

**-НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

**Індивідуальний дослідницький проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Програмний застосунок тестування знань студентів»**

Виконав:

студент ІV курсу, групи ІА-82
Афонін Артем Павлович

Керівник:

Ст. викладач каф. ІСТ,
Моргаль Олег Михайлович

Засвідчую, що у цьому проєкті немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАВДАННЯ
на дипломний проєкт студенту
Афоніну Артему Павловичу

1. Тема проєкту «Програмний застосунок тестування знань студентів», керівник проєкту Моргаль Олег Михайлович старший викладач
2. Термін подання студентом проєкту: 15 червня 2022 року
3. Вихідні дані до проєкту: Система керування версіями, мова програмування C#, середовище програмування Visual Studio 2019, обраний фреймворк для розробки – .NET framework 4.7, СУБД – Access.
4. Зміст пояснювальної записки: Аналіз предметної області та існуючих рішень, аналіз вимог та моделювання програмного забезпечення, аналіз та вибір технологій, розгортання та реалізація застосунку.
5. Перелік графічного матеріалу: Use-case діаграма, ERD діаграма, діаграма класу BLL, діаграма класів домена, діаграма інтерфейсу програми, діаграма класів DAO.
6. Дата видачі завдання 1 грудня 2021 року

Календарний план

№ з/п	Назва етапів виконання дослідницького проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд області дослідження	09.05.2022	
2	Аналіз існуючих рішень	16.05.2022	
3	Попередня підготовка текстових матеріалів	23.05.2022	
4	Розробка графічних матеріалів	30.05.2022	
5	Програмна реалізація системи	13.06.2022	
6	Оформлення документації дослідницького проєкту	19.06.2022	

Студент

Артем АФОНІН

Керівник

Олег МОРГАЛЬ

АНОТАЦІЯ

Афонін А.П. Програмний застосунок тестування знань студентів. КПІ ім. Ігоря Сікорського, Київ, 2022.

Проект містить 69 с. тексту, 38 рисунків, 8 таблиць, посилання на 20 літературних джерел, додаток та 4 конструкторські документи.

Ключові слова: користувач, програма, дисципліна, тестування, запити, діаграма.

Об'єктом розробки є створення програмного забезпечення тестування знань студентів.

Мета розробки – є розробка програмного забезпечення тестування знань студентів на .NET 4.7 в середовищі Visual Studio 2019, яке дозволить створювати тести із різних дисциплін та перевірки знань студентів.

У дипломному проєкті проведено аналіз аналогічних систем, що дасть можливість розробити якісний продукт. На базі результатів було розроблено систему тестування знань студентів з використанням технологій .NET та бази даних Access.

Отримані результати можуть бути корисними при створенні аналогічних чи подібних систем.

SUMMARY

Afonin A.P. Software application for testing students knowledge. Igor Sikorsky KPI, Kyiv, 2022.

The project contains 69 pages. text, 38 figures, 8 tables, references to 20 literature sources, appendix and 4 design documents.

Keywords: user, program, .discipline, testing, queries, diagram.

The object of development is to create software for testing students' knowledge.

The purpose of the development is to develop software for testing students' knowledge on .NET 4.7 in Visual Studio 2019, which will allow you to create tests in various disciplines and test students' knowledge.

The diploma project analyzes similar systems, which will allow to develop a quality product. Based on the results, a system of testing students' knowledge using .NET technologies and the Access database was developed.

The results obtained can be useful in creating similar or similar systems.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA82.010БАК.003 ПЗ	Пояснювальна записка	69		
6						
7	A3	IA82.010БАК.003 Д1	Програмний застосунок	1		
8			тестування знань студентів			
9			Use-case діаграма			
10						
11	A3	IA82.010БАК.003 Д2	Програмний застосунок	1		
12			тестування знань студентів			
13			ERD діаграма			
14						
15	A3	IA82.010БАК.003 Д3	Програмний застосунок	1		
16			тестування знань студентів			
17			діаграма інтерфейсу			
18						
19	A3	IA82.010БАК.003 Д4	Програмний застосунок	1		
20			тестування знань студентів			
21			діаграма класів домену			
22						
23						
24						
25						
26						
27						
28						
				IA82.010БАК.003 ТП		
Зм.	Аркуш	№ докум.	Підпис	Дата		
Розроб.		Афонін А.П			Літ.	Аркуш
Керівн.		Моргаль О.М.			Т	Аркушів
					1	1
Затв.					КПІ ім. Ігоря Сікорського ФІОТ Група ІА-82	
			Програмний застосунок тестування знань студентів Відомість проекту			

Пояснювальна записка
до індивідуального дослідницького проєкту
на тему: «Програмний застосунок тестування знань
студентів»

Київ – 2022 року

ЗМІСТ

ВСТУП	7
1 АНАЛІТИЧНИЙ ОГЛЯД.....	9
1.1 Аналіз об’єкту автоматизації.....	9
1.2 Приклади аналогічних програмних розробок.....	12
1.2.1 Пакет програм SunRav.....	12
1.2.2 Система тестування INDIGO	13
1.2.3 Конструктор тестів Keepsoft	16
1.3 Постановка задачі	17
1.4 Висновок до розділу	19
2 РОЗРОБКА	20
2.1 Аналіз вимог. Use-case діаграми. Основні прецеденти.....	20
2.2 Архітектура проекту	27
2.3 Особливості розробки бази даних. ERD діаграма з описанням сутностей	28
2.4 Особливості розробки рівня BLL.....	31
2.5 Особливість реалізації бізнес логіки – діаграма домена.....	32
2.6 Особливості розробки рівня UI	33
2.7 Особливості розробки класів для доступу до даних бази даних (DAL)	34
2.8 Висновок до розділу	35
3 РЕАЛІЗАЦІЯ.....	37
3.1 Вибір технологій	37
3.2 Середовище розробки Visual Studio.....	40

					IA82.010БАК.003 ПЗ					
Змі	Аркуш	№ докум.	Підп	Дата	Програмний застосунок тестування знань студентів			Літ	Аркуш	Аркушів
Розроб	Афонін А.П.							Т	2	69
Керівн	Моргаль О.М.				Відомість проекту			КПІ ім. Ігоря Сікорського ФІОТ Група ІА-82		

3.3 Переваги та недоліки середовища розробки	43
3.4 Розробка програмних модулів системи	45
3.5 Результати функціонального тестування розробленого додатку	55
3.6 Інструкція користувачеві програми	58
3.6.1 Опис процедури розгортання програмного продукту, створеного на платформі .NET	58
3.6.2 Використання програмного продукту	59
3.7 Висновок до розділу	67
ВИСНОВКИ.....	68
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТОК А.....	72
Лістинг програми	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ADGO.NET – ActiveX Data Object для .NET

API – Application Programme Interface

BLL – business logic layer

CIL – Common Intermediate Language

CLR – Common Language Runtime

DAL – Data Access Layer

DB - Data Base

DI - Dependency Injection

ER - модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків

ERD – Entity Relationship Diagram

IDE - комплексне програмне рішення для розробки програмного забезпечення

IL – Intermediate Language

JIT – Just In Time

OLE — технологія зв'язування та впровадження об'єктів в інші документи та об'єкти, розроблена корпорацією Microsoft

SQL – Structured Query Language

UI - інтерфейсу користувача

UI – User Interface

VS - Visual Studio

WCF – Windows Communication Foundation

WF – Windows Forms

СУБД – система управління базою даних

ООП - об'єктно-орієнтоване програмування.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		4

ВСТУП

Значення інформації та питома вага інформаційних послуг у сучасному суспільному житті різко зросли. Можна сказати, що головну роль у процесі інформатизації відіграє власне інформація, а сама інформація не створює матеріальної цінності. Під інформацією (із загальної позиції) ми маємо на увазі інформацію про фактичні дані та сукупність знань про взаємозв'язки між ними, способи, за допомогою яких суспільство може дізнатися про себе та функціонувати як ціле. Природно припустити, що інформація має бути науково обґрунтованою, доступною з точки зору її здатності здобувати, розуміти й засвоювати; дані, з яких отримують інформацію, мають бути об'ємними та відповідними сучасному стану науки.

Інформаційне середовище включає безліч інформаційних об'єктів і зв'язків між ними, засобів і прийомів збору, накопичення, передачі, обробки, виробництва і поширення інформації, знання про себе, організаційно-правові структури, що підтримують інформаційний процес. Суспільство, створюють інформаційне середовище, в якому можна функціонувати, змінюватися, удосконалювати його. Сучасні наукові дослідження показали, що вдосконалення соціального інформаційного середовища викликало формування прогресивної тенденції розвитку продуктивних сил, а процес інтелектуалізації в усіх сферах життя суспільства, у тому числі в освіті, змінив структуру суспільних відносин і відносини.

Одним із пріоритетних напрямків процесу інформатизації в сучасному суспільстві є інформатизація освіти – впровадження нових інформаційних технологій в освітню систему. Це дозволить:

- удосконалення механізмів управління освітньою системою на основі використання автоматизованих баз науково-педагогічної інформації та інформаційно-методичні матеріали та комунікаційні мережі;

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		5

– удосконалити методи та стратегії відбору змісту, методів та організаційних форм навчання, які відповідають індивідуальним цілям студентів в умовах інформізації сучасного суспільства;

– налагодити організовану систему навчання, спрямовану на розвиток інтелектуального потенціалу студентів, формування навичок здобуття знань, проведення інформаційної освіти, експериментально-дослідницької діяльності та різних видів самостійної діяльності з обробки інформації;

– створення та використання комп'ютеризованих систем тестування, контролю та оцінювання.

