

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об'єктно-орієнтоване програмування»

на тему:

**«Програма проведення тестування»**

студента I курсу групи ПЗ-21-4  
спеціальності 121 «Інженерія програмного  
забезпечення»

Грибана Владислава Валентиновича  
(прізвище, ім'я та по-батькові)

Керівник старший викладач кафедри ПЗ  
Чижмотря О.В

Дата захисту: " \_\_ " \_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії

\_\_\_\_\_  
(підпис) Т.А.Вакалюк  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис) Д.В.Прохорчук  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис) О.В.Власенко  
(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
Факультет інформаційно-комп'ютерних технологій  
Кафедра інженерії програмного забезпечення  
Освітній рівень: бакалавр  
Спеціальність 121 «Комп'ютерна інженерія»

«ЗАТВЕРДЖУЮ»

В.о.зав. кафедри

\_\_\_\_\_ А.В.Морозов  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

ЗАВДАННЯ  
НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ  
Грибану Владиславу Валентиновичу

- Тема роботи: розробка програми проведення тестування,  
керівник роботи: старший викладач кафедри інженерії програмного забезпечення Чижмотря О.В.
- Строк подання студентом: “ 25 ” червня 2022р.
- Вихідні дані до роботи: Розробити програму проведення тестування
- Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
  - Постановка завдання
  - Аналіз аналогічних розробок
  - Алгоритми роботи програми
  - Опис роботи програми
  - Програмне дослідження
- Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
  - Презентація до КП \_\_\_\_\_ 2.

Посилання на репозиторій: <https://github.com/Fr0stYANN/courseworkd>  
Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,1			
1,2			
1,2,3			

6. Дата видачі завдання “ 15 ” квітня 2022 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	25.03.2022	
2	Пошук, огляд та аналіз аналогічних розробок	30.03.2022	
3	Формулювання технічного завдання	05.04.2022	
4	Опрацювання літературних джерел	10.04.2022	
5	Проектування структури	20.04.2022	
6	Написання програмного коду	10.05.2022	
7	Відлагодження	20.05.2022	
8	Написання пояснювальної записки	15.06.2022	
9	Захист	09.07.2022	

**Студент**

\_\_\_\_\_

(підпис)

**Керівник проекту**

\_\_\_\_\_

(підпис)

**Грибан В.В**

(прізвище та ініціали)

**Чижмотря О.В**

(прізвище та ініціали)

## РЕФЕРАТ

Завданням курсового проекту була розробка «програми проведення тестування».

Пояснювальна записка до курсового проекту (роботи) на тему «розробка програми проведення тестування» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 20 сторінках друкованого тексту. Пояснювальна записка має 10 сторінок додатків. Список використаних джерел містить 8 найменувань і займає 1 сторінку. В роботі наведено 12 рисунків. Загальний обсяг роботи – 30 сторінок.

Ключові слова: С#, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМА ПРОВЕДЕННЯ ТЕСТУВАННЯ, ООП.

					ДУ «Житомирська політехніка».22.121.08.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Грибан В.В			Розробка програми проведення тестування	Літ.	Арк.	Аркушів
Перевір.		Чижмотря О.В					4	46
Керівник						ФІКТ Гр. ІПЗ 21-4(1)		
Н. контр.								
Зав. каф.								

## Зміст

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	7
1.1. Аналіз задачі, засобів та методів її вирішення .....	7
1.2. Аналіз існуючого програмного забезпечення за тематикою курсового проекту (роботи). ....	7
Висновки до першого розділу: .....	10
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	10
2.1 Проектування загального алгоритму роботи програми.....	10
2.2 Розробка функціональних алгоритмів роботи програми.....	11
2.3 Розробка програмного забезпечення.....	11
Висновки до другого розділу:.....	12
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ .....	12
3.1 Опис роботи з програмним додатком (Опис інтерфейсу) .....	12
3.2 Тестування роботи програмного забезпечення.....	19
Висновки до третього розділу .....	19
ВИСНОВКИ.....	20
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	20
Додатки.....	21

## ВСТУП

У сучасному світі пандемії, воєнного стану у нашій країні постало питання дистанційного навчання для студентів, таким чином, щоб викладач міг перевірити засвоєні студентом знання, адже контролювати засвоєння знань у дистанційному форматі вкрай важко. Тому популярними стали програмні продукти для проведення тестувань онлайн. Слідкуючи за трендами онлайн тестувань у веб-додатках, я вирішив обрати тему розроблення десктопного додатку проведення тестування. Мова програмування с# одна з найпопулярніших та найкращих мов, яку використовують для розробки windows додатків. Одна з головних переваг С# є велика кількість зручних бібліотек та фреймворків які дозволяють сильно покращити швидкість та якість написання коду. Цей курсовий проект включає в себе розробку програмного додатку проведення тестування, яка була реалізована за допомогою мови програмування С# та Windows Presentation Foundation, та реляційної бази даних MS SQL SERVER.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		6

## РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

### 1.1. Аналіз задачі, засобів та методів її вирішення

Програма проведення тестування – програма, яка має полегшити викладачам збирання результатів проходження учнями тестувань.

Завданням курсового проекту є створення програми проведення тестування, яке буде мати наступні функції :

- зберігання результатів проведення тестування у базі даних SQL
- реєстрацію користувача
- авторизація користувача
- обрання певного предмету, для проходження тесту з нього
- можливість отримати результати тестування

Програма була розроблена за допомогою Visual Studio 2019 та Windows Presentation Foundation. Для зберігання тестів використовувалась десеріалізація та серіалізація XML. Дані користувачів зберігаються у базі даних MS SQL.

### 1.2. Аналіз існуючого програмного забезпечення за тематикою курсового проекту (роботи).

На даний момент часу найбільш популярними програмами проведення тестування є :

**Айрен** – програма, яка дозволяє безкоштовно проводити тестування в локальній мережі, через інтернет або на одиночних комп'ютерах.

<https://irenproject.com/>

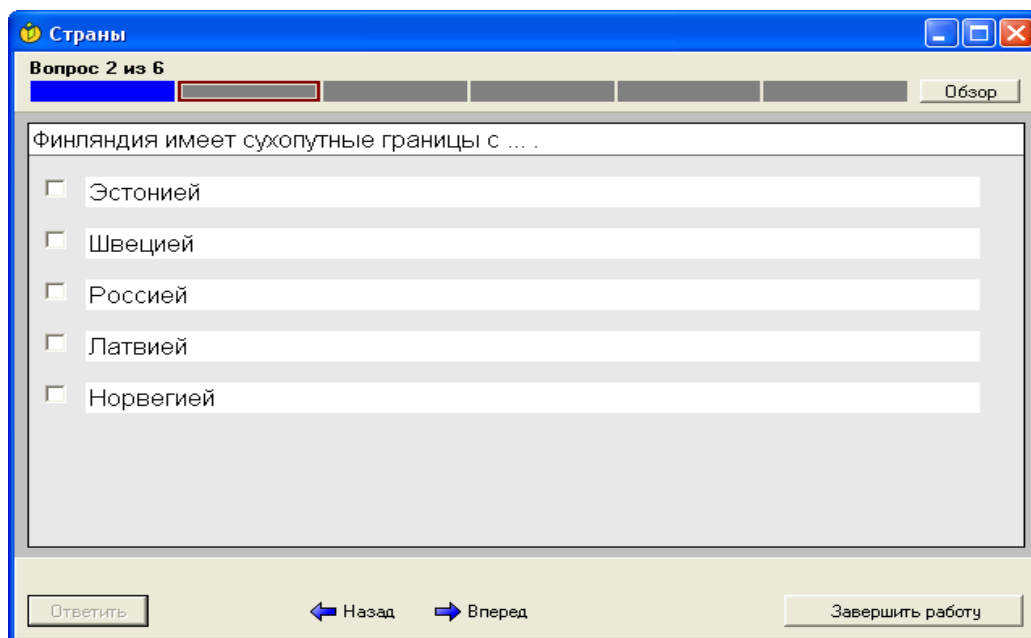


рис. 1.1. Питання тесту

		Грибан В.В.			ДУ «Житомирська політехніка». 22.121.08.000 - ПЗ	Арк. 7
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

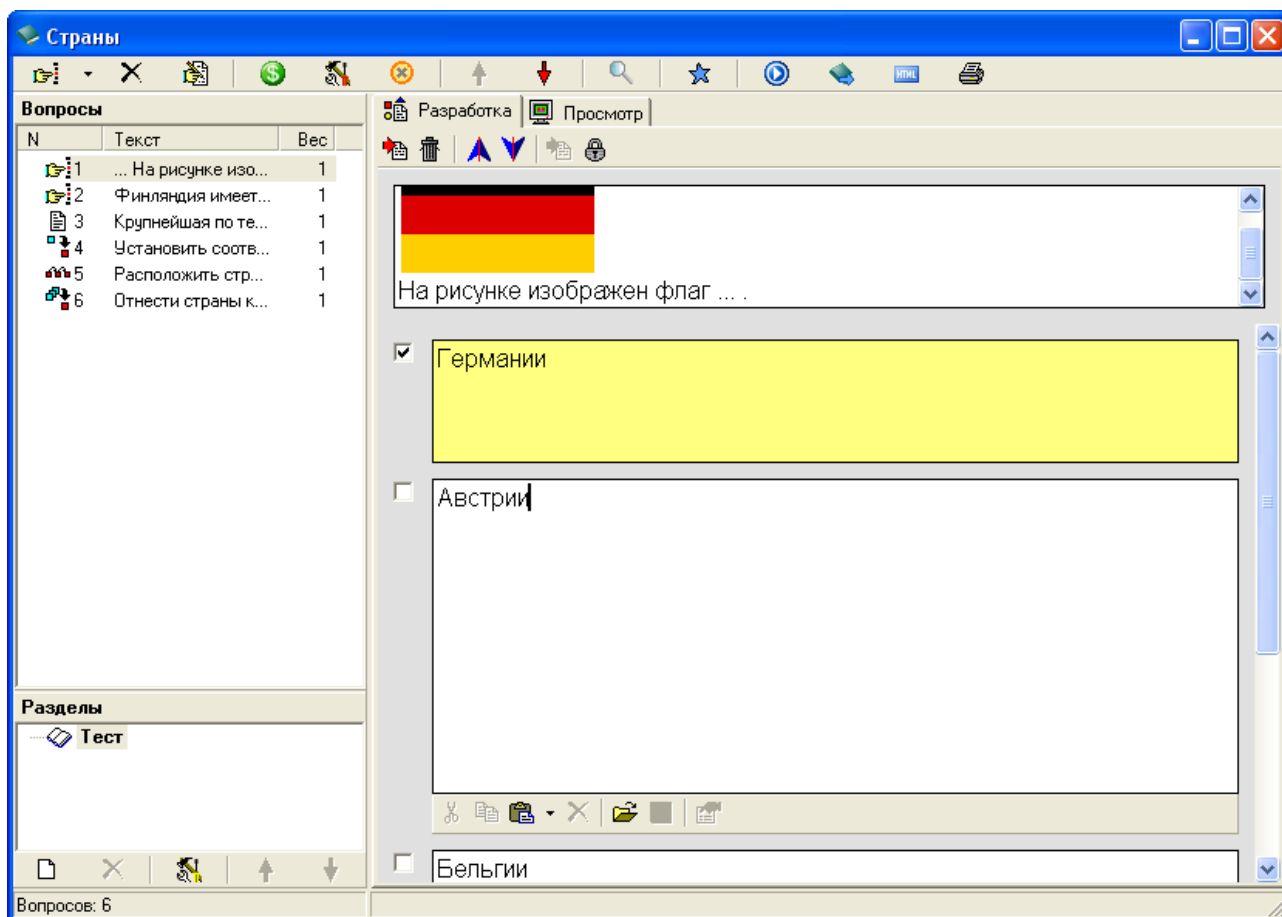


рис. 1.2. Питание тесту.

