start < DateTime.Parse("01.01.1901") || start >
DateTime.Parse("01.01.2076")
            || finish < DateTime.Parse("01.01.1901") || finish >
DateTime.Parse("01.01.2076"))
            return;

        PrintClimbings();
    }

    private void PrintClimbings()
    {
        Groups_SP.Children.Clear();
        string query = $"SELECT dbo.Groups.Group_ID,
dbo.Mountains.Mountain_Name, dbo.Groups.Start, dbo.Groups.Finish FROM
dbo.Groups INNER JOIN dbo.Group_Mountain ON dbo.Groups.Group_ID =
dbo.Group_Mountain.Group_ID INNER JOIN dbo.Mountains ON
dbo.Group_Mountain.Mountain_ID = dbo.Mountains.Mountain_ID WHERE
(dbo.Groups.Start BETWEEN ('{start}')) AND ('{finish}')) OR
(dbo.Groups.Finish BETWEEN ('{start}')) AND ('{finish}'))";
        var getCl = db.ReadDB(query);
        try
        {
            while (getCl.Read())
            {
                Label lbl = new Label
                {
                    Content = $"Tpyna {getCl.GetValue(0).ToString()}
- {NormalizeStr( getCl.GetValue(1).ToString())},
{getCl.GetValue(2).ToString()} - {getCl.GetValue(3).ToString()}",
                    FontSize = 23, Cursor = Cursors.Hand,
                    Foreground = new
SolidColorBrush(Color.FromArgb(255, 0, 3, 131)),
                };
                lbl.MouseLeftButtonDown += GroupAbout_Click;
                Groups_SP.Children.Add(lbl);
            }
            getCl.Close(); db.CloseConnection();
        }
        catch
        {
            getCl.Close(); db.CloseConnection();
        }
    }

    private void GroupAbout_Click(object sender, MouseButtonEventArgs
e)
    {
        GroupAbout ga = new GroupAbout();
        ga.SetGroup(int.Parse((sender as
Label).Content.ToString().Split(' ')[1]));
    }

```

```

        ga.Show();
        this.Close();
    }

    private void ToAddClimbing_Click(object sender, RoutedEventArgs
e)
    {
        AddClimbing acw = new AddClimbing();
        acw.Show();
        this.Close();
    }
}

```

До вікна `GroupAbout`

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для GroupAbout.xaml
    /// </summary>
    public partial class GroupAbout : Window
    {
        public GroupAbout()
        {
            InitializeComponent();
        }
        DataBase db = new DataBase();
        private void Close_Click(object sender, MouseButtonEventArgs e)
        {
            this.Close();
        }
        private void Minimized_Click(object sender, MouseButtonEventArgs
e)
        {
            this.WindowState = WindowState.Minimized;
        }
        public void SetGroup(int gID)
        {
            Climbers_TB.Text = "";
            Group_Name_LB.Content = "Група " + gID;
            var getClimbers = db.ReadDB($"SELECT dbo.Groups.Group_ID,
            dbo.Group_Human.Passport, dbo.Humans.Surname FROM dbo.Groups INNER JOIN

```

```

dbo.Group_Human ON dbo.Groups.Group_ID = dbo.Group_Human.Group_ID INNER
JOIN dbo.Humans ON dbo.Group_Human.Passport = dbo.Humans.Passport WHERE
dbo.Groups.Group_ID = {gID}");
    while (getClimbers.Read())
    {
        Climbers_TB.Text += getClimbers.GetInt32(1) + ":" +
getClimbers.GetString(2) + " ";
    }
    getClimbers.Close();
    db.CloseConnection();
    ToAdd_Btn.Content = " Додати \нальпініста";
    ToAdd_Btn.BorderBrush = new
SolidColorBrush(Color.FromArgb(255, 61, 87, 209));
    ToAdd_Btn.Foreground = new
SolidColorBrush(Color.FromArgb(255, 85, 113, 246));
    AddClimber_Grid.Visibility = Visibility.Hidden;
}
private void ToAddClimber_Click(object sender, RoutedEventArgs e)
{
    Button btn = sender as Button;
    if (btn.Content.ToString() == " Додати \нальпініста")
    {
        AddClimber_Grid.Visibility = Visibility.Visible;
        btn.Content = "Відміна";
        btn.BorderBrush = new
SolidColorBrush(Color.FromArgb(255, 209, 125, 61));
        btn.Foreground = new SolidColorBrush(Color.FromArgb(255,
246, 160, 84));
    }
    else
    {
        AddClimber_Grid.Visibility = Visibility.Hidden;
        btn.Content = " Додати \нальпініста";
        btn.BorderBrush = new SolidColorBrush(Color.FromArgb(255,
61, 87, 209));
        btn.Foreground = new SolidColorBrush(Color.FromArgb(255,
85, 113, 246));
    }
}

private void Climblings_Click(object sender, RoutedEventArgs e)
{
    Climblings climblings = new Climblings();
    climblings.Show();
    this.Close();
}

private void ClimberToAdd_Changed(object sender,
TextChangedEventArgs e)
{
    TextBox tb = sender as TextBox;
    if (tb.Text.Length > 0)
    {
        Placeholder_NewClimber.Visibility = Visibility.Hidden;
        CanAddClimber(tb.Text);
    }
    else
    {
        Placeholder_NewClimber.Visibility = Visibility.Visible;
    }
}