У дипломній роботі розглянуто один аспект процесу інформатизації освіти – створення та використання на практиці форми навчання, що використовує автоматизоване тестування систем знань і вмінь. Досліджуються можливості нових інформаційних технологій, необхідні умови їх успішного використання, розглядається та аналізується прикладне програмне забезпечення, необхідне для створення та подальшого використання комп'ютерних навчальних систем (КНС).

Використання такої системи дозволить вирішити такі питання:

– зменшити напруженість процедури контролю знань під час роботи з комп'ютером;

– мати можливість освоювати та закріплювати практичний матеріал під час роботи з програмою в режимі тренування;

– викладач матиме економію часу у зв'язку з автоматизацією процесу контролю знань.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз об'єкту автоматизації

Для активізації оволодіння новими видами діяльності, знаннями та вміннями щорічно в галузі освіти з'являються сотні комп'ютерних навчальних систем (КНС), які з успіхом застосовуються як засіб навчання та надають допомогу і у викладанні, і у вивченні того чи іншого предмета.

Це досягається за рахунок швидких обчислень, візуального відображення та збереження результатів, введення та швидкості інформації, активізації наочного мислення та зорової пам'яті учнів, можливості виконання нетрадиційних завдань та швидкого їх виконання. Комп'ютери можна успішно використовувати на всіх етапах навчальної програми: вони суттєво впливають на контрольні-оцінювальні функції, надають йому цікавого характеру та сприяють активізації навчально-пізнавальної діяльності учнів.

Комп'ютерні технології дозволяють вивести викладений матеріал на якісно вищий рівень наочності, значно розширити можливості включення різноманітних вправ у навчальний процес, а також, підкріплені продуманими стимулами навчання, безперервним зворотним зв'язком, активізацією процесу навчання, що в кінцевому підсумку підсилює його динамізм. Це веде до реалізації майже основних завдань практично-процесуальних аспектів навчання - формування позитивного ставлення учня до матеріалу, що вивчається, інтересу до нього, задоволення результатами кожного часткового етапу навчання. . Комп'ютерне навчання сьогодні наповнює свій зміст новими елементами, не змінюючи традиційної структури навчально-пізнавальної діяльності.

Викладачі побачили у комп'ютері потужний засіб навчання та намагаються його застосувати у навчальному процесі. Але в розробників КНС виникає безліч проблем: від неможливості використання певних методів подання інформації до складності програмування педагогічних задач.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		7

Так що мріючи створити універсальний засіб, часто обмежуються фрагментарними вставками програмних продуктів у навчальний процес. Найчастіше це виникає через відсутність необхідної інформації про можливості КНС. Тому необхідно визначити структуру та найважливіші суттєві вимоги, що пред'являються до КНС.

Більшість КНС побудовані за схемою: виклад матеріалу, тренування, контроль.

Програма може і не містити всі перелічені пункти, а зводиться тільки до одного з них, і тоді вона стає вузько направленою: демонстраційною, контролюючою або тренажером. Та й важко придумати щось інше, оскільки загальна структура визначається зрештою поставленими навчальними цілями [1].

У зв'язку зі сказаним у структурі більшості КНС має містити такі блоки:

- інформаційно-теоретичний блок, що охоплює основні відомості дисципліни або розділу.
- інформаційно-довідковий блок або інформаційно-довідкову систему, що дозволяє організувати швидкий доступ до основних ухвал.
- блок контролю знань та обробки його результатів.

Інформаційно-теоретичний блок – блок пред'явлення навчальної інформації – містить набір екранів, де відображається весь обсяг навчального матеріалу КНС. За простотою підготовки матеріалу цього блоку стоїть надзвичайно складна проблема отримання виграшу від електронного помічника. Враховуючи, що читання книги або іншого друкованого тексту більш звичне та зручне, ніж робота за монітором, рекомендується уникати примітивної заміни паперової літератури на електронну.

Позбутися книжкового методу пред'явлення навчальної інформації можна шляхом використання наступних прийомів [2]:

- інформацію видавати в короткій формі, а повний текст пред'являти лише за запитом, при цьому студенти швидко вивчать коротку форму, а повна версія зацікавить просунутих студентів.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		8

– ширше використовувати структурування навчального матеріалу та представляти його на екрані не текстом, а у вигляді схем, графіків, мнемонічних процедур.

– за необхідності використовувати посилання підручники, навчальні посібники, довідники.

– урізноманітнити форму та порядок пред'явлення інформації: наприклад, подвійне підсвічування виділяти ключові слова, використовувати можливості фігурного заповнення екрана текстовим матеріалом, висвічувати не відразу весь текст, а із затримкою в часі, що відповідає раціональному темпу читання.

Перерахований набір не претендує на повноту, але психологічно дуже важливо здійснити перехід від цікавості формою до глибшого інтересу за змістом.

У сучасній освітній практиці тестування переважно використовується для вимірювання навчальних досягнень учнів, проведення якісного та кількісного аналізу результатів їхньої навчально-пізнавальної діяльності.

Комп'ютерні навчальні тести повинні відрізнятися за своєю структурою від контролюючих тестових програм, відповідно до своїх цілей та завдань у навчальному процесі. На відміну від контролюючих тестів, де незмінно присутня оцінка знань, умінь і навичок, яка нерідко призводить до стресу, страху втрати рейтингу перед товаришами по навчанню і перед собою, навчальні тести повинні спиратися принцип накопичення успіху при освоєнні знань. Інакше висловлюючись, в навчальному тестуванні має бути присутнім не «віднімальна», а накопичувальна система балів, що підвищує мотивацію до навчання з допомогою посилення емоційної компоненти. Крім того, уникнення звичайної оцінки знань знижує стрес при постійному застосуванні навчальних та розвиваючих тестів у навчальному процесі.

1.2 Приклади аналогічних програмних розробок

Перед виробленням вимог до проектного додатку слід вивчити існуючі аналоги. На ринку їх є достатньо багато, тому візьмемо найпопулярніші із них.

1.2.1 Пакет програм SunRav

SunRav — програма для створення тестів, проведення тестування та обробки його результатів [3].

Програмний пакет SunRav TestOfficePro включає програми для створення тестів, тестування та обробки результатів тестів. Може використовуватися для організації та проведення іспитів та іспитів у навчальних закладах (університетах, коледжах, школах), підприємства та організації можуть атестувати та атестувати своїх працівників.

Інтерфейс програми «SunRav» показано на рис. 1.1.

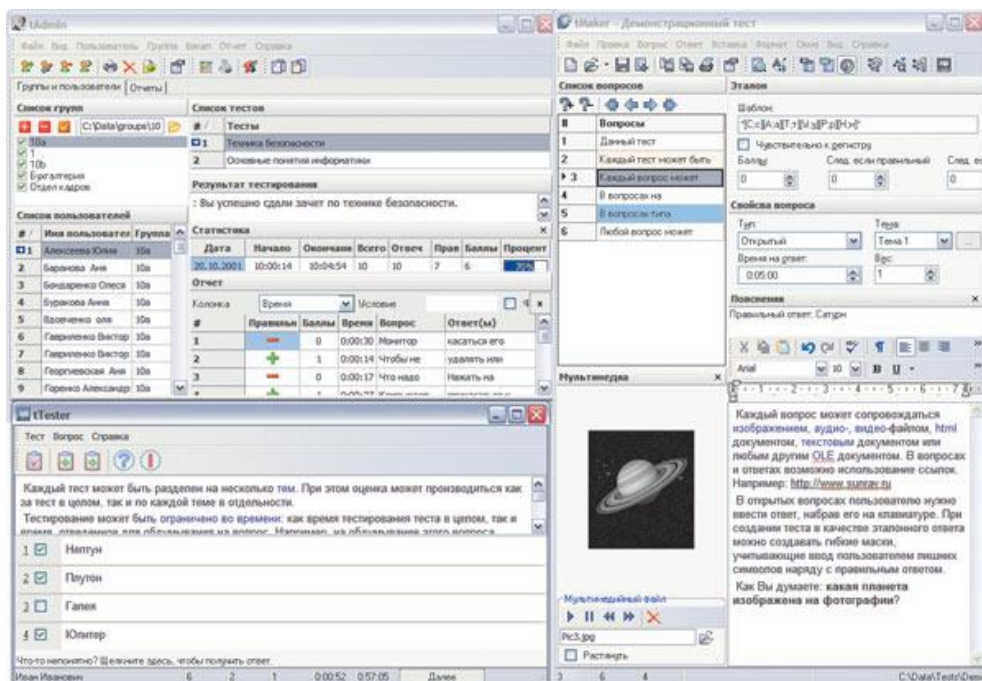


Рисунок 1.1 – Інтерфейс програми «SunRav»

Слід відмітити, що інтерфейс програми є перевантаженим, що може створити труднощі для деяких користувачів.

Також результати тестування шифруються методами криптографії, що робить роботу складною підробку результатів.

На сьогоднішній день пакет програм SunRav Software є потужним інструментом для проведення комп'ютерного тестування та створення електронних книг і підручників.

В таблиці 1.1 представлені недоліки та переваги даного пакету.

Таблиця 1.1 – Переваги та недоліки системи тестування SunRav

Переваги	Недоліки
Тести та результати тестування шифруються методами стійкої криптографії, що унеможлиблює підробку результатів	Недостатня тех. підтримка
На тест можна встановити паролі	Перевантажений функціонал
Запитання та варіанти відповідей можна форматувати, використовуючи для цього вбудований редактор	Висока ціна (корпоративна ліцензія на ПЗ SunRav TestOfficePro / XT - 30000 грн.
Тестування можна обмежити за часом як для тесту, так і для кожного питання	Немає українського інтерфейсу

1.2.2 Система тестування INDIGO

Система тестування INDIGO – це професійний інструмент автоматизації процесу тестування та обробки результатів, який призначений на вирішення широкого спектра задач [4]:

- тестування та контроль знань учнів;
- визначення професійного рівня співробітників;
- проведення психологічного тестування;
- проведення опитувань;
- організація олімпіад та конкурсів.