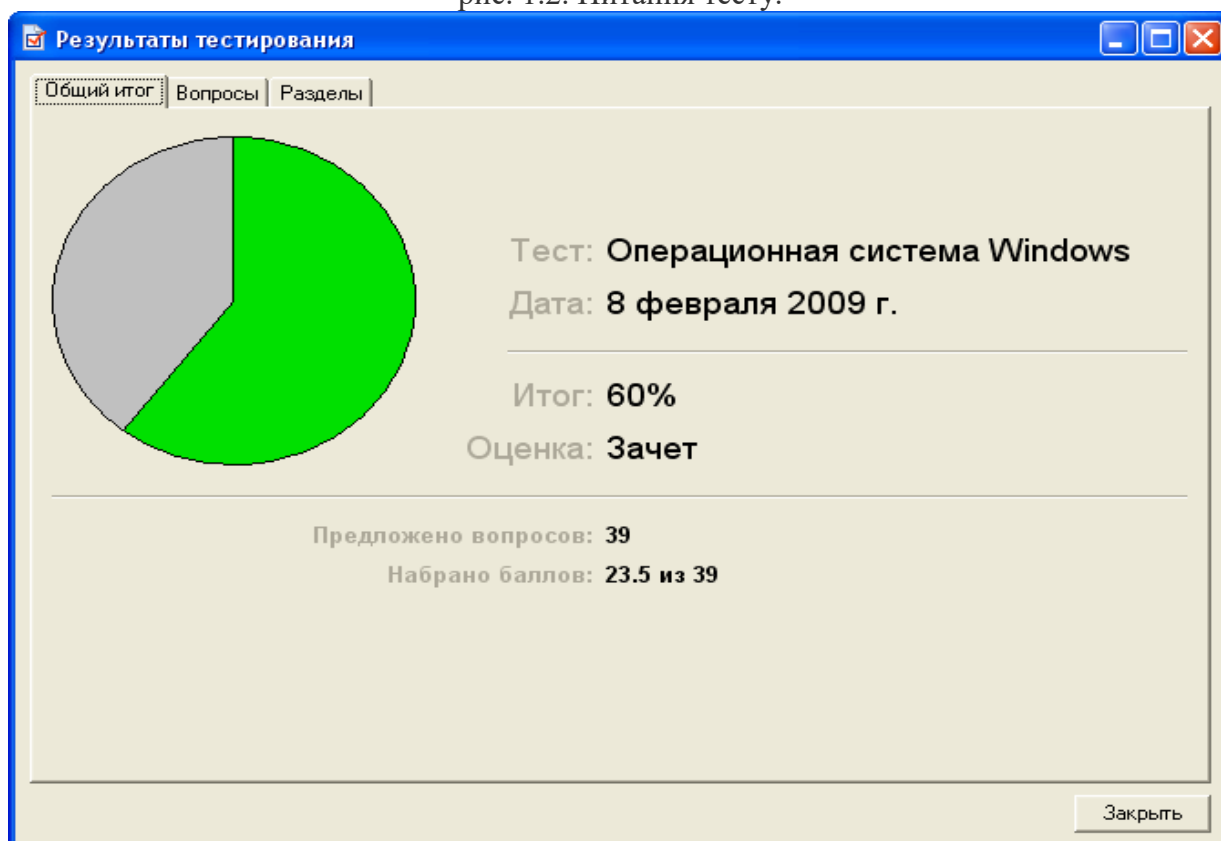


рис. 1.3. Результат тесту.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк. 8
		Чижевська О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		



Переваги:

1. Продумано дуже велику кількість можливого виду запитання.
2. Можна проводити тестування в локальній мережі.

Недоліки:

1. Програма була написана давно, та її ніхто не оновлює.
2. Складний процес реєстрації.

**MyTestXPro** – це система програм для створення і проведення комп'ютерного тестування знань, збору та аналізу результатів.

<https://mytest.klyaksa.net>

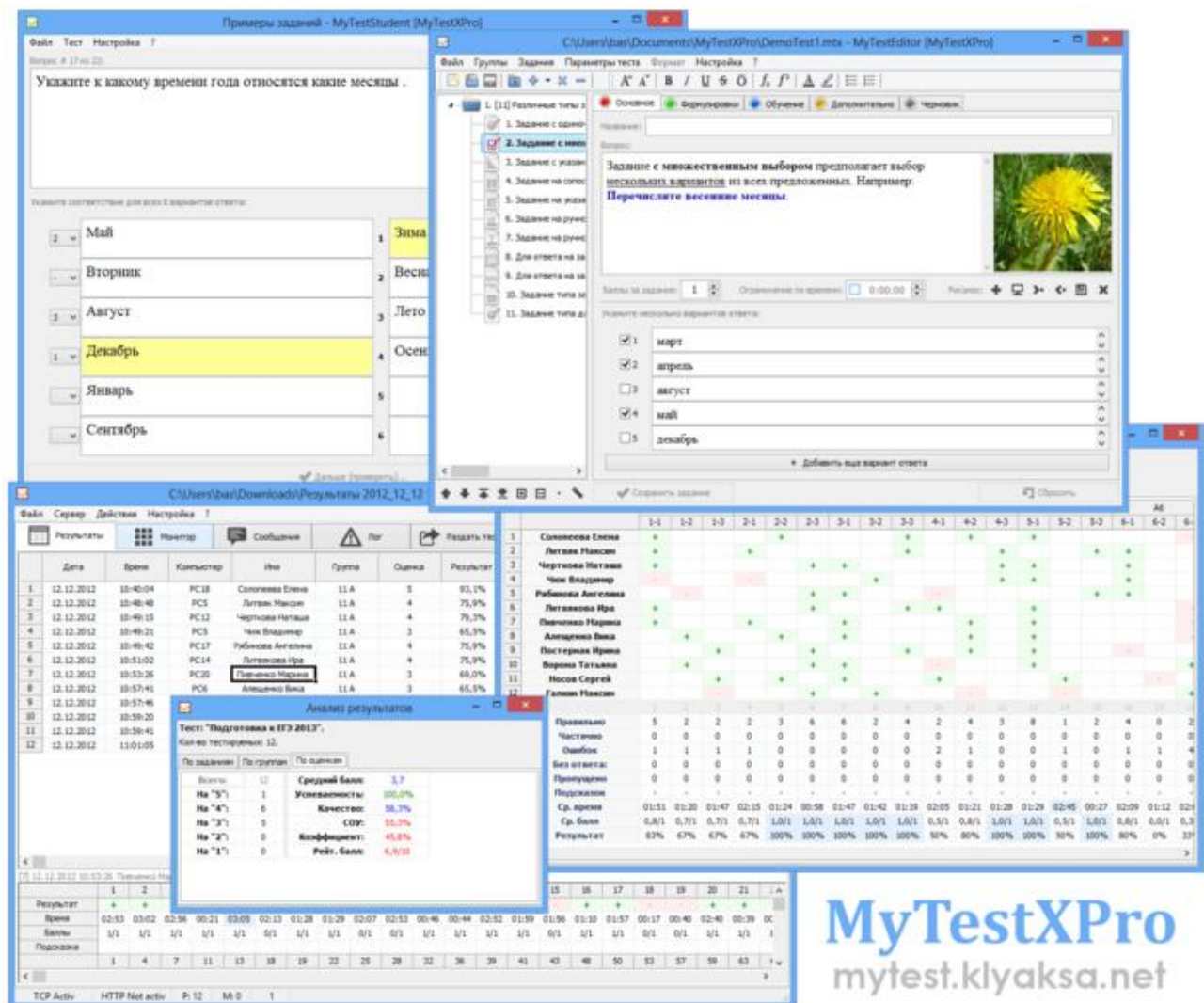


рис. 1.4.Інтерфейс MyTestXPro.

Переваги:

1. Дуже хороша система аналізу результатів.
2. Продумано дуже багато варіантів видів запитань.
3. Приємний інтерфейс.

Недоліки:

1. Завдання обов'язково мають бути поділеними на певні групи.
2. Немає можливості завантажити тест формату xml, тощо.

### Висновки до першого розділу:

Після проведення аналізу аналогів програми для проведення тестування, я надихнувся, зрозумів які переваги та недоліки мають подібні аналоги, постарався перейняти позитивні сторони до свого проекту, та уникнути негативні.

## РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Проектування загального алгоритму роботи програми

Після етапу аналізу та чіткого уявлення поставленої задачі, можна перейти до проектування алгоритму роботи програми.

На рис. 2.1. зображена схема програми, того які можливості будуть мати певні види користувачів програми.