```

```

        CanAdd(false);
    }
}
private void CanAddClimber(string climber)
{
    int passport = -1;
    try
    {
        passport = int.Parse(climber);
    }
    catch { CanAdd(false); return; }

    var IDAvailable = db.ReadDB($"SELECT dbo.Groups.Group_ID,
dbo.Group_Human.Passport, dbo.Humans.Surname FROM dbo.Groups INNER JOIN
dbo.Group_Human ON dbo.Groups.Group_ID = dbo.Group_Human.Group_ID INNER
JOIN dbo.Humans ON dbo.Group_Human.Passport = dbo.Humans.Passport WHERE
dbo.Humans.Passport = {passport} AND dbo.Groups.Group_ID =
{Group_Name_LB.Content.ToString().Split(' ')[1]}");
    while (IDAvailable.Read())
    {
        IDAvailable.Close(); db.CloseConnection();
        CanAdd(false);
        return;
    }
    IDAvailable.Close(); db.CloseConnection();

    CanAdd(true);
}
private void CanAdd(bool check)
{
    if (check)
    {
        InputIncorrect.Visibility = Visibility.Hidden;
        AddCl_Btn.IsEnabled = true;
    }
    else
    {
        InputIncorrect.Visibility = Visibility.Visible;
        AddCl_Btn.IsEnabled = false;
    }
}
private void AddClimber_Click(object sender, RoutedEventArgs e)
{
    string passport = NewClData.Text.Split(':')[0];
    int gID = int.Parse(Group_Name_LB.Content.ToString().Split('
')[1]);

    db.ExecuteQuery($"INSERT INTO Group_Human VALUES
({gID},{passport})");

    NewClData.Text = "";
    CanAdd(false);
    SetGroup(gID);
}
}
}

```

До вікна Mountain_About

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для Mountain_About.xaml
    /// </summary>
    public partial class Mountain_About : Window
    {
        public Mountain_About()
        {
            InitializeComponent();
        }
        private void Close_Click(object sender, MouseButtonEventArgs e)
        {
            this.Close();
        }

        private void Minimized_Click(object sender, MouseButtonEventArgs
e)
        {
            this.WindowState = WindowState.Minimized;
        }

        private void ToMountain_Click(object sender, RoutedEventArgs e)
        {
            Mountains mountains = new Mountains();
            mountains.Show();
            this.Close();
        }
        DataBase db = new DataBase();
        string newCountry = "";
        int newHeight = 0;
        public void SetMountain(string mountain)
        {
            Mountain_Name_LB.Content = mountain;
            var getMntInf = db.ReadDB($"Select * FROM Mountains WHERE
Mountain_Name = '{mountain}'");
            while (getMntInf.Read())
            {
                MountainCountry_TB.Text = getMntInf.GetString(2);
                MountainHeight_TB.Text =
getMntInf.GetInt32(3).ToString();
            }
            getMntInf.Close(); db.CloseConnection();
        }
    }
}
```



```

        Climblings_SP.Children.Clear();
        var getClimbs = db.ReadDB($"SELECT
dbo.Mountains.Mountain_Name, dbo.Group_Mountain.Group_ID,
dbo.Groups.Start, dbo.Groups.Finish FROM dbo.Mountains INNER JOIN
dbo.Group_Mountain ON dbo.Mountains.Mountain_ID =
dbo.Group_Mountain.Mountain_ID INNER JOIN dbo.Groups ON
dbo.Group_Mountain.Group_ID = dbo.Groups.Group_ID WHERE Mountain_Name =
'{mountain}'");
        while (getClimbs.Read())
        {
            int groupID = getClimbs.GetInt32(1);
            DateTime start = (DateTime)getClimbs.GetSqlDateTime(2);
            DateTime finish = (DateTime)getClimbs.GetSqlDateTime(3);

            SolidColorBrush forge = DateTime.Now > finish ?
            new SolidColorBrush(Color.FromArgb(255, 21, 131, 0)) :
            new SolidColorBrush(Color.FromArgb(255, 131, 0, 0));