Система INDIGO є потужним та гнучким засобом для проведення випробувань. Можлива робота із системою як через локальну мережу, так і через Інтернет.

Є вбудований редактор тестів, система правил для проходження тестів, розроблені засоби для підвищення безпеки даних.

На рисунку 1.2 представлена форма вибору відповіді на запитання проходженні тесту на платформі INDIGO. З малюнка видно, що елементи управління оформлені класичним стилем, внаслідок чого інтерфейс відповіді на запитання у INDIGO менш зручний.

Рисунок 1.2 – Форма вибору відповіді питання при проходженні тесту на платформі INDIGO

Функціональні можливості:

- система тестування встановлюється на головний комп'ютер (сервер тестування) за допомогою інсталяційного пакета;
- система може працювати як ізольованому комп'ютері, і у локальній мережі чи через Інтернет;
- центр тестування можна розгорнути на Вашому комп'ютері або у хмарі наших Інтернет-серверах;
- усі дані зберігаються централізовано у базі даних системи;
- адміністратори працюють через програму клієнта;
- одночасно можуть працювати скільки завгодно адміністраторів із різних комп'ютерів;
- користувачі працюють через веб-браузери (Google Chrome, Яндекс.Браузер, Mozilla Firefox, Opera, Safari, Internet Explorer, Microsoft Edge та інші). Існує підтримка web-браузерів на мобільних пристроях (адаптивний інтерфейс);
- є підтримка інтеграції з Active Directory, 1С:Підприємством або з довільною інформаційною системою організації.

Таблиця 1.2 – Переваги та недоліки системи тестування INDIGO

Переваги	Недоліки
Можливість гнучкого налаштування підрахунку результатів	Висока ціна (стандартна ліцензія 12000 грн.)
Існує можливість проведення аналізу виконаних робіт, перегляд помилок	Перевантажений інтерфейс
Простота інтерфейсу, все просто і зрозуміло	Немає української мови
Можна швидко виводити результати як по одному студенту, так і по всій групі	Для користування програмою необхідна установка певного програмного забезпечення
Система гнучка в налаштуванні	

Зручність в адмініструванні та
веденні статистики

Отже, система тестування INDIGO призначена для закладів, які мають високі бюджети можуть та можуть дозволити собі цей продукт.

1.2.3 Конструктор тестів Keepsoft

Загальна процедура перевірки знань. Додаток можна використовувати для тестування вдома та в школі. Програма дозволяє використовувати необмежену кількість тем, запитань і відповідей [5].

Інтерфейс конструктору тестів «Keepsoft» показано на рис. 1.3.

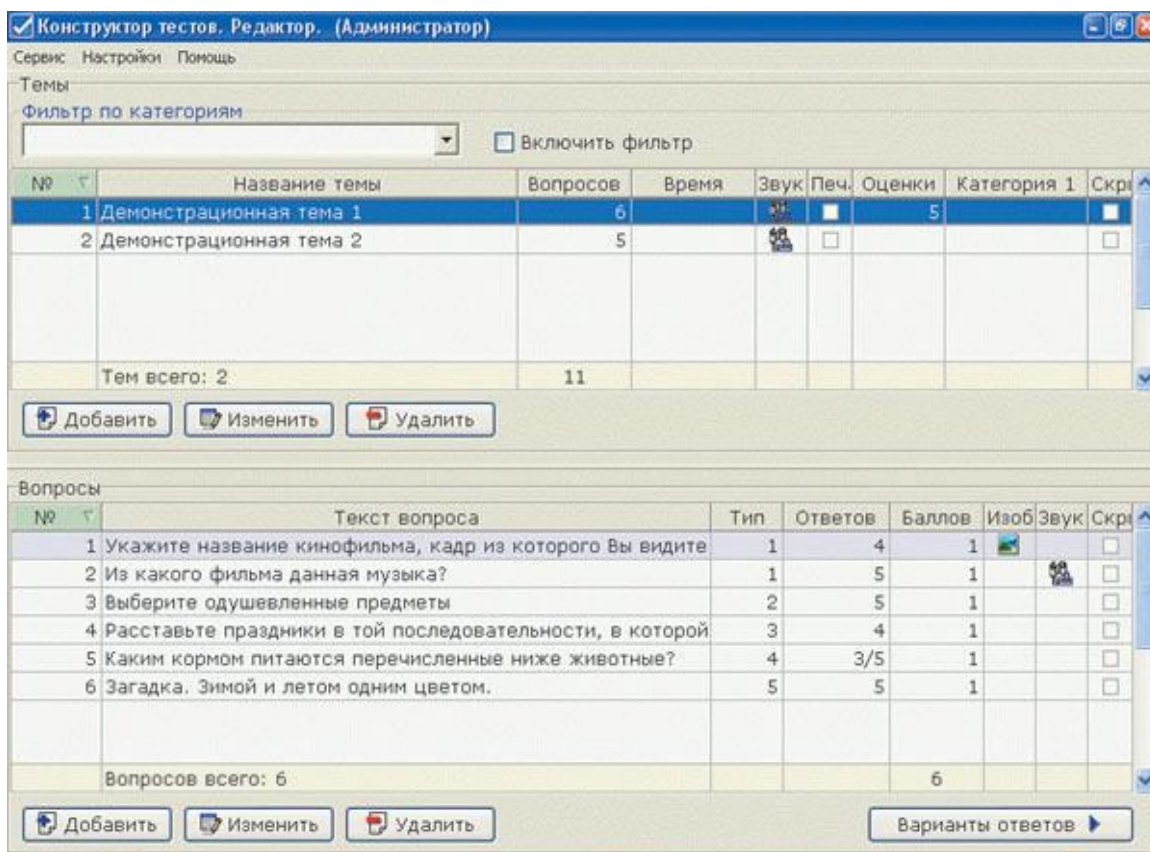


Рисунок 1.3 – Конструктору тестів «Keepsoft»

Програму можна завантажити з сайту розробника і читати безкоштовно протягом 30 днів. Подальше використання програми вимагає платної реєстрації. Особисті ключі реєстрації та безкоштовна технічна підтримка надаються зареєстрованим користувачам електронною поштою.

Нижче в таблиці 1.3 представлені недоліки та переваги конструктору тестів «Keepsoft».

Таблиця 1.3 – Переваги та недоліки конструктору тестів «Keepsoft»

Переваги	Недоліки
Питання можуть містити музику (файли WAV, MID, RMI), зображення (файли JPG, BMP, ICO, EMF, WMF), відеоролики	Застарілий дизайн тестів
Підтримуються питання всіх п'яти перерахованих вище типів	Немає повного перегляду тесту
Призначення ціни по кожному у балах	Немає української мови
Обмеження часу на відповідь	
Синхронізація бази даних; за допомогою цієї функції ви можете легко обмінюватися даними з іншими користувачами та передавати дані з комп'ютера на комп'ютер	

1.3 Постановка задачі

Дипломна робота припускає:

- розробку схеми БД;
- розробку та реалізацію додатка, що включає три основні модулі:

- 1) модуль вводу/редагування інформації в БД;
- 2) модуль тестування знань;
- 3) модуль формування звітності.

Реалізація додатка виконується з використанням вивчених технологій.

Результат – Windows додаток із використанням технології .NET.

Тема дипломної роботи – програмий застосунок тестування знань студентів

Вона повинна містити в собі:

Сутності:

- користувачі системи (інформація про користувачів та їхні облікові записи): прізвище, ім'я, нікнейм облікового запису, пароль, загальна інформація;
- дисципліни (інформація про дисципліни): назва дисципліни, опис та час тестування знань по цій дисципліні;
- програми: назва навчальної програми, надпис програми у вікні, час запуску програми, проміжок часу запущеної програми, ідентифікатор тесту студента, статус програми запуску тестування (до запуску тестування, під час запуску тестування);
- тести (містить тести для тестування студентів): запитання, текстова відповідь А, логічна відповідь А, текстова відповідь В, логічна відповідь В, текстова відповідь С, логічна відповідь С, текстова відповідь D, логічна відповідь D, , ідентифікатор компанії, ідентифікатор дисципліни;
- статистика (інформація про проходженню тестів): ідентифікатор дисципліни, тривалість тестування, ідентифікатор студента, кількість запитань в тесті, кількість правильних відповідей, кількість неправильних відповідей ;
- логи (інформація про активність користувачів системи) ідентифікатор користувача, подія в системі, дата/час події.

Введення:

					IA82.010BAK.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		16

Вводиться і редагується: користувачі системи, дисципліни, тести для тестування студентів.

Проводяться формування запитань для студентів.

Проводиться тестування студентів.

Фіксується проходження тестування студентами.

Фіксується запис подій в системі.

Звітність: по пройденим тестам за весь період часу (коли, хто, по якому предмету, яка програма була запущена).

1.4 Висновок до розділу

Підводячи підсумки про вище перелічені системи проходження тестування, треба сказати, що всі платформи є досить хорошими для тестування знань, але нажаль вони спрямовані на отримання прибутку від користувачів.

Також слід відмітити наявність ряд мінусів у певних систем, таких як відсутність української мови, неможливість користуватися деякими програмами безкоштовно постійно.