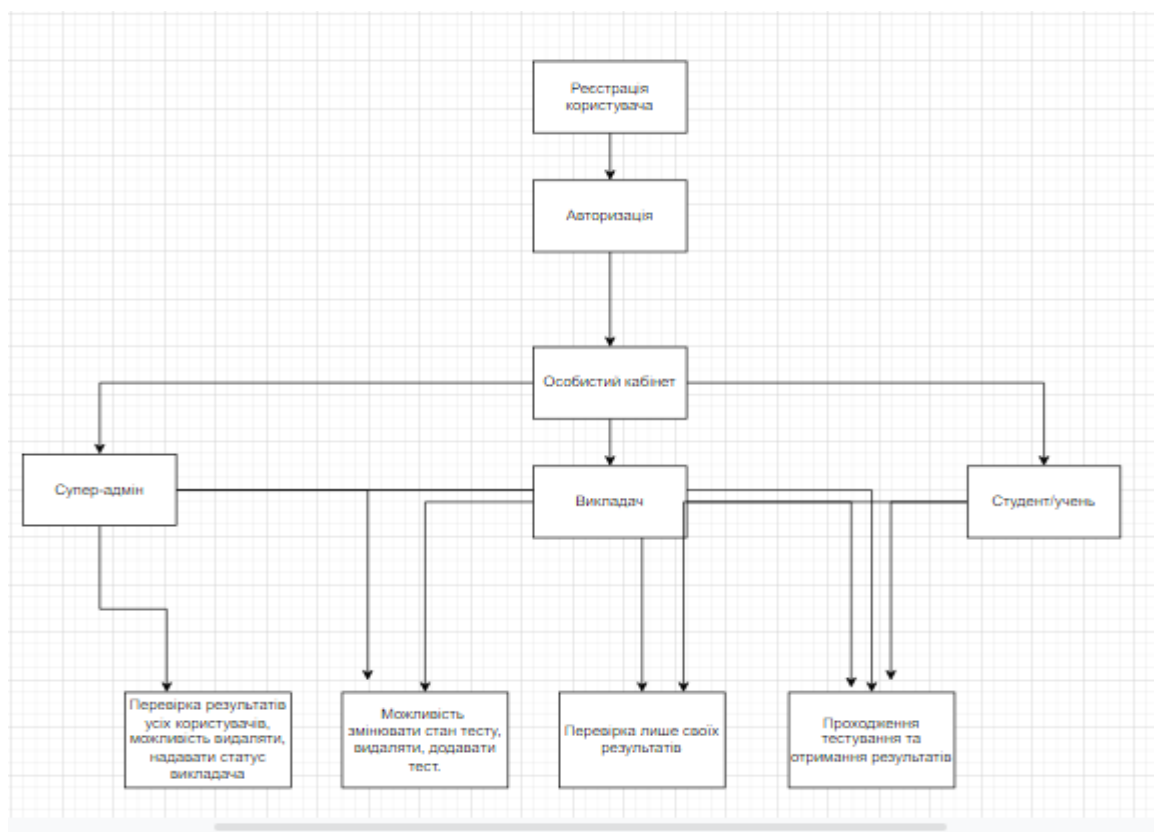


рис. 2.1 схема програми

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк. 10
		Чижмотря О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		

Візуалізація алгоритму роботи програми надає можливість оцінити та структурувати можливості, які буде мати кожен користувач.

## 2.2 Розробка функціональних алгоритмів роботи програми

Основоположним класом є клас PropertyChangedBaseImplementation, який на мою думку, є одним з найважливіших класів, які взагалі використовують у WPF. Цей клас містить подію PropertyChanged, та метод який сигналізує про зміну властивості. Саме цей метод допомагає нам сповістити інтерфейс, що у властивості відбулись зміни та потрібно відобразити це на екрані. Наслідниками цього класу є класи ViewModels, які реалізують властивості та логіку, яка відображається на екрані.

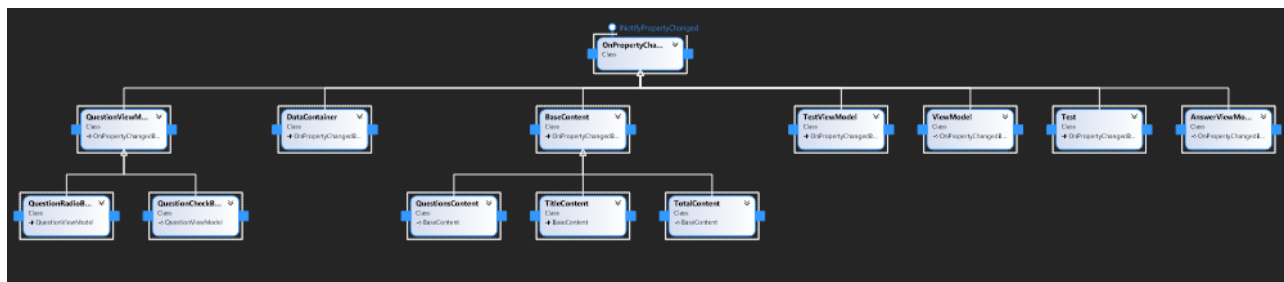


рис. 2.1 діаграма класів

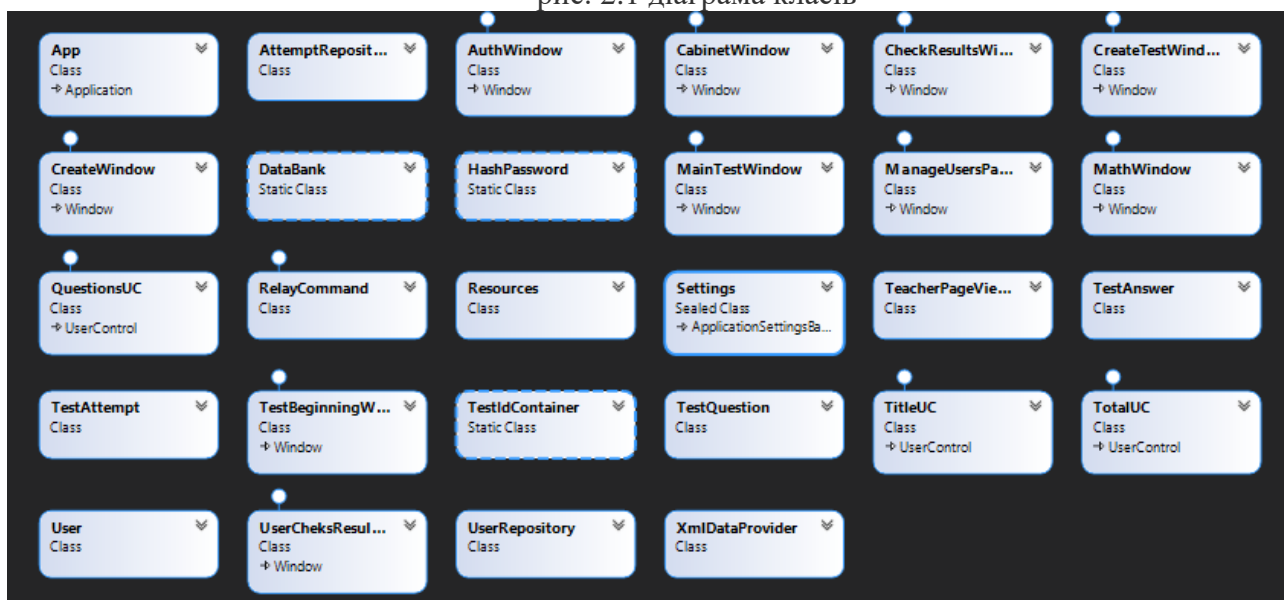


рис. 2.2 діаграма класів

На цьому скріншоті ми можемо побачити такі класи як AttemptRepository та UserRepository, ці класи допомагають нам взаємодіяти з базою даних. У класі XmlDataProvider відбувається серіалізація або десеріалізація нашого файлу Tests.xml. у якому зберігаються тести. Також тут є допоміжний клас RelayCommand, який є також одним з основоположних класів для реалізації паттерну MVVM. Також тут є класи User та Attempt, це є наші моделі.

## 2.3 Розробка програмного забезпечення

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмотря О.В				11
Змн.	Арк.	№ докум.	Підпис	Дата		

За реалізацію логіну та авторизації відповідає вікно AuthWindow, уся логіка в ньому реалізована у так званому Code-Behind. Далі дані авторизованого юзера потрапляють до статичного класу DataBank, та на основі цього у особистому кабінеті відображаються ті можливості, які є доступними цьому юзеру. За парсинг цих тестів відповідає клас XmlDataProvider. Відтак кожен юзер може розпочати тестування, за яке відповідають класи ViewModel, TestViewModel, QuestionsContent, TitleContent, TotalContent, та відповідні їм представлення у вигляді UserControls. Різні можливості викладачів та супер-адміна реалізовані за допомогою “Code-Behind” та репозиторіїв. Кожна спроба за допомогою репозиторію AttemptRepository потрапляє то таблиці Attempts у базі даних. Кожен користувач потрапляє до UserRepository, звідти у таблицю Users у базі даних.

### **Висновки до другого розділу:**

На етапі проектування програми мені довелося визначитися з структурою програми та обдумати усі проблеми з якими я міг би зіштовхнутися під час розробки. Потрібно було зробити структуру достатньо гнучкою, щоб можливо було без великих зусиль модифікувати програму, що дозволило мені після завершення реалізації основного функціоналу додавати нові можливості до гри(такі як персонаж та розблокування шапок для нього).

## **РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ**

### **3.1 Опис роботи з програмним додатком (Опис інтерфейсу)**

Після запуску гри ми бачимо можливість зареєструватись або авторизуватись.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижевотря О.В				12
Змн.	Арк.	№ докум.	Підпис	Дата		

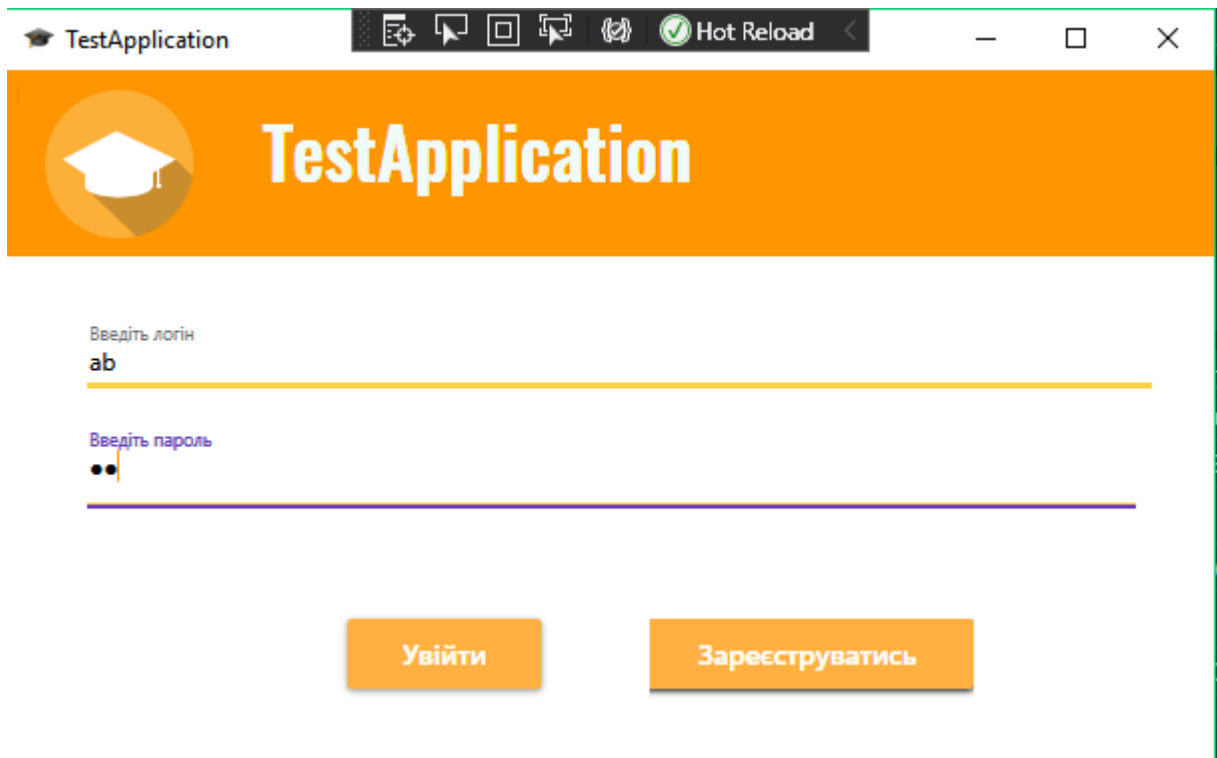


Рис. 3.1. Меню авторизації/реєстрації

Після реєстрації, бачимо діалогове вікно, яке сигналізує про успішне завершення реєстрації.