            Climblings_SP.Children.Add(new Label
            {
                FontSize = 23,
                Foreground = forge,
                Content = $"Трпна {groupID}: Час: {start.ToString()}
- {finish.ToString()}"
            });
        }
        getClimbs.Close(); db.CloseConnection();
    }

    private void InfoChanged(object sender, TextChangedEventArgs e)
    {
        if (IncorrectInput_TB == null) return;
        newCountry = MountainCountry_TB.Text.ToString();
        try
        {
            newHeight =
            Convert.ToInt32(MountainHeight_TB.Text.ToString());
        }
        catch
        {
            ChangeInf_Button.Visibility = Visibility.Hidden;
            IncorrectInput_TB.Visibility = Visibility.Visible;
            return;
        }
        if (IsMountCountryAvailable(newCountry) && newHeight > 1 &&
        newHeight < 10000)
        {
            ChangeInf_Button.Visibility = Visibility.Visible;
            IncorrectInput_TB.Visibility = Visibility.Hidden;
        }
        else
        {
            ChangeInf_Button.Visibility = Visibility.Hidden;
            IncorrectInput_TB.Visibility = Visibility.Visible;
        }
    }
    private bool IsMountCountryAvailable(string country)
    {
        foreach (char c in country)

```

```

        {
            try
            {
                int a = Convert.ToInt32(c.ToString());
                return false;
            }
            catch { }
        }
        if (country.Length <= 2 || country.Length >= 20)
        {
            return false;
        }
        return true;
    }

    private void ChangeInf_Button_Click(object sender,
RoutedEventArgs e)
    {
        db.ExecuteQuery($"UPDATE Mountains SET Mountain_Country =
'{newCountry}', Mountain_Height = {newHeight} WHERE Mountain_Name =
'{Mountain_Name_LB.Content}'");

        Mountains mountains = new Mountains();
        mountains.Show();
        this.Close();
    }
}
}

```

До вікна Mountains

```

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

namespace Climbing_Club
{
    /// <summary>
    /// Логика взаємодії для Mountains.xaml
    /// </summary>
    public partial class Mountains : Window
    {
        public Mountains()
        {
            InitializeComponent();
            Mountain_Search_SP.Children.Clear();
        }
        DataBase db = new DataBase();
    }
}

```

```

e) private void Minimized_Click(object sender, MouseEventArgs
{
    this.WindowState = WindowState.Minimized;
}
private void Close_Click(object sender, MouseEventArgs e)
{
    this.Close();
}
private string NormalizeStr(string str)
{
    string alphabet = "абвггдеежзиііклмнопрстуфхцщщья";

    string normalEl = "";
    foreach (char c in str)
    {
        if (alphabet.Contains(c.ToString().ToLower()))
        {
            normalEl += c;
        }
    }
    return normalEl;
}
private void MountainName_Changed(object sender,
TextChangedEventArgs e)
{
    TextBox tb = sender as TextBox;
    Mountain_Search_SP.Children.Clear();
    if (tb.Text.Length > 0)
    {
        Placeholder_MountainSearch.Visibility =
Visibility.Hidden;
        var getMountains = db.ReadDB($"SELECT * FROM Mountains
WHERE Mountain_Name LIKE '%{tb.Text}%'");
        while (getMountains.Read())
        {
            Label lbl = new Label
            {
                Content = $"{NormalizeStr(
getMountains.GetString(1))}, Висота - {getMountains.GetInt32(3)}",
                FontSize = 23,
                Foreground = new
SolidColorBrush(Color.FromArgb(255, 0, 3, 131)),
                Cursor = Cursors.Hand,
            };
            lbl.MouseLeftButtonDown += MountainInfo_Click;

            Mountain_Search_SP.Children.Add(lbl);
        }
        getMountains.Close();
        db.CloseConnection();
    }
    else Placeholder_MountainSearch.Visibility =
Visibility.Visible;
}

private void MountainInfo_Click(object sender,
MouseButtonEventArgs e)
{

```

```

        Mountain_About maw = new Mountain_About();
        maw.SetMountain((sender as
Label).Content.ToString().Split(',')[0]);
        maw.Show();
        this.Close();
    }

    private void ToMain_Click(object sender, RoutedEventArgs e)
    {
        MainWindow mw = new MainWindow();
        mw.Show();
        this.Close();
    }

    private void Add_Mountain(object sender, RoutedEventArgs e)
    {
        AddMountain amw = new AddMountain();
        amw.Show();
        this.Close();
    }
}
}

```

Функціональна програмна реалізація класів, використаних у WPF вікнах для коректної взаємодії з БД розробленої інформаційної системи

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Climbing_Club
{
    internal class DataBase
    {
        private static string connectStr = "Data Source=DESKTOP-
LG3P941;Initial Catalog=Climbing_Club;Integrated Security=True";
        private SqlConnection connection = new SqlConnection(connectStr);

        public SqlDataReader ReadDB(string query)
        {
            connection.Open();
            SqlCommand command = new SqlCommand(query, connection);
            SqlDataReader reader = command.ExecuteReader();
            return reader;
        }

        public void ExecuteQuery(string query)
        {
            connection.Open();
            new SqlCommand(query, connection).ExecuteNonQuery();
            connection.Close();
        }

        public void CloseConnection()
    }
}

```

```
        {  
            connection.Close();  
        }  
    }  
}
```

ДОДАТОК Б - Основні характеристики ПК, на якому розроблена та протестована інформаційна система

Процесор	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
Оперативна пам'ять	8,00 ГБ
Тип системи	64-розрядна операційна система, процесор на базі архітектури x64