Розроблену в майбутньому систему без проблем можна копіювати та переносити на інші комп'ютери на яких встановлена операційна система Windows та встановлена бібліотека класів .NET framework 4.7 або вище. Буде розроблений зручний та інтуїтивно зрозумілий інтерфейс.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		17

2 РОЗРОБКА

2.1 Аналіз вимог. Use-case діаграми. Основні прецеденти

Згідно з поставленою задачею можна висунути такий список вимог до проекту:

Таблиця 2.1 – Функціональні вимоги до додатку «Тестування знань студентів»

Вимоги	Опис
REQ-1	Система повинна дозволяти здійснювати реєстрацію користувача в програмі
REQ-2	Система повинна вести журнал подій, що відбулися в програмі
REQ-3	Система повинна дозволяти користувачеві додати та редагувати інформацію про облікові записи користувачів
REQ-4	Система повинна дозволяти користувачеві додати та редагувати інформацію про дисципліни
REQ-5	Система повинна дозволяти користувачеві додати та редагувати інформацію тестування студентів
REQ-6	Система повинна дозволяти користувачеві проходити тестування студентами
REQ-7	Система повинна дозволяти проводити запис запускених програм під час проходження тестування
REQ-8	Система повинна дозволяти формувати звітність про успішність проходження тестування

Таблиця 2.2 – Нефункціональні вимоги до додатку «Тестування знань студентів»

Вимоги	Опис
REQ-9	Додаток повинен мати простий дизайн та зручну навігації
REQ-10	Поля повинні бути не порожніми, унікальними відносно вже існуючих записів

Таблиця 2.3 – Актори та цілі додатку «Тестування знань студентів»

Актори	Цілі
Користувач	Мета користувача полягає в роботі з системою.
База даних	Мета бази даних полягає у зберіганні інформації

Таблиця 2.4 – Опис варіантів використання додатка «Тестування знань студентів»

Варіант використання	Ім'я	Опис
UC1	Ідентифікація в системі	Дозволяє користувачу пройти ідентифікацію в системі
UC2	Вивід каталогу користувачів	Дозволяє користувачеві з правами системного адміністратора вивести каталог всіх користувачів системи
UC3	Додати користувача	Дозволяє користувачеві додати нового користувача системи

UC4	Редагувати користувача	Дозволяє користувачеві редагувати інформацію вибраного із списку користувача
UC5	Видалити користувача	Дозволяє користувачеві видалити вибраного із списку користувача
UC6	Вивід каталогу дисциплін	Дозволяє користувачеві вивести каталог всіх дисциплін, по яких система буде проводити тестування
UC7	Додати дисципліну	Дозволяє користувачеві додати нову дисципліну
UC8	Редагувати дисципліну	Дозволяє користувачеві редагувати інформацію вибраної із списку дисципліни
UC9	Видалити дисципліну	Дозволяє користувачеві видалити вибрану із списку дисципліну
UC10	Вивід каталогу запитань	Дозволяє користувачу вивести список всіх запитань для різних спеціальностей
UC11	Додати запитання	Дозволяє користувачу додати нове запитання в систему для вибраної дисципліни
UC12	Редагувати запитання	Дозволяє користувачеві редагувати інформацію вибраного із списку запитання
UC13	Видалити запитання	Дозволяє користувачеві видалити вибране із списку запитання
UC14	Проведення тестування	Дозволяє студенту пройти тестування
UC15	Звітність по тестуванню	Дозволяє користувачеві програми формувати звітність по проходженню тестування студентами
UC16	Вивід системних подій	Дозволяє вивести всі події, які відбулися в системі

З поставлених вимог (див. розділ 1) тепер можна виставити повний опис вимог із сценаріями, що будуть вхідними даними для візуального моделювання мовою UML.

UC1 Ідентифікація в системі

Актор: користувач.

Ціль актора: пройти ідентифікацію в системі.

Задіяний актор: база даних.

Передумова: користувач, вибравши ім'я та ввівши пароль, натискає кнопку «Підтвердити».

Післяумова: система відображає головне вікно програми з меню.

UC2 Вивід каталогу користувачів

Актор: користувач.

Ціль актора: вивести інформацію про всіх користувачів системи.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Користувачі».

Післяумова: система відображає екран для виведення списку всіх зареєстрованих користувачів системи.

UC3 Додати користувача

Актор: користувач.

Ціль актора: додати нового клієнта.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про користувача натискає на кнопку «Додати».

Післяумова: система додає нового користувача та відображає екран із списком всіх користувачів.

UC4 Редагувати користувача

Актор: користувач.

Ціль актора: редагувати інформацію про вибраного користувача.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку користувача.

Післяумова: система відображає екран для редагування інформації про вибраного користувача.

UC5 Видалити користувача

Актор: користувач.

Ціль актора: видалити користувача із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідного із списку користувача.

Післяумова: система відображає екран з можливістю видалення даних про користувача.

UC6 Вивід каталогу дисциплін

Актор: користувач.

Ціль актора: вивести інформацію про дисципліни.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Дисципліни».

Післяумова: система відображає екран для виведення списку всіх дисциплін.

UC7 Додати дисципліну

Актор: користувач.

Ціль актора: додати нову дисципліну.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про дисципліну натискає на кнопку «Додати».

Післяумова: система додає нову дисципліну і відображає екран із списком всіх дисциплін.

UC8 Редагувати дисципліну

Актор: користувач.

Ціль актора: редагувати вибрану із списку дисципліну.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідну із списку дисципліну.

Післяумова: система відображає екран для редагування інформації про вибрану дисципліну.

UC9 Видалити дисципліну

Актор: користувач.

Ціль актора: видалити вибрану дисципліну із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідну із списку дисципліну.

Післяумова: система відображає екран з можливістю видалення даних цієї дисципліни.

UC10 Вивід каталогу запитань

Актор: користувач.

Ціль актора: вивести інформацію про всі запитання.

Задіяний актор: база даних.

Передумова: користувач натиснув на пункт меню «Тести».

Післяумова: система відображає екран для виведення списку всіх запитань.

UC11 Додати запитання

Актор: користувач.

Ціль актора: додати нове запитання.

Задіяний актор: база даних.

Передумова: користувач ввівши всю необхідну інформацію про запитання, що буде проводитись в підготовці до співбесіди та натискає на кнопку «Додати».

Післяумова: система додає нове запитання та відображає екран із списком всіх запитань.

UC12 Редагувати запитання

Актор: користувач.

Ціль актора: редагувати вибране із списку запитання.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідне із списку запитання.

Післяумова: система відображає екран для редагування інформації про вибране запитання.

UC13 Видалити запитання

Актор: користувач.

Ціль актора: видалити запитання із бази даних.

Задіяний актор: база даних.

Передумова: користувач вибирає необхідне із списку запитання.

Післяумова: система відображає екран з можливістю видалення даних про запитання.

UC14 Проведення тестування

Актор: користувач.

Ціль актора: пройти тестування із вибраної дисципліни

Задіяний актор: база даних.

Передумова: користувач вибравши свій обліковий запис та ввівши пароль натискає на реєстраційній формі кнопку «Підтвердити» .

Післяумова: система відображає екран з можливістю проведення навчальної тестування знань по вибраній дисципліні.

UC15 Звітність по проходженню тестування

Актор: користувач.

Ціль актора: вивести детальну інформацію по проходженню тестування.

Задіяний актор: база даних.

Передумова: користувач натискає на пункт меню «Рейтинги».

Післяумова: система відображає екран з можливістю формування звітності по проходженню тестування.

UC16 Вивід системних подій

					IA82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		24

Актор: користувач.

Ціль актора: вивести список всіх подій в системі.

Задіяний актор: база даних.

Передумова: користувач вибирає пункт меню «Системний журнал».

Післяумова: система відображає екран із списком всіх системних подій.

В рамках дослідницького проекту була розроблена діаграма варіантів використання, що наведена в документі ІА82.010БАК.003 Д1

2.2 Архітектура проекту

Розроблений продукт повинен відповідати характеристикам якості, таким як: стійкість, корисність, доступність, масштабованість, відкритість, гнучкість, можливість тестування [14]. Це вимагає від процесу розробки додаткові обмеження/правила, а саме:

- дотримання шаблонів і стилів;
- документування розробки на різних рівнях;
- тестування компонентів, окремих модулів, підсистем;
- управління проектами, процесами.

З урахуванням вимог до забезпечення стійкості та гнучкості системи при її розробці було обрано шаблон Layers, який розбиває систему на дві частини: клієнт та сервер.

Проектування системи буде покладатись на предметну область (DDD підхід) та принципи SOLID [5].

При розробці клієнта був обраний користувацький інтерфейс Windows Forms.

Сервер, в свою чергу, буде складатись з таких модулів:

- 1) BLL (англ. Business Logic Layer) – логіка та всі необхідні обчислення додатку на мові бізнесу;

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		25

- 2) DAL (англ. Data Acces Layer) – рівень доступу до даних;
- 3) DB (англ. Data Base) – база даних для зберігання даних.

З урахування всіх вище перерахованих шаблонів структура проекту буде виглядати наступним чином:

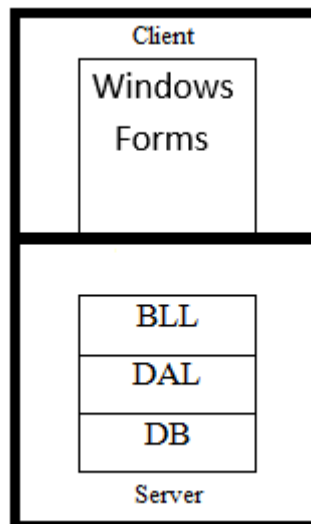


Рисунок 2.1 – Структура проекту

Кожен компонент буде реалізувати контракт (інтерфейс), який надає на гнучкість компоненту.

Оскільки бізнес постійно і відносно швидко змінюється, то кожен із компонентів повинен швидко адаптуватись під ці зміни. Для цього використаємо шаблон Dependency Injection (DI).