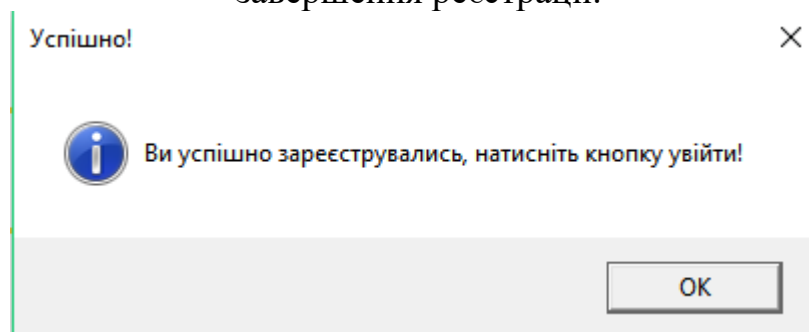


Рис. 3.2. Діалогове вікно після успішної реєстрації.

Після натиснення кнопки увійти, бачимо діалогове вікно, яке сигналізує про успішну авторизацію до особистого кабінету.

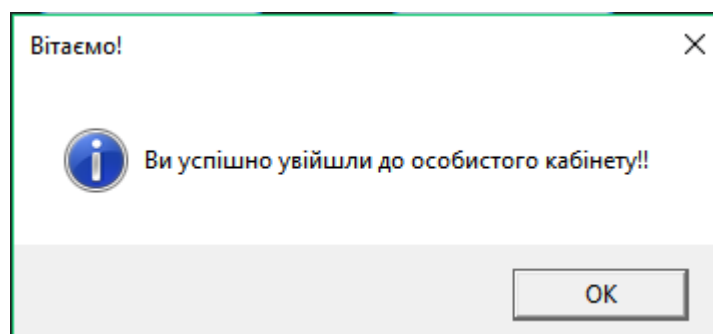


Рис. 3.3. Діалогове вікно після успішної авторизації.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Після логіну, звичайний користувач потрапляє до особистого кабінету, де у нього є можливість пройти тестування та подивитись лише свої спроби проходження тесту.

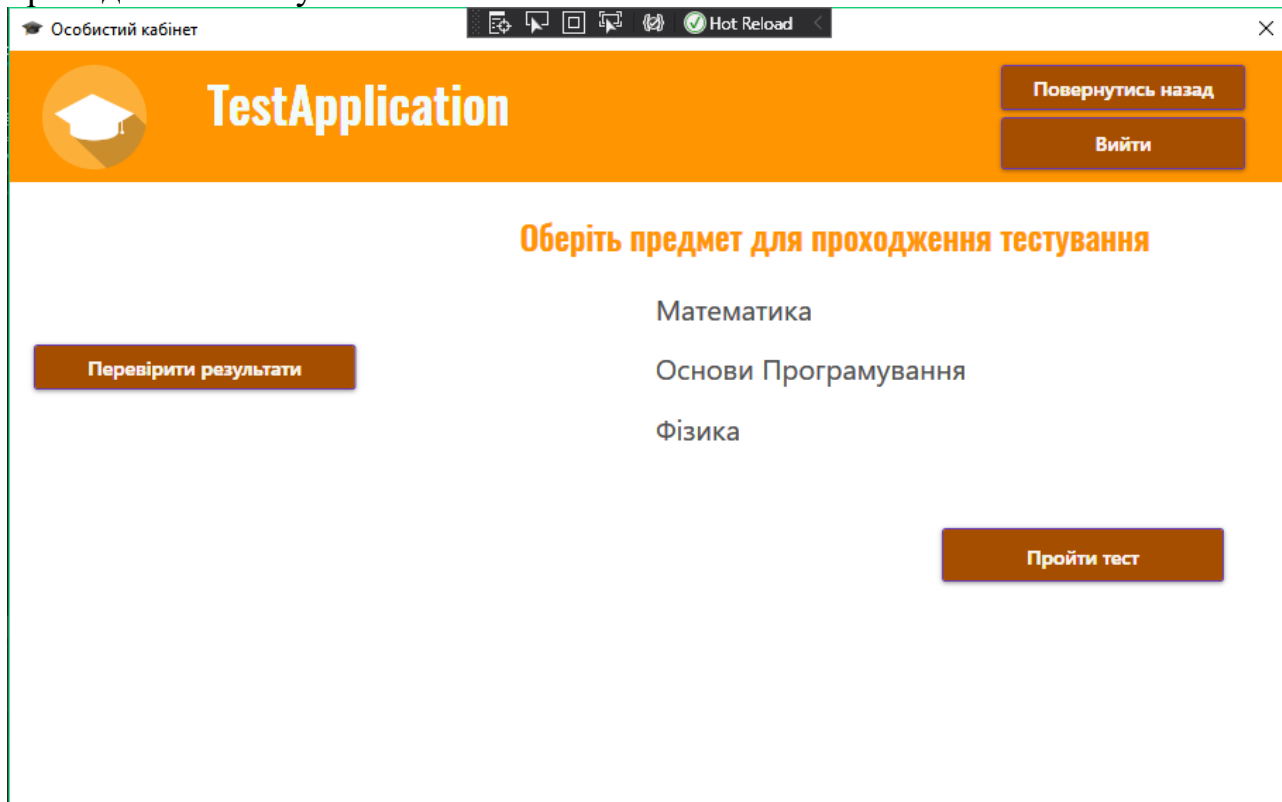


Рис. 3.4. Особистий кабінет звичайного користувача.

Користувач обрав математику, та почав проходити тест. Це сторінка початку тесту, тут є лише назва та кнопка розпочати тест.

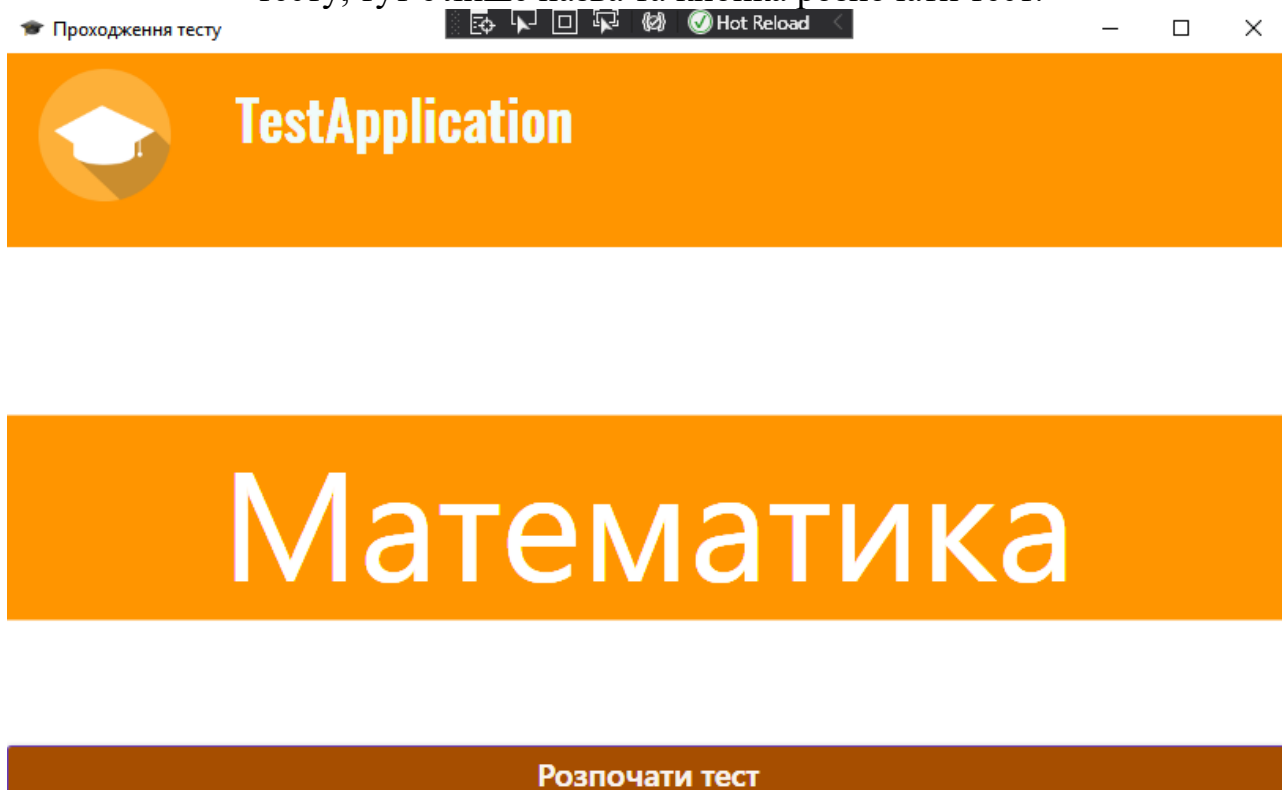


Рис. 3.5. Вікно початку тестування.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк. 14
		Чижмотря О.В				
Змн.	Арк.	№ докум.	Підпис	Дата		

Тест розпочато, у користувача є певна кількість хвилин для виконання тесту.  
Цей тип питань лише з однією правильною відповіддю.

Рис. 3.6. Вікно питань тесту.

.Тип питань з декількома правильними відповідями.

Рис. 3.7. Тестове питання ще одного виду.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувач натиснув завершити тест, на усі питання відповів правильно.