2.3 Особливості розробки бази даних. ERD діаграма з описанням сутностей

Шаблон взаємовідносин сутностей (також відомий як діаграма ERD або ER) — це блок-схема, яка показує, як різні «сутності» (люди, об'єкти, поняття тощо) пов'язані один з одним у системі. ER-діаграми найчастіше застосовуються для проектування та налагодження реляційних баз даних у

сфері освіти, дослідження та розробки програмного забезпечення та інформаційних систем для бізнесу [15].

Діаграми ER (або моделі ER) спираються на стандартний набір символів, включаючи прямокутники, ромби, еліпси та сполучні лінії, для представлення сутностей, їхніх властивостей та зв'язків. Ці діаграми влаштовані за тими ж принципами, що й граматичні структури: сутності діють як іменники, а зв'язки діють як дієслова.

ER-діаграми - "родичі" схем структури даних (DSD), де замість зв'язків між самими сутностями відображається відношення між елементами всередині них. ER-діаграми часто використовуються у поєднанні з діаграмами DFD, що схематично показують рух потоків інформації в рамках процесу або системи.

У ER-моделях та моделях даних зазвичай виділяють до трьох рівнів деталізації:

Концептуальна модель даних – схема найвищого рівня з мінімальною кількістю подробиць. Перевага цього підходу полягає у можливості відобразити загальну структуру моделі та всю архітектуру системи. Менш масштабні системи можуть обійтися без цієї моделі. І тут можна відразу переходити до логічної моделі.

Логічна модель даних: містить більш детальну інформацію, ніж концептуальна модель. На цьому рівні визначаються докладніші операційні та транзакційні сутності. Логічна модель залежить від технології, у якій застосовуватиметься.

Фізична модель даних: на основі кожної логічної моделі даних можна становити одну або дві фізичні моделі. В останніх має бути достатньо технічних подробиць для складання та впровадження самої бази даних.

Можна звернути увагу на той факт, що схожі рівні масштабу та деталізації зустрічаються і в інших видах схем (наприклад, у діаграмах DFD).

Проте дана класифікація відрізняється від трьохсхемного підходу у розробці ПЗ, де розподіл інформації здійснюється за дещо іншим принципом. Щоправда, іноді розробники застосовують ER-діаграми з додатковими ієрархіями, якщо дизайн бази даних потребує більше інформаційних рівнів. Наприклад, розробник може додати нові групи за принципом розширення вгору (суперкласи) та вниз (підкласи). А саме:

- лише реляційні дані. Слід чітко розуміти, що мета ER-діаграм – показати зв'язки та відносини між елементами, тому вони відображають лише реляційну структуру;

- лише для структурованих даних. Дані мають бути чітко розбиті на поля, стовпці та рядки, інакше користі від ER-діаграми буде мало. Це стосується і частково структурованих даних, оскільки лише деякі з них будуть придатними для роботи;

- складність інтеграції з базою даних. Застосування ER-моделей для інтеграції з існуючою базою даних — непросте завдання через різницю в архітектурах.

Згідно завдання будемо ERD діаграму, яка приведена в документі ІА82.010БАК.003 Д2.

Як ми бачимо наша база даних складається із 7 сутностей. Кожна сутність має свою таблицю, а саме:

1. Таблиця «TestType» – зберігає інформацію про типи тесту, а саме: картинка або текст.

2. Таблиця «Disciplines» - зберігає інформацію про всі дисципліни, по яких будуть проводитись тести.

3. Таблиця «Logs» - зберігає інформацію про активність користувачів системи і їхні дії.

4. Таблиця «Programs» - зберігає інформацію про всі програми, що були запущені до і під час тестування знань.

5. Таблиця «Statistic» - зберігає інформацію про результати проходження тестування.

6. Таблиця «Tests» - містить інформацію про всі тести (запитання відповіді) для всіх дисциплін.

7. Таблиця «Users» - містить інформацію про всіх користувачів системи та їхні облікові дані.

2.4 Особливості розробки рівня BLL

Бізнес-логіка - при розробці інформаційних систем - набір правил, принципів, залежностей від поведінки об'єктів у предметній області (сфері діяльності людини, що підтримується системою). Іншими словами, бізнес-логіка — це реалізація правил і обмежень для автоматизованих операцій. Синонім терміну «логіка домену». Бізнес-логіка визначає правила управління даними предметної області.

Простіше кажучи, бізнес-логіка – це реалізація предметної галузі в інформаційній системі. До неї відносяться: формули розрахунку щомісячних виплат з позик (у фінансовій індустрії), автоматизована відправка повідомлень електронної пошти керівнику проекту після закінчення частин завдання всіма підлеглими (у системах управління проектами), відмова від готелю при скасуванні рейсу авіакомпанією (у туристичному бізнесі)) і т.д.

У фазі бізнес-моделювання та розробки вимог бізнес-логіка описується у вигляді:

- тексту;
- концептуальних аналітичних моделей предметної галузі (онтології);
- бізнес-правил;
- рішних алгоритмів;
- діаграм діяльності;
- графів та діаграм переходу станів;
- моделей бізнес-процесів.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		29

У фазі аналізу та проектування системи бізнес-логіка втілюється в різних діаграмах мови UML або подібних до неї. У фазі програмування бізнес-логіка втілюється в коді класів та їх методів, у разі використання об'єктно-орієнтованих мов програмування, або процедур та функцій у разі застосування процедурних мов.

На жаргоні розробників програмного забезпечення «бізнес-логікою» також називаються програмні модулі, що її реалізують, та рівень системи, на якому ці модулі знаходяться (англ. business logic layer).

У багаторівневих (багатошарових) інформаційних системах цей рівень взаємодіє з нижчим рівнем інфраструктурних сервісів (англ. infrastructure layer), наприклад, інтерфейсом доступу до бази даних або файлової системи (англ. data-access layer, DAL) і рівнем сервісів додатка (англ. application services layer), який, своєю чергою, взаємодіє з рівнем користувальницького інтерфейсу (англ. user interface layer) чи зовнішніми системами.

BLL буде обробляти речі, які є частиною бізнес-сфери, а не частиною бази даних і не частиною користувальницького інтерфейсу.

В даному проекті реалізовано клас для формування звітів під назвою: «StatisticBLL».

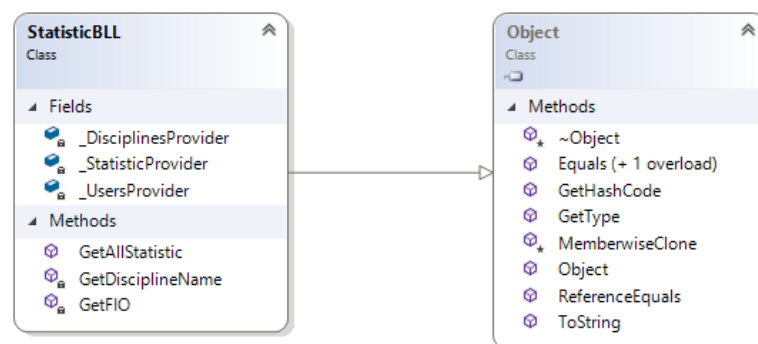


Рисунок 2.2 – Діаграма класу BLL

2.5 Особливість реалізації бізнес логіки – діаграма домена

Будуємо діаграму класів домена. Кожен клас описує конкретну таблицю бази даних для зручного опрацювання даних.

Була побудована діаграма класів домена, що наведена в документі ІА82.010БАК.003 Д4.

2.6 Особливості розробки рівня UI

User interface (UI) елементи – це частини, які дизайнери використовують для створення програм або веб-сайтів. Вони додають інтерактивність в інтерфейс користувача, надаючи користувачеві точки зіткнення при навігації по них. Кнопки, смуги прокручування, пункти меню та чекбокси.

Інтерфейсу користувача (UI) використовує UI елементи для створення візуальної мови і забезпечення узгодженості продукту, що робить його зручним для користувача і простим у навігації без особливих зусиль з боку користувача.

UI елементи зазвичай поділяються на одну з наступних чотирьох категорій:

- елементи керування введенням (Input Controls) – дозволяють користувачам вводити інформацію до системи;
- компоненти навігації (Navigation Components) – допомагають користувачам переміщатися продуктом або веб-сайтом. Загальні навігаційні компоненти включають панелі вкладок та головне меню програми;
- інформаційні компоненти (Informational Components) – діляться інформацією з користувачем;
- контейнери (Containers) – містять зв'язаний контент разом.

В даному проекті для розробки користувацького інтерфейсу я використав Windows Forms.

Windows Forms - це платформа користувача інтерфейсу для створення класичних додатків Windows. Вона забезпечує один з найефективніших способів створення класичних додатків за допомогою візуального конструктора в Visual Studio.

В рамках дослідницького проєкту була розроблена діаграма варіантів використання, що наведена в документі ІА82.010БАК.003 ДЗ.

2.7 Особливості розробки класів для доступу до даних бази даних (DAL)

Рівень рівня доступу до даних (DAL) у програмному забезпеченні — це рівень комп'ютерної програми, який забезпечує спрощений доступ до даних, що зберігаються в деякому типі постійного сховища, наприклад у реляційній базі даних. Ця аббревіатура в основному використовується в середовищі Microsoft .NET.

DAL може повертати посилання на об'єкт (з точки зору об'єктно-орієнтованого програмування) та його властивості, а не на рядок поля в таблиці бази даних. Завдяки цьому створюються клієнтські (або користувацькі модулі) модулі з більшим рівнем абстракції. Така модель може бути реалізована шляхом створення класу з методами доступу до даних, які безпосередньо посилаються на відповідний набір процедур бази даних. Інша реалізація потенційно може отримувати записи або записувати записи до файлової системи. DAL приховує складність базового сховища.