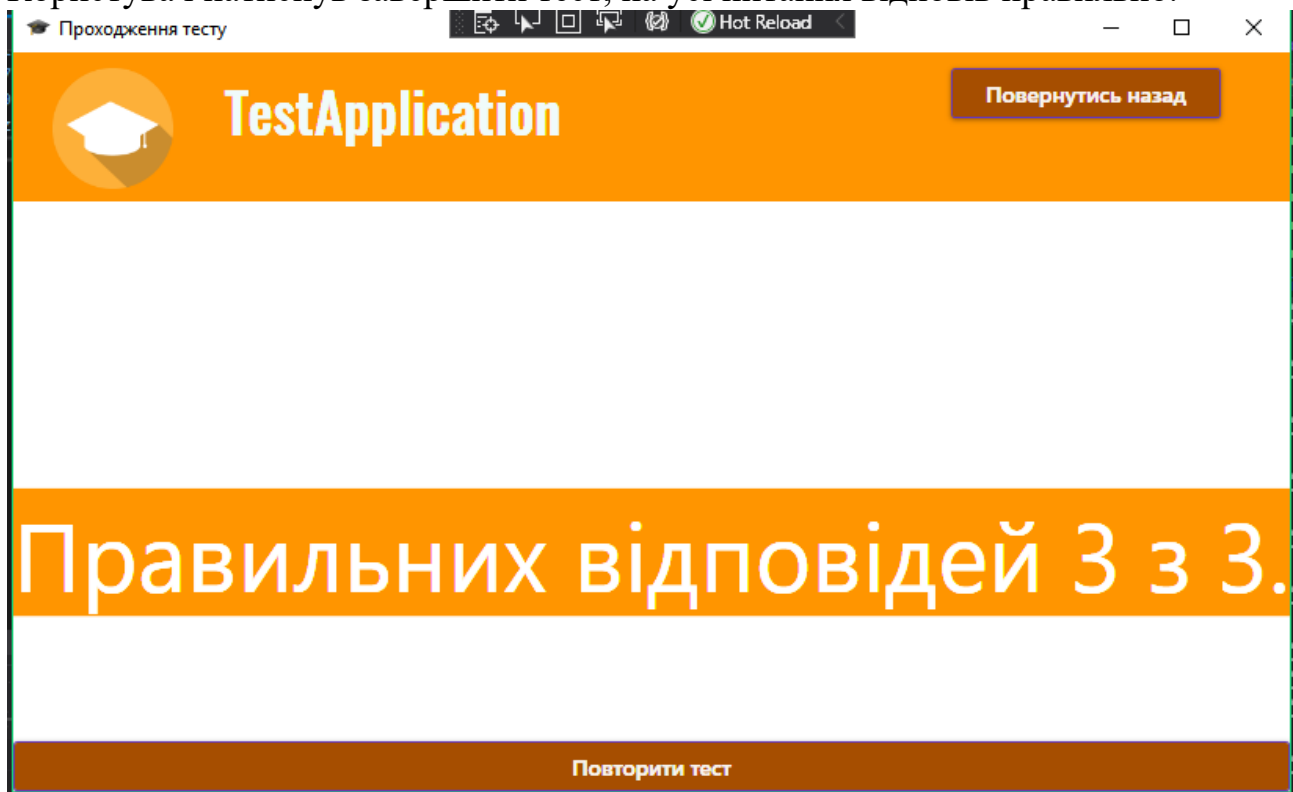


Рис. 3.8. Сторінка кінця тесту.

Тепер даний користувач зайшов у вкладку результати, та може переглянути лише свої результати.

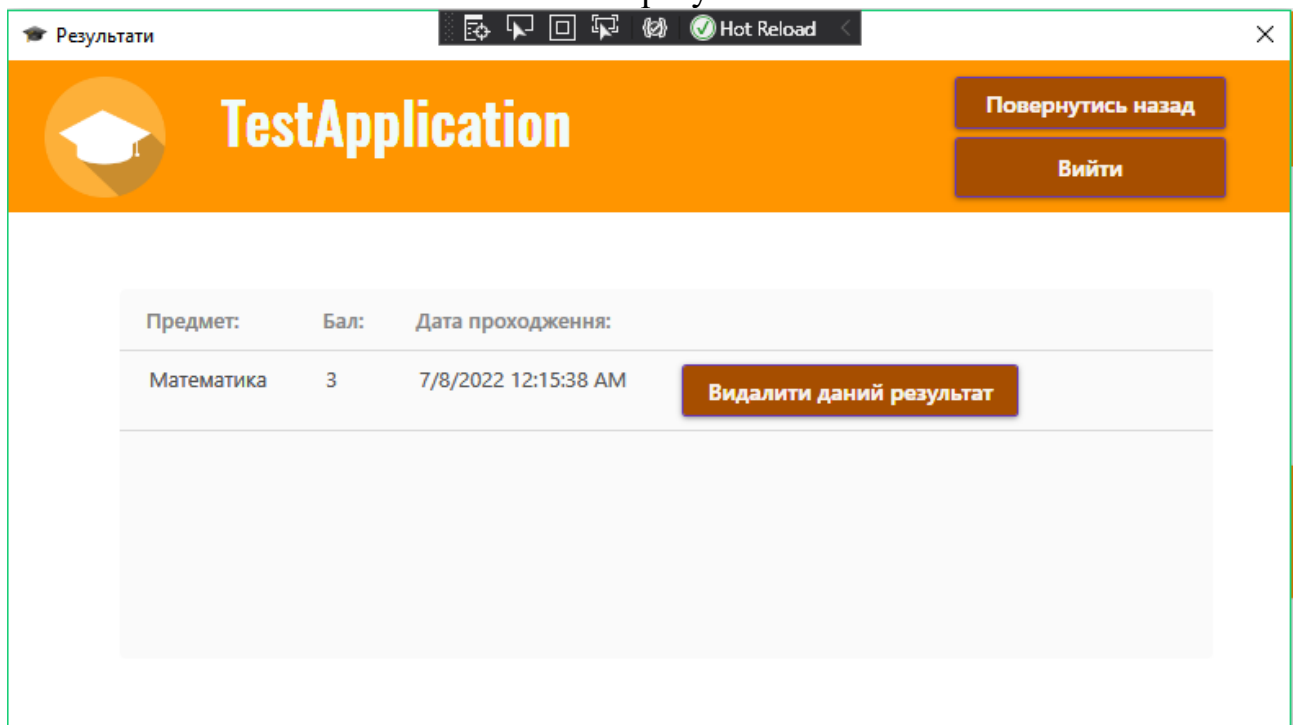


Рис. 3.9. Сторінка перевірки результатів.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В				16
Змн.	Арк.	№ докум.	Підпис	Дата		



У користувача з доступом рівня викладач, з'являється кнопка керування тестами, також він може переглядати результати усіх людей.

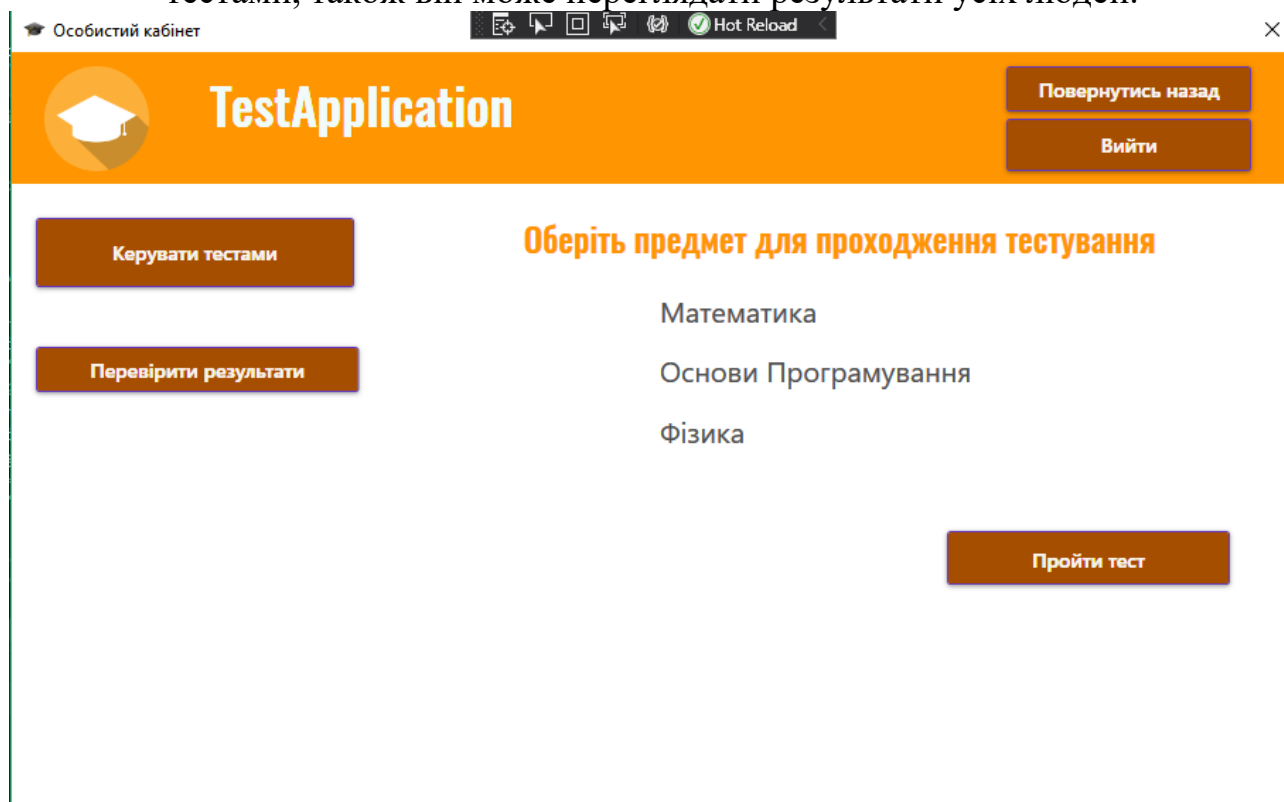


Рис. 3.10. Особистий кабінет викладача.

Ось приклад тих результатів, які доступні викладачу.