Замість використання таких команд як «створити», «видалити» або «оновити» в певній таблиці в базі, клас і кілька процедур, що зберігаються, можуть бути створені в базі. Ці процедури можуть викликатися методом усередині класу, який поверне об'єкт, що містить запитані значення. Або команди створення, видалення та оновлення можуть бути виконані всередині простих функцій, що зберігаються у шарі доступу до даних.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		32

Також методи бізнес-логіки із програми можуть бути співвіднесені до шару доступу до даних.

Для роботи з базою даних було реалізовано 7 класів. Назва кожного класу починається з відповідній їй назві таблиці в базі даних та закінчується приставкою «Provider».

На рис. 2.4 приведено діаграму класів з методами для роботи з базою даних.

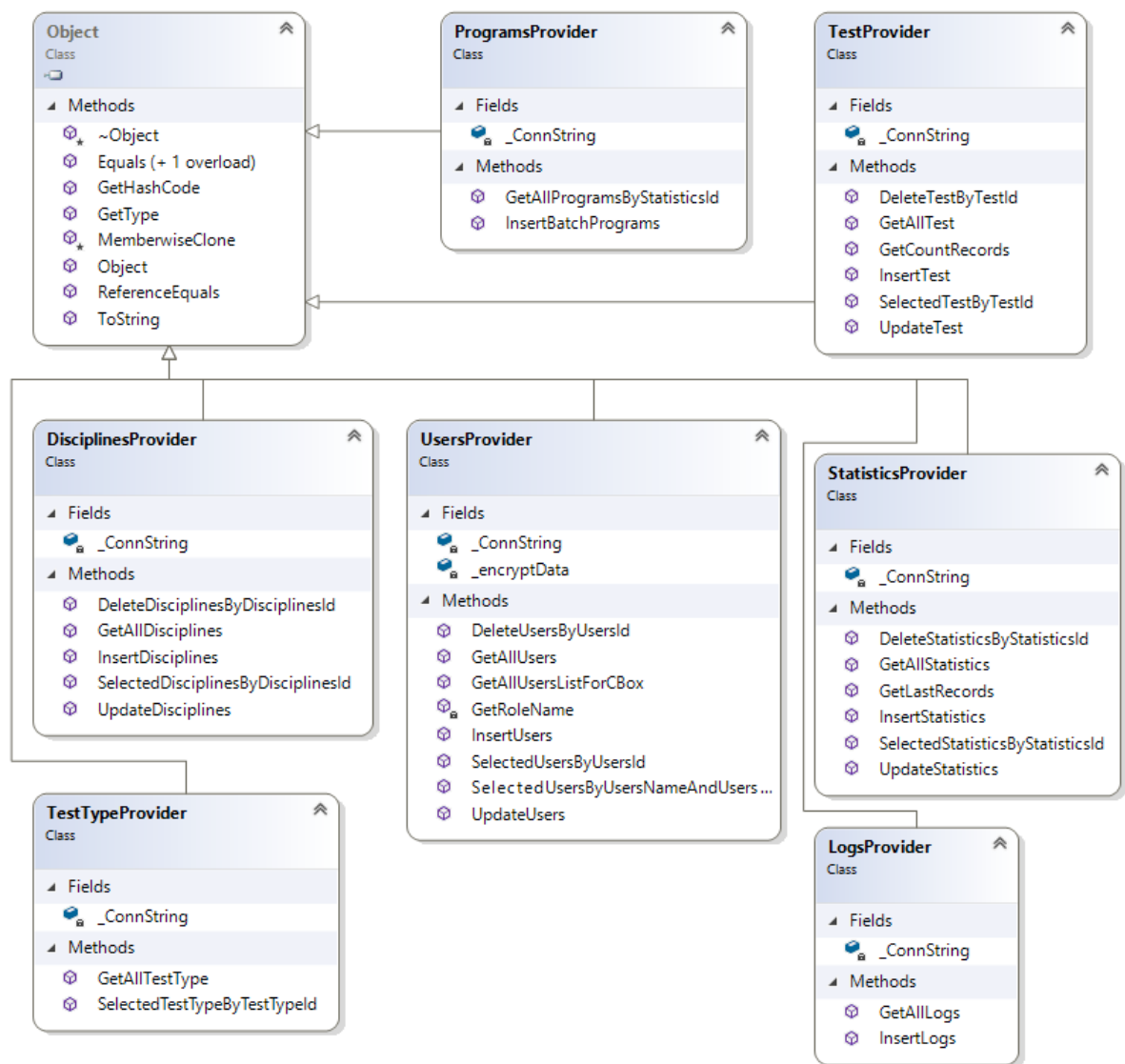


Рисунок 2.4 – Діаграма класів з методами для роботи з базою даних

2.8 Висновок до розділу

Літ	Арку	№ докум.	Підп	Дата
-----	------	----------	------	------

Побудова сучасних інформаційних система займає дуже багато часу. Вона починається з аналізу предметної області, детального планування системи, описання вимог, моделювання її поведінки за допомогою uml діаграм. Визначається шаблони, які будуть використовуватись при розробці. Після планування починається стадія розробки. Розробка починається зі створення користувацького інферфейсу і закінчується базою даних.

При розробці слід враховувати, що вимоги змінюються швидко і потрібно будувати систему так, щоб вона була гнучкою.

Розробка системи виконується по окремим компонентам. Кожний створений компонент потрібно детально тестувати, щоб мінімізувати помилки на наступних етапах розробки.

Якщо дотримуватися всіх вищеперерахованих вимог, то можна побудувати гнучку і стійку інформаційну систему.

					IA82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		34

3 РЕАЛІЗАЦІЯ

3.1 Вибір технологій

Для побудови користувацького інтерфейсу будемо використовувати середовище розробки Visual Studio 2019 та мову програмування C#.

C# широко використовується професіоналами для розробки великих програмних продуктів завдяки наступним аспектам мови [14]:

- Сучасна мова програмування загального призначення.
- Підтримка об'єктно-орієнтованої парадигми.
- Підтримка компонентно-орієнтованої парадигми.
- Мова легка для вивчення навчитися.
- Добре структурована.
- Дозволяє розробляти ефективні програми.
- Мова має підтримку різних комп'ютерних платформах.
- Це частина .Net Framework.

C# був розроблений для відповідності традиційним мовам високого рівня C і C++ і є об'єктно-орієнтованою мовою програмування. Мова дуже схожа на Java і має багато потужних функцій програмування, які роблять її привабливою для багатьох програмістів у всьому світі [5].

.NET Framework - це платформа для розробників із відкритим кодом, яку можна використовувати для створення широкого кола програм. Цей безкоштовний крос-платформний фреймворк підтримує декілька мов і має великі бібліотеки коду, які спрощують створення додатків для мобільних пристроїв, робочих столів та Інтернету [15].

Платформа .NET була розроблена для досягнення наступних цілей: [16]:

- сумісність;
- підтримка різних платформам;
- мовна незалежність;

- бібліотека базових класів;
- легка розробка;
- безпека.

Платформа .Net має кілька технологій для розробки інтерфейсу користувача, однією з яких є WinForms. Хоча технологія недостатньо нова, важко сказати, що вона застаріла. Він надає широкий спектр різноманітних інструментів для створення зручних і сучасних інтерфейсів. Крім того, IDE, які підтримують C# і .Net, забезпечують зручний інтерфейс для графічної побудови користувацьких програм, які розробляються.

Тому платформи C# і .Net мають деякі характеристики, які відповідають вимогам розробки клієнтської частини системи складського обліку. Набір готових класів у стандартній бібліотеці, стислий мовний синтаксис і зручний конструктор інтерфейсу зробиють розробку легкою та досить швидкою. Об'єктно-орієнтована парадигма дозволить вам спроектувати вашу систему таким чином, щоб розширення функціональних можливостей не вимагало додаткових витрат на розробку ресурсів. Платформа .Net забезпечить безпеку, ефективність програмного забезпечення та підтримку кількох платформ.

Microsoft Access використовується для розробки сховищ інформації. Система управління базами даних MS Access, частина пакету Microsoft Office [17]. Він має широкий спектр функцій, включаючи пов'язані запити, сортування за різними полями, зв'язок із зовнішніми таблицями та базами даних.

Переваги використання [18]:

- дуже простий графічний інтерфейс, що дозволяє не тільки створювати власну базу даних, а й розробляти додатки з використанням вбудованих бібліотек;
- зберігає всі дані у одному файлі, хоч і розподіляє їх у різних таблицях, як і належить реляційної СУБД. До цих даних належить як інформація у таблицях, а й інші об'єкти бази даних;

- пропонує велику кількість «Майстрів», які виконують основну роботу за користувача при роботі з даними та розробці додатків, допомагають уникнути рутинних дій і полегшують роботу досвідченому в програмуванні користувачеві;
 - поширеність, яка зумовлена тим, що Access є продуктом компанії Microsoft;
 - постійно оновлюється виробником, підтримує безліч мов;
 - повністю сумісний з операційною системою Windows;
 - орієнтованість на користувача з різною професійною підготовкою, що виявляється у наявності великої кількості Майстрів, розвинену систему довідки та зрозумілий інтерфейс;
 - широкі можливості імпорту/експорту даних у різні формати, від таблиць Excel і текстових файлів, до практично будь-який серверної СУБД через механізм ODBC;
 - наявність розвинених інтегрованих засобів розробки додатків. Більшість програм, що розповсюджуються серед користувачів, містить той чи інший обсяг коду VBA (Visual Basic for Applications);
 - наявність вбудованої мови макрокоманд.
- Недоліки:
- обмежені можливості щодо забезпечення розрахованого на багато користувачів середовища;
 - має нескладні способи захисту з використанням пароля БД (можливе застосування додаткових заходів щодо захисту від несанкціонованого доступу з використанням процедур VBA);
 - в питаннях підтримки цілісності даних відповідає лише моделям БД невеликої та середньої складності;
 - не розповсюджується безкоштовно.

Отже СУБД Access було мною вибрано тому, що розроблена програма

					ІА82.010БАК.003 ПЗ	Арку
						37
Літ	Арку	№ докум.	Підп	Дата		

може легко переноситись на будь-які системи сімейства Windows і не потребує додаткового встановлення сервера бази даних. Реляційність бази даних дає можливість ефективно використовувати пам'ять та уникнути дублювання інформації. Завдяки відкритому коду не потрібно витрачати зайві ресурси на різні ліцензії, тобто розробка та використання системи на базі СУБД Access значно дешевша.

Підтримка проекту відбувається вже багато років розробниками з усього світу, через що надійність та безпека продукту на високому рівні. Таким чином, СУБД Access повністю покриває вимоги щодо розробленого проекту.

Для роботи з Базою даних реалізуємо класи (рис. 3.1). В кожному класі є методи, за допомогою яких можна: вибирати, вставляти, редагувати та видаляти дані. Також є методи, які працюють з вибіркою даних із декількох таблиць.

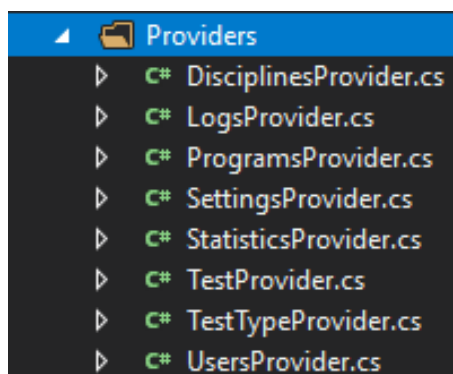


Рисунок 3.1 – Класи для роботи з БД

3.2 Середовище розробки Visual Studio

Microsoft Visual Studio - повнофункціональне інтегроване середовище розробки (IDE) за допомогою популярних мов програмування, серед яких C, C++, VB.NET, C#, F#, JavaScript, Python (рис. 3.2).

Можливості Функції Visual Studio охоплюють усі етапи розробки програмного забезпечення, надаючи найсучасніші інструменти для написання коду, проектування графічних інтерфейсів, компіляції, налагодження та тестування програм. Функціональність Visual Studio можна доповнити наступними способами підключення необхідних розширень.

Редактор коду Visual Studio підтримує підсвічування синтаксису, фрагменти коду, відображення структури та пов'язані функції. Технологія IntelliSense допомагає значно прискорити вашу роботу — код завершується автоматично, коли ви вводите символи з клавіатури.

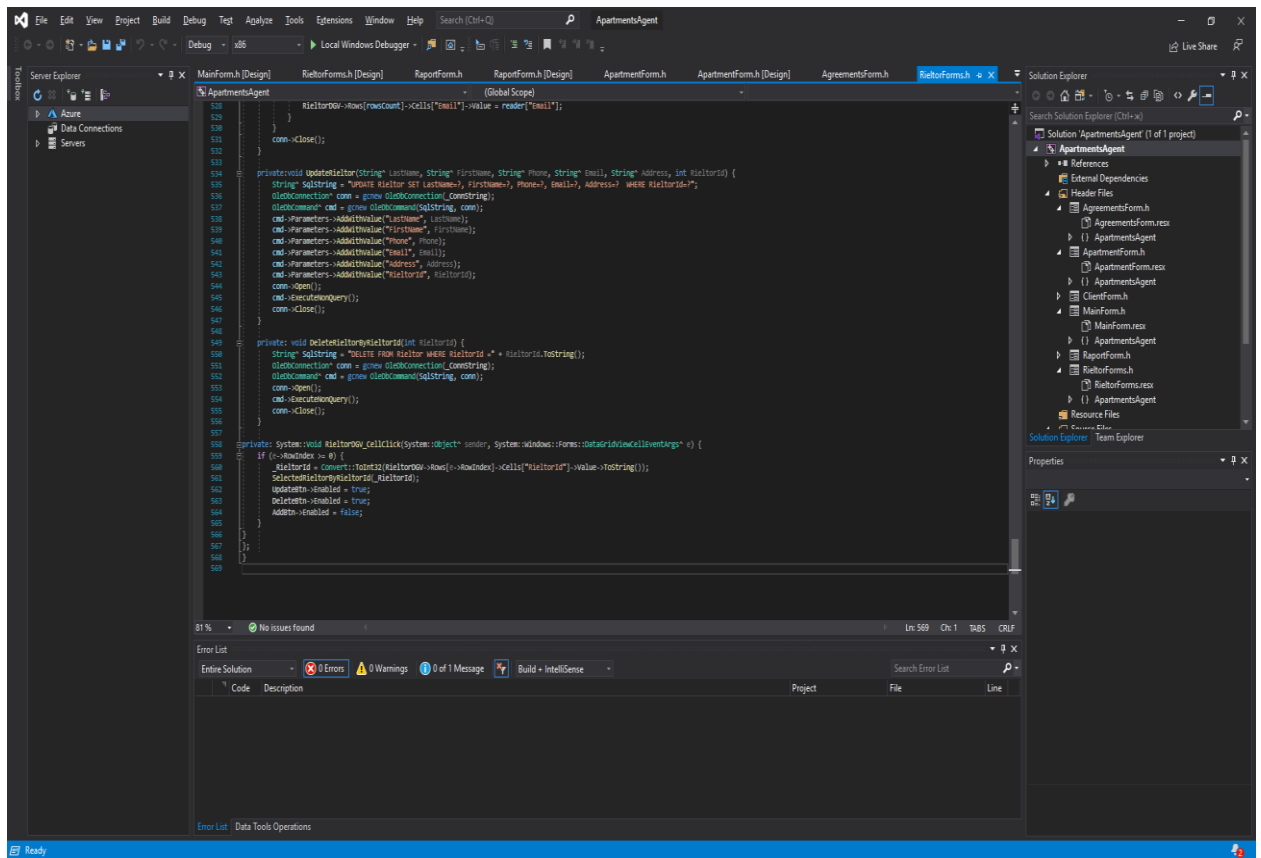


Рисунок 3.2 – Інтерфейс середовища розробки Visual Studio

Вбудований налагоджувач Visual Studio використовується для пошуку та виправлення помилок у вихідному коді, включаючи помилки низького рівня обладнання. Інструменти діагностики дозволяють оцінити якість вашого коду.

Літ	Арку	№ докум.	Підп	Дата

З точки зору продуктивності та використання пам'яті, Visual Studio Forms Designer дуже важливий при розробці додатків із графічним інтерфейсом, допомагаючи розробляти зовнішній вигляд кожної програми та роботу кожного елемента інтерфейсу.

Для командних проектів Visual Studio забезпечує підтримку групової роботи, що дозволяє спільно редагувати та налагоджувати будь-яку частину коду в режимі реального часу, а також систему контролю версій.

Використовуйте Team Foundation або Git.

Основним розширенням файлу, пов'язаним з Microsoft Visual Studio, є SLN – файл рішення Visual Studio, який відкриває всі дані та проекти, пов'язані з програмним рішенням, що розробляється.

Функціональна структура середовища включає:

- редактор вихідного коду з багатьма додатковими функціями, такими як автозавершення IntelliSense, рефакторинг коду тощо;
- налагоджувач коду;
- редактор форм, призначений для спрощення оформлення графічного інтерфейсу;
- веб-редактор;
- дизайнер класу;
- схема проектування бази даних.

Visual Studio також дозволяє створювати та підключати сторонні доповнення (плагіни), щоб розширити функціональність майже на кожному рівні, включаючи додавання підтримки систем контролю джерел (Subversion і VisualSourceSafe), додавання нових наборів інструментів для редагування та візуалізації тем дизайну. код мови.) програмування або інструменти, що використовуються в інших аспектах процесу розробки програмного забезпечення).

					IA82.010BAK.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		40

3.3 Переваги та недоліки середовища розробки

IntegratedDevelopmentEnvironment (IDE) Visual Studio надає багато розширених функцій

Функції за межами керування базовим кодом.

Ось основні переваги Visual Studio IDE:

- під час розробки підтримується кілька мов. Visual Studio дозволяє завжди використовувати той самий інтерфейс (IDE) для кодування вашою або будь-якою іншою мовою. Крім того, Visual Studio дозволяє створювати веб-сторінки різними мовами, але розміщувати їх усі в одному веб-додатку;

- менше коду для написання. Більшість програм потребує достатньої кількості стандартного стереотипного коду та сторінок ASP. NET не є винятком. Наприклад, для додавання веб-елементів керування, приєднання обробників подій та налаштування форматування потрібно вказати багато деталей у макеті сторінки. У Visual Studio ці відомості встановлюються автоматично;

- інтуїтивно зрозумілий стиль кодування. За замовчуванням Visual Studio форматує ваш код під час введення, автоматично вставляючи необхідні відступи та застосовуючи кольорове кодування для виділення елементів типу коментаря. Такі тонкі відмінності роблять код легшим для читання і менш схильним до помилок. Форматування, яке використовується Visual Studio, можна навіть автоматично коригувати, що корисно, коли розробники віддають перевагу іншому стилю дужок (наприклад, стиль K&R, де дужки розташовані в тому ж рядку, що й попередня реклама);

- вища швидкість розробки. Багато функцій Visual Studio призначені для того, щоб допомогти розробникам виконувати роботу якомога швидше. Такі зручні функції, як IntelliSense (який може виявляти помилки та запропонувати правильний варіант), пошук і заміна (дозволяють знаходити ключові слова в окремому файлі або в цілому проекті), а також автоматичне

додавання та видалення коментарів (що може тимчасово приховати блоки код), дозволити персоналу розробників працювати швидко та ефективно;

- можливості налагодження. Пропоновані Visual Studio інструменти налагодження є найкращим засобом для відстеження загадкових помилок і діагностування дивної поведінки. Розробники можуть виконувати власний код рядок за рядком, встановлювати розумні точки зупинки, зберігати для подальшого використання та переглядати поточну інформацію в пам'яті в будь-який час;

- висока швидкість створення програм з графічним інтерфейсом для Microsoft Windows;

- простий синтаксис мови забезпечує низький поріг входження;
- можливість компіляції як у машинний код, так і в Р-код (на вибір програміста);

- безпека типів забезпечує захист від помилок, пов'язаних із застосуванням покажчиків та доступом до пам'яті. Цей аспект робить Visual Basic програми більш стабільними, але є об'єктом критики;

- можливість використання більшості функцій WinAPI[8] для розширення функціональних можливостей програми. Це питання найбільш повно досліджено Деном Епплманом, який написав книгу "Visual Basic Programmer's Guide to the Win32 API".