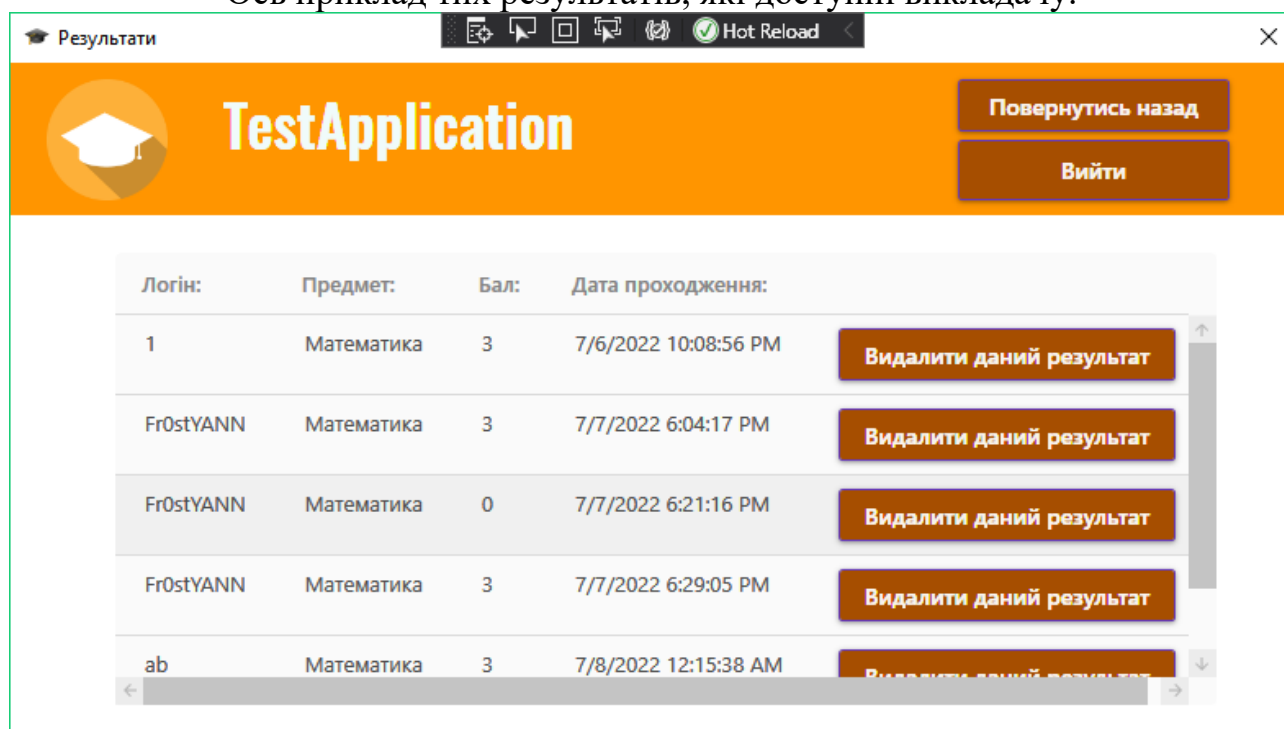


Рис. 3.11. Сторінка результатів, доступних викладачу.

Ось тут ми можемо побачити сторінку управління тестами, користувач може натиснути створити тест, та завантажити файл з тестом у форматі Xml, також він може змінити стан тесту, обравши тест та натиснувши на кнопку змінити стан тесту, також може видалити тест.

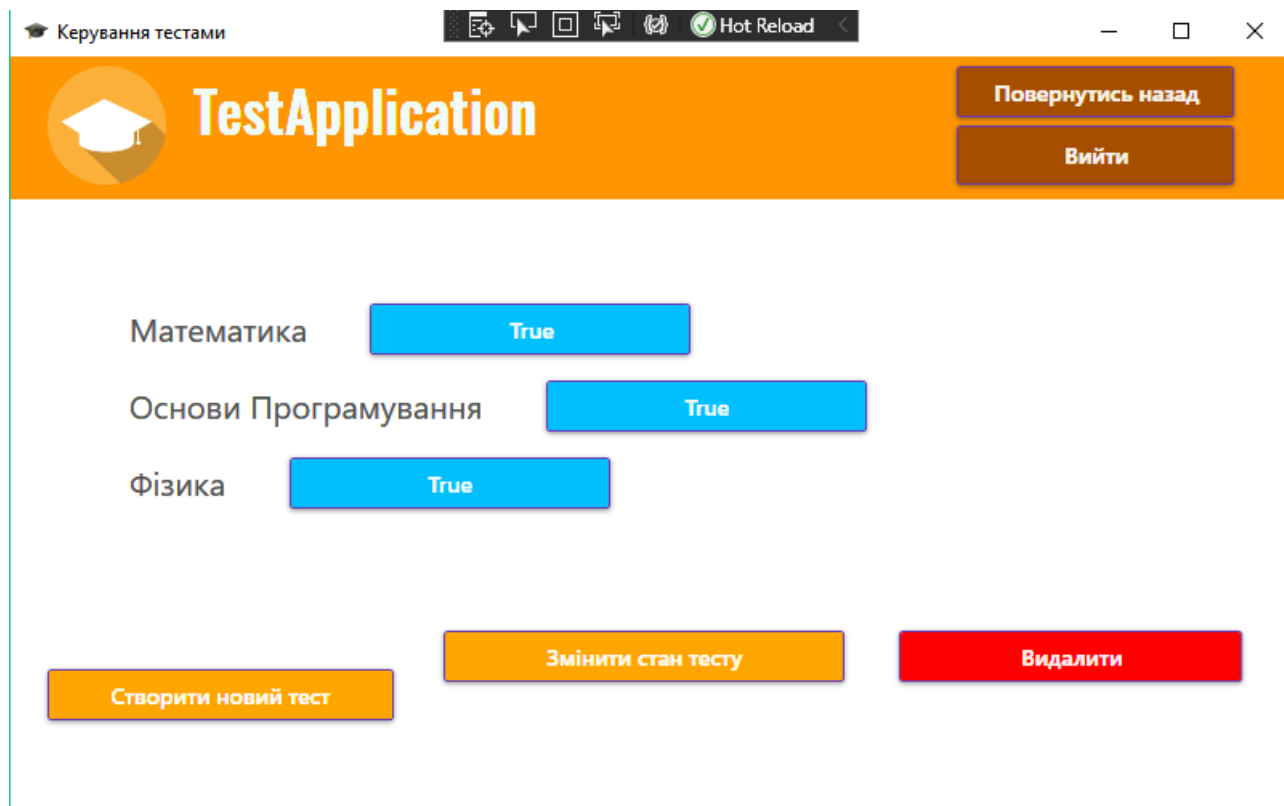


Рис. 3.12. Сторінка управління тестами.

У користувача з доступом супер-адмін з'являється ще одна кнопка, керування користувачами.

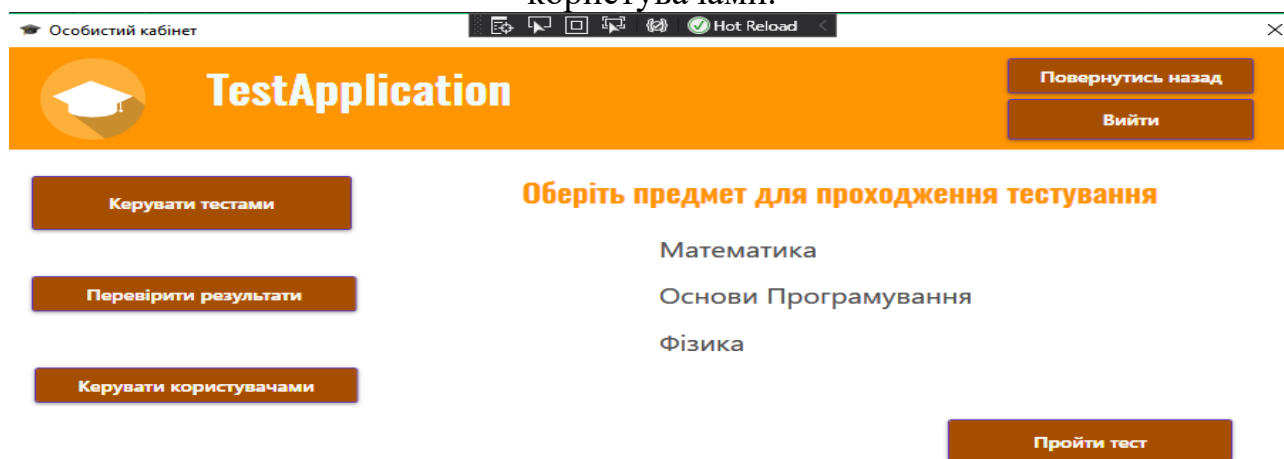


Рис. 3.13. Особистий кабінет супер-адміна.

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмотря О.В				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Тут цей користувач може видаляти користувачів, надавати доступ суперадмін та викладач.

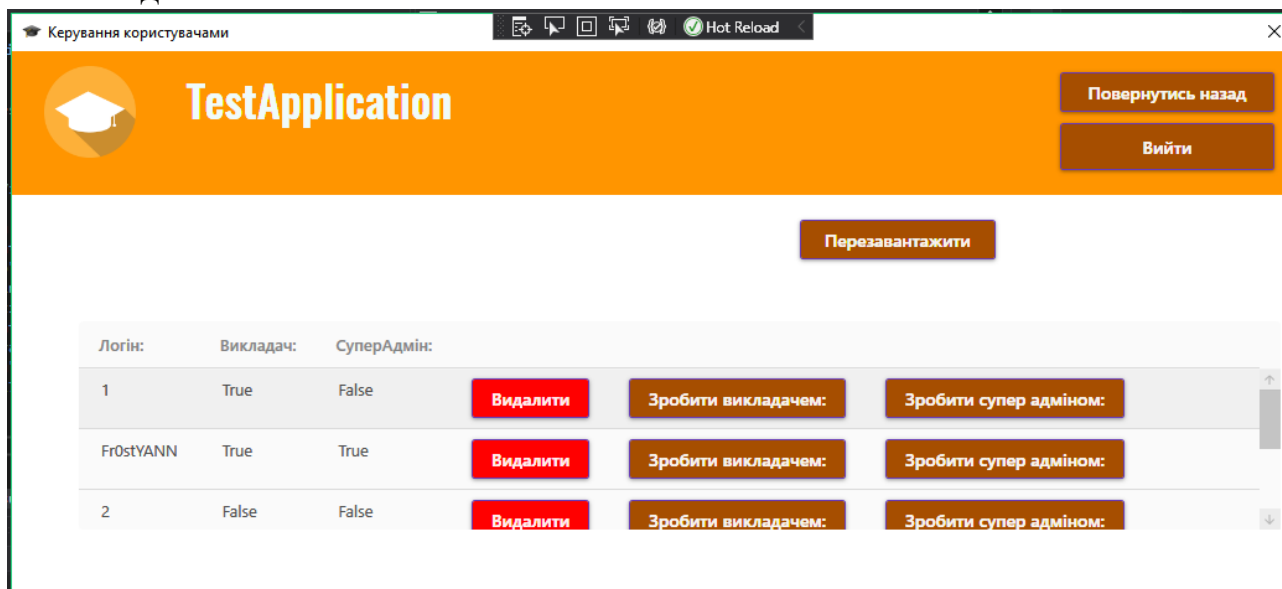


Рис. 3.14. Сторінка менеджменту юзерів.