Недоліки:

- підтримка операційних систем лише сімейства Windows та Mac OS X (Виняток - VB1 for DOS);

- відсутність повноцінного механізму наслідування реалізації об'єктів [9]. Існуюче у мові успадкування дозволяє успадковувати лише інтерфейси, але з їх реалізацію;

- практично всі вбудовані функції мови реалізовані через бібліотеку часу виконання, яка, своєю чергою, робить багато «додаткової» роботи з

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		42

перевірки та/або перетворення типів, що уповільнює швидкість роботи програм.

3.4 Розробка програмних модулів системи

Для ефективної розробки додатку процес було розділено на два етапи.

Перший етап – створення програмної частини, що стосується ролі «викладач», другий етап – ролі «студент».

Після проходження аутентифікації користувачем, система знаючи, яка в нього роль відкриє функціонал програми відповідно до неї.

Для авторизації в системі необхідно вибрати ім'я користувача та ввести пароль (рис. 3.3).

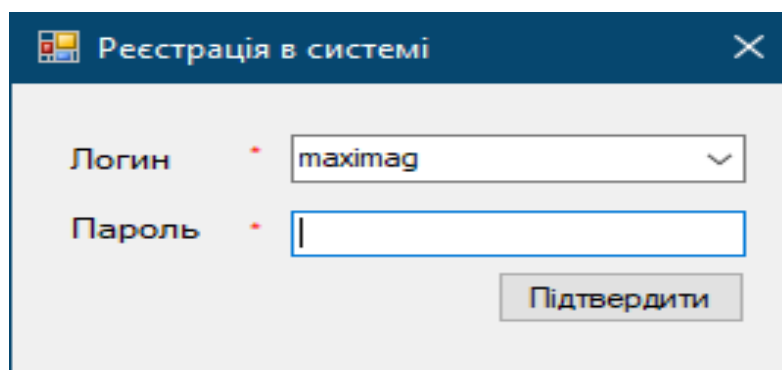


Рисунок 3.3 – Форма авторизації користувача в системі

Для перевірки ролі користувача, була реалізовано два методи програми: GetSubmitData (рис 3.4) та DataLoad (рис 3.5).

```
1 reference
private void GetSubmitData() {
    try {
        if (IsDataEnteringCorrect()) {
            List<Users> listUsers = new List<Users>();
            listUsers = _UserProvider.SelectedUsersByUserNameAndPassword(UserNameCBox.Text, UserPasswordTbx.Text);
            if (listUsers.Count > 0) {
                CurrentUser = listUsers[0];
                DataLoad();
            } else {
                MessageBox.Show(NamesMy.MessageBoxExaption.ThisUserLoginAndUserPasswordNotExistInSystem, NamesMy.MessageBoxExaption.CaptionMessage);
            }
        }
    } catch {
        MessageBox.Show("Немає з'єднання!");
    }
}
```

Літ	Арку	№ докум.	Підп	Дата
-----	------	----------	------	------

Рисунок 3.4 – Код методу «GetSubmitData»

Метод «GetSubmitData» викликається після натискання клавіші «Підтвердити» користувачем. Після чого, метод SelectedUsersByUsersNameAndUsersPassword повертає список всіх користувачів із вибраним ім'ям користувача та введеним паролем. Якщо такого користувача система не знайде, тоді буде виведено повідомлення "Користувача з таким ім'ям або паролем не існує в системі". Інакше, в залежності від того, яка роль буде у вибраного користувача, система виведе відповідне вікно. Код цього методу приведений на рис. 3.5.

```
1 reference
private void DataLoad() {
    _LogsProvider.InsertLogs(CurrentUser.UsersId, "Користувач ввійшов в систему", DateTime.Now);
    this.Visible = false;
    if (CurrentUser.RoleId == 1) {
        (new TeacherMDI()).ShowDialog();
    } else {
        (new TestsForm()).ShowDialog();
    }
    _LogsProvider.InsertLogs(CurrentUser.UsersId, "Користувач вийшов із системи", DateTime.Now);
    this.Close();
}
```

Рисунок 3.5 – Код методу «DataLoad»

Вікно, що відкриється після аутентифікації у системі користувача із роллю «викладач» (рис 3.6).

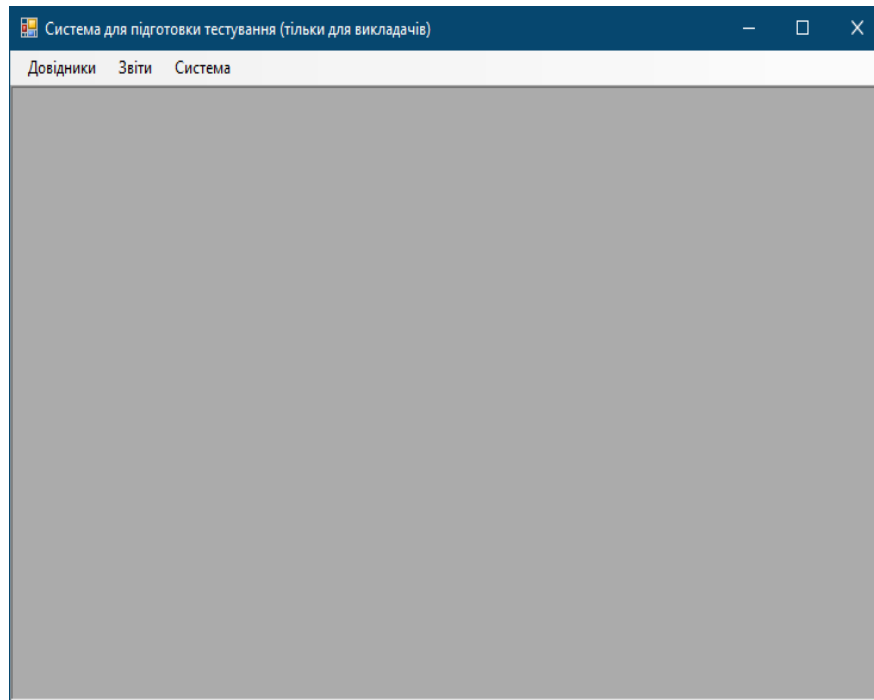


Рисунок 3.6 – Головне вікно програми для користувача із роллю «викладач»

Для користувача із роллю «студент» буде виведено вікно, де пропонується вибрати із списку дисципліну по якій буде проводитись тестування знань (рис. 3.7).

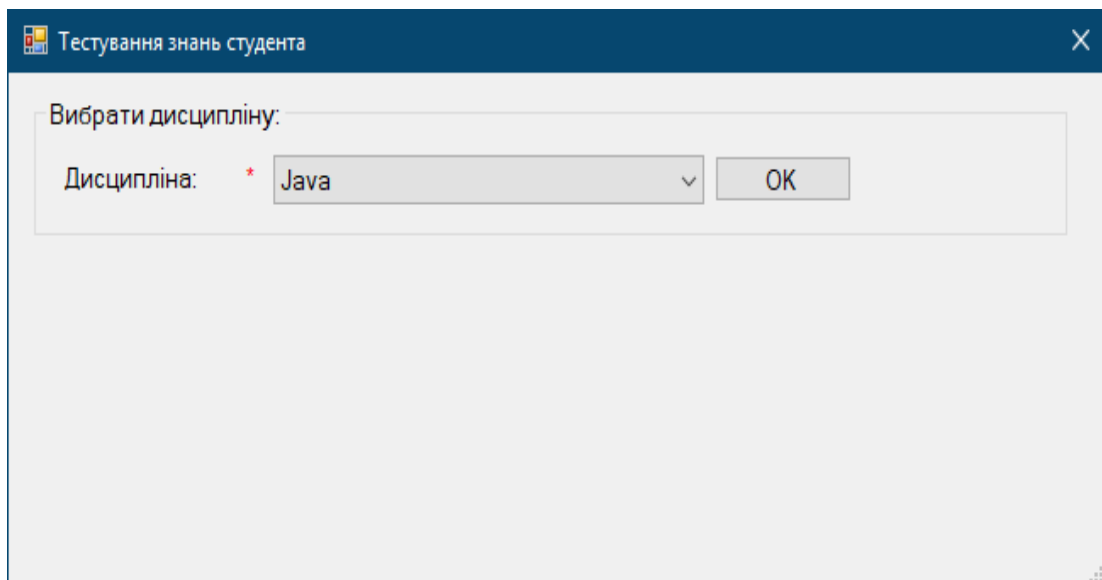


Рисунок 3.7 – Вікно програми для користувача із роллю «студент»

Після, того, як дисципліна вибрана, користувачу необхідно натиснути клавішу «ОК» для того, щоб програма розпочала тестування знань (рис 3.8).

Слід зазначити, що після натискання кнопки «ОК» піде таймер, який по завершенню заблокує доступ до тестування