### 3.2 Тестування роботи програмного забезпечення

Основним методом тестування програми були прописані умови перевірки вхідних даних. Програма передбачає діалогові вікна та перекриття доступу до функціоналу в разі якихось проблем.

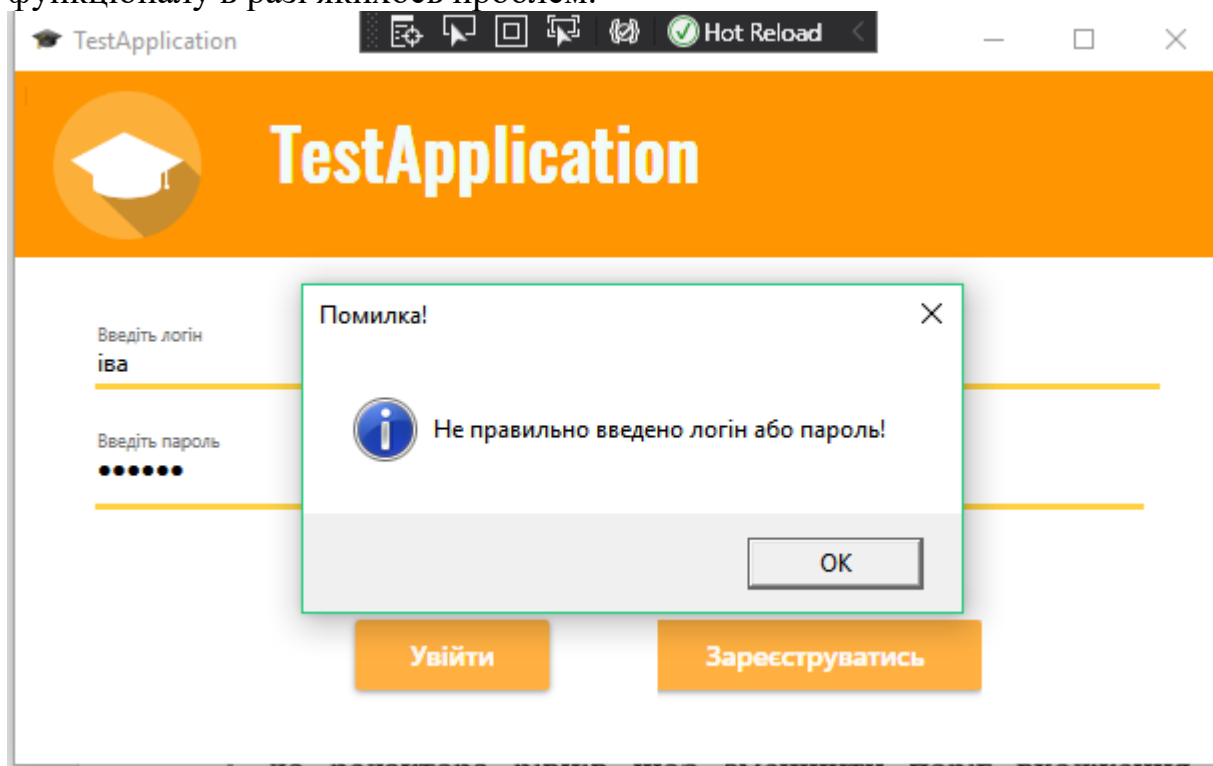


Рис. 3.15. Валідація введених паролю та логіну

### Висновки до третього розділу

Під час розробки інтерфейсу гри мне довелося уявляти як буде бачити мою програму інша людина, не знайома з структурою програми. Треба було зробити максимально очевидні елементи управління та уникнути усіх можливих помилок повідомляючи про них користувачу.

## ВИСНОВКИ

Отже, під час виконання даної курсової роботи я закріпив матеріал, який вивчався протягом 2 семестру, а саме мова програмування C# та проектування додатків Використовуючи парадигму ООП.

Я попрактикувався у аналізі власної роботи основуючись на вже існуючих аналогах. Також навчився оцінювати переваги та недоліки інших проектів щоб мати уяву про те що можна покращити під час розробки власного продукту.

Отже, дана робота демонструє що з допомогою мови програмування C# , Windows Presentation Foundation, баз даних MS SQL, можливо створити повноцінний екранний додаток для створення та проведення тестувань.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Эндрю Т. Язык Программирования C# 5.0 и платформа .NET 4.5 / Троелсен Эндрю., 2013. – 1168 с. – (6-е издание).
2. Об'єктно-орієнтоване програмування [Електронний ресурс] / Освітній портал ДУ «Житомирська політехніка» – 2020. – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/course/view.php?id=1628>
3. Технології програмування. Мова C# : навч. посібн/ В.В. Томашевський. – Житомир: ЖВІ НАУ, 2012. – 484 с.
4. Шилдт Герберт C# 4.0: полное руководство.: Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
5. Документація до бібліотеки newtonsoft[Електронний ресурс] - <https://www.newtonsoft.com/json>
6. Спосіб зберігання параметрів програми в Properties.Settings [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/271483/>
7. Сериалізація в C# [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/serialization/>
8. Complete wpf-tutorial - <https://wpf-tutorial.com/>

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижевська О.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

# Додатки

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмотря О.В				21
Змн.	Арк.	№ докум.	Підпис	Дата		

## Лістинг програми:

### Класи:

#### DataBank

```
public static class DataBank
{
    public static int UserId;
    public static string Login;
    public static string Password;
    public static bool IsTeacher;
    public static bool IsSuperAdmin;
}
```

#### DataContainer

```
public class DataContainer : OnPropertyChangedBaseImplementation
{
    private List<Test> tests;
    public List<Test> Tests { get => tests; set { tests = value; OnPropertyChanged(); }
}
}
```

#### HashPassword

```
public static class HashPassword
{
    public static string HashPasswordFunc(string password)
    {
        MD5 md5 = MD5.Create();

        byte[] b = Encoding.ASCII.GetBytes(password);
        byte[] hash = md5.ComputeHash(b);
        StringBuilder sb = new StringBuilder();
        foreach (var a in hash)
            sb.Append(a.ToString("X2"));
        return Convert.ToString(sb);
    }
}
```

#### BaseContent

```
public class BaseContent : OnPropertyChangedBaseImplementation
{
    public RelayCommand JumpCommand { get; }
    public BaseContent(ExecuteHandler execute, CanExecuteHandler canExecute = null)
    {
        JumpCommand = new RelayCommand(execute, canExecute);
    }
}
```

#### QuestionsContent

```
public class QuestionsContent : BaseContent
{
    private QuestionViewModel[] questions;
    private QuestionViewModel currentQuestion;
    private int currentQuestionIndex;
    public int currentQuestionIndexToShow;
    private RelayCommand jumpQuestionCommand;
    private int seconds = 0;
```

		Чижмотря О.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

```

private int minutes = 0;
private string timerString;
private bool isEnabled = true;
DispatcherTimer timer = new DispatcherTimer();
public string MinutesLimitToShow { get; set; }
public bool IsEnabled { get => isEnabled; set { isEnabled = value;
OnPropertyChanged(); } }
public string TimerString { get => timerString; set { timerString = value;
OnPropertyChanged(); } }
public int Seconds { get => seconds; set { seconds = value; OnPropertyChanged(); }
}
public int Minutes { get => minutes; set { minutes = value; OnPropertyChanged(); }
}
public QuestionViewModel[] Questions { get => questions; set { questions = value;
OnPropertyChanged(); currentQuestionIndex = -1; JumpQuestionMethod(1); } }
public QuestionViewModel CurrentQuestion { get => currentQuestion; set {
currentQuestion = value; OnPropertyChanged(); } }
public int MinutesLimit { get; set; }
public int CurrQuestionIndex { get => currentQuestionIndex; set {
currentQuestionIndex = value; OnPropertyChanged(); } }
public int CurrentQuestionIndexToShow { get => currentQuestionIndexToShow; set {
currentQuestionIndexToShow = value; OnPropertyChanged(); } }
public QuestionsContent(ExecuteHandler execute, int MinutesLimit, CanExecuteHandler
canExecute = null)
: base(execute, canExecute)
{
this.MinutesLimit = MinutesLimit;
MinutesLimitToShow = $"Для проходження тесту ви маєте {MinutesLimit}
хвилин(и)";
timer.Interval = TimeSpan.FromSeconds(1);
timer.Tick += timer_Tick;
timer.Start();
}
void timer_Tick(object sender, EventArgs e)
{
if(Minutes >= MinutesLimit)
{
Seconds = 0;
timer.Stop();
isEnabled = false;
}
if (Seconds >= 59)
{
Minutes = 0;
Minutes++;
}
else
{
Seconds++;
}
if (Seconds < 10)
{
TimerString = $"0{Minutes}:0{Seconds}";
}
else
{
TimerString = $"0{Minutes}:{Seconds}";
}
}
public RelayCommand JumpQuestionCommand => jumpQuestionCommand ??
(jumpQuestionCommand = new RelayCommand(JumpQuestionMethod, JumpQuestionCanMethod));
private bool JumpQuestionCanMethod(object parameter)
=> parameter != null
&& int.TryParse(parameter.ToString(), out int parInt)
&& currentQuestionIndex + parInt >= 0 && currentQuestionIndex +
parInt < Questions.Length;

private void JumpQuestionMethod(object parameter)
{
int newIndex = currentQuestionIndex + int.Parse(parameter.ToString());

```

Грибан В.В.

Чижмоторя О.В.

Змн.

Арк.

№ докум.

Підпис

Дата

ДУ «Житомирська політехніка». 22.121.08.000 - ПЗ

Арк.

23

```

        if (newIndex != currentQuestionIndex)
        {
            currentQuestionIndex = newIndex;
            currentQuestionIndexToShow = newIndex + 1;
            CurrentQuestion = Questions[currentQuestionIndex];
        }
    }
}

```

## TitleContent

```

private string testName;

public string TestName { get => testName; set { testName = value;
OnPropertyChanged(); } }
public TitleContent(ExecuteHandler execute, CanExecuteHandler canExecute = null) :
base(execute, canExecute)
{
}

```

## TotalContent

```

public class TotalContent : BaseContent
{
    private int countRight;
    private int countTotal;
    public int CountRight { get => countRight; set { countRight = value;
OnPropertyChanged(); } }
    public int CountTotal { get => countTotal; set { countTotal = value;
OnPropertyChanged(); } }
    public TotalContent(ExecuteHandler execute, CanExecuteHandler canExecute = null) :
base(execute, canExecute) {
}
}

```

## OnPropertyChangedBaseImplementation

```

public class OnPropertyChangedBaseImplementation : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;
    public void OnPropertyChanged([CallerMemberName] string propertyName = "") =>
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}

```

## RelayCommand

```

#region Делегати для команд
public delegate void ExecuteHandler(object parameter);
public delegate bool CanExecuteHandler(object parameter);
#endregion
public class RelayCommand : ICommand
{
    private readonly CanExecuteHandler canExecute;
    private readonly ExecuteHandler onExecute;

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

```

public RelayCommand(ExecuteHandler execute, CanExecuteHandler canExecute = null)

Чижмотря О.В

ДУ «Житомирська політехніка».22.121.08.000 - ПЗ

Арк.

24

Змн. Арк. № докум. Підпис Дата



```

{
    this.onExecute = execute;
    this.canExecute = canExecute;
}
public bool CanExecute(object parameter) => canExecute == null ? true :
canExecute.Invoke(parameter);
public void Execute(object parameter) => onExecute?.Invoke(parameter);
}

```

## TestAttempt

```

public class TestAttempt
{
    public int AttemptId { get; set; }
    public DateTime AttemptDate { get; set; }
    public string UserLogin { get; set; }
    public string TestName { get; set; }
    public int Points { get; set; }
}

```

## User

```

public class User
{
    public int Id { get; set; }
    public string Login { get; set; }
    public string Password { get; set; }
    public bool IsTeacher { get; set; }
    public bool IsSuperAdmin { get; set; }
}

```

## AttemptRepository

```

public class AttemptRepository
{
    string ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB; Initial
Catalog=TestApp; Integrated Security=True";
    public void SetPointsToAttemptTable(string userLogin, string testName, int points)
    {
        var sqlQuery = @"INSERT INTO Attempts (TestName, UserLogin, Points,
AttemptDate) VALUES(@TestName, @UserLogin, @Points, @AttemptDate)";
        using (IDbConnection db = new SqlConnection(ConnectionString))
        {
            db.Execute(sqlQuery, new { UserLogin = userLogin, TestName = testName,
Points = points, AttemptDate = DateTime.Now });
        }
    }
    public List<TestAttempt> GetAllResults()
    {
        var sqlQuery = @"SELECT * FROM Attempts";
        using (IDbConnection db = new SqlConnection(ConnectionString))
        {
            var testAttempts = db.Query<TestAttempt>(sqlQuery);
            return testAttempts.ToList();
        }
    }
    public void DeleteAttempt(int attemptId)
    {
        var sqlQuery = "DELETE FROM Attempts WHERE AttemptId = @AttemptId";
        using (IDbConnection db = new SqlConnection(ConnectionString))
        {
            db.Execute(sqlQuery, new { AttemptId = attemptId });
        }
    }
    public void MakeUserTeacher(int userId)

```

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижевська О.В.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        var sqlQuery = "UPDATE Users SET IsTeacher = 1 WHERE Id = @Id";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            db.Execute(sqlQuery, new { Id = userId });
        }
    }
    public void MakeUserSuperAdmin(int userId)
    {
        var sqlQuery = "UPDATE Users SET IsSuperAdmin = 1, IsTeacher = 1 WHERE Id =
@Id";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            db.Execute(sqlQuery, new { Id = userId });
        }
    }
}

```

## UserRepository

```

public class UserRepository
{
    string connectionString = @"Data Source=(localdb)\MSSQLLocalDB; Initial
Catalog=TestApp; Integrated Security=True";
    public bool CheckUserExists(string password, string login)
    {
        var sqlQuery = "Select * from Users where Login = @Login and Password =
@Password";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            var res = db.QueryFirstOrDefault(sqlQuery, new { Login = login, Password =
password });
            if(res == null)
            {
                return false;
            }
        }
        return true;
    }
    public User GetUser(string login)
    {
        var sqlQuery = "Select * from Users where Login = @Login";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            var res = db.QueryFirstOrDefault<User>(sqlQuery, new { Login = login });
            return res;
        }
    }
    public List<User> GetAllUsers()
    {
        var sqlQuery = "SELECT * FROM Users";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            var result = db.Query<User>(sqlQuery);
            return result.ToList();
        }
    }
    public int RegisterUser(string password, string login)
    {
        var sqlQuery = "Insert into Users (Login, Password) Values (@Login,@Password)";
        using (IDbConnection db = new SqlConnection(connectionString))
        {
            db.Execute(sqlQuery, new { Login = login, Password = password });
        }
        return 1;
    }
    public void DeleteUser(int userId)
    {
        var sqlQuery = "DELETE FROM Users WHERE Id = @Id";
        using (IDbConnection db = new SqlConnection(connectionString))
    }
}

```

		Григор'ян В.В.			ДУ «Житомирська політехніка». 22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        db.Execute(sqlQuery, new { Id = userId });
    }
}

```

## XmlRepository

```

public class XmlRepository
{
    XmlSerializer xmlSerializer = new XmlSerializer(typeof(DataContainer));
    string FileString = @"C:\Users\Phoenix\Desktop\Coursework-main\WpfApp1\Test.xml";
    public List<Test> GetAllTests()
    {
        DataContainer data = new DataContainer();
        using (FileStream fs = new FileStream(FileString, FileMode.OpenOrCreate))
        {
            data = (DataContainer)xmlSerializer.Deserialize(fs);
        }
        return data.Tests;
    }
    public List<Test> SetSomeChanges(List<Test> Tests)
    {
        DataContainer data = new DataContainer();
        data.Tests = Tests;
        using (FileStream fs = new FileStream(FileString, FileMode.Truncate))
        {
            xmlSerializer.Serialize(fs, data);
        }
        return data.Tests;
    }
}

```

## Test

```

[Serializable()]
public partial class Test : OnPropertyChangedBaseImplementation
{
    private string testName;
    private bool isActive;
    public string TestName { get => testName; set { testName = value;
OnPropertyChanged(); } }
    public bool IsActive { get => isActive; set { isActive = value;
OnPropertyChanged(); } }
    public int TestId { get; set; }
    public int MinutesLimit { get; set; }
    [XmlAttribute("Question", IsNullable = false)]
    public List<TestQuestion> Questions { get; set; }
}

```

## TestAnswer

```

[Serializable()]
public partial class TestAnswer
{
    [XmlAttribute()]
    public bool IsRight { get; set; }
    [XmlText()]
    public string AnswerText { get; set; }
}

```

## TestQuestion

```

[Serializable()]
public partial class TestQuestion
{
    public string QuestionText { get; set; }
}

```

		[XmlAttribute()]			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижмоторя О.В				27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public bool IsOnlyOne { get; set; }
[XmlAttribute("Answer", IsNullable = false)]
public List<TestAnswer> Answers { get; set; }
}

```

## AnswerViewModel

```

public class AnswerViewModel : OnPropertyChangedBaseImplementation
{
    private readonly TestAnswer testAnswer;
    private bool isRightView;
    public AnswerViewModel(TestAnswer _testAnswer)
    {
        testAnswer = _testAnswer;
    }
    public string Text => testAnswer.AnswerText;
    public bool IsRightView { get => isRightView; set { isRightView = value;
OnPropertyChanged();} }
    public bool Value => IsRightView == testAnswer.IsRight;
}

```

## QuestionCheckBoxViewModel

```

public class QuestionCheckBoxViewModel : QuestionViewModel
{
    public QuestionCheckBoxViewModel(TestQuestion testQuestion) : base(testQuestion)
    {
    }
}

```

## QuestionRadioButtonViewModel

```

public class QuestionRadioButtonViewModel : QuestionViewModel
{
    public QuestionRadioButtonViewModel(TestQuestion testQuestion) : base(testQuestion)
    {
    }
}

```

## QuestionViewModel

```

public class QuestionViewModel : OnPropertyChangedBaseImplementation
{
    protected static readonly Random rand = new Random();
    protected readonly TestQuestion TestQuestion;
    public AnswerViewModel[] Answers { get; }
    public QuestionViewModel(TestQuestion testQuestion)
    {
        TestQuestion = testQuestion;
        Answers = TestQuestion.Answers.Select(ans => new AnswerViewModel(ans))
            .OrderBy(x => rand.Next()).ToArray();
    }
    public string Text => TestQuestion.QuestionText;
    public bool Value => Answers.All(ans => ans.Value);
    public static QuestionViewModel Create(TestQuestion testQuestion)
    {
        if (testQuestion.IsOnlyOne)
            return new QuestionRadioButtonViewModel(testQuestion);
        return new QuestionCheckBoxViewModel(testQuestion);
    }
}

```

## TestViewModel

```

public class TestViewModel : OnPropertyChangedBaseImplementation

```

Ришан В.В.					Арк.
Чижмоторя О.В.					ДУ «Житомирська політехніка».22.121.08.000 - ПЗ
Змн.	Арк.	№ докум.	Підпис	Дата	28

```

{
    private static readonly Random rand = new Random();
    private readonly Test Test;
    public QuestionViewModel[] Questions { get; }
    public TestViewModel(Test test)
    {
        Test = test;
        Questions = Test.Questions.Select(q => QuestionViewModel.Create(q))
            .OrderBy(x => rand.Next()).ToArray();
    }
    public string TestName => Test.TestName;
    public int RightCount() => Questions.Count(q => q.Value);
}

```

## ViewModel

```

public class ViewModel : OnPropertyChangedBaseImplementation
{
    private readonly Test test;
    public string TestName => test.TestName;
    public TestViewModel TestView { get; private set; }
    private BaseContent _content;
    public BaseContent Content { get => _content; set { _content = value;
OnPropertyChanged(); } }
    public ViewModel(int testId)
    {
        XmlRepository xmlDataProvider = new XmlRepository();
        test = xmlDataProvider.GetAllTests().Where(test => test.TestId ==
testId).FirstOrDefault();
        TotalMethod(null);
    }
    private void TitleMethod(object parameter)
    {
        Content = new QuestionsContent(QuestionsMethod, test.MinutesLimit) { Questions
= TestView.Questions };
    }
    private void QuestionsMethod(object parameter)
    {
        int CountRight = TestView.RightCount();
        Content = new TotalContent(TotalMethod)
        {
            CountRight = CountRight,
            CountTotal = TestView.Questions.Length
        };
        AttemptRepository attemptRepository = new AttemptRepository();
        attemptRepository.SetPointsToAttemptTable(DataBank.Login, test.TestName,
CountRight);
    }
    private void TotalMethod(object parameter)
    {
        TestView = new TestViewModel(test);
        Content = new TitleContent(TitleMethod) { TestName = TestView.TestName };
    }
}

```

		Грибан В.В.			ДУ «Житомирська політехніка».22.121.08.000 - ПЗ	Арк.
		Чижевська О.В				29
Змн.	Арк.	№ докум.	Підпис	Дата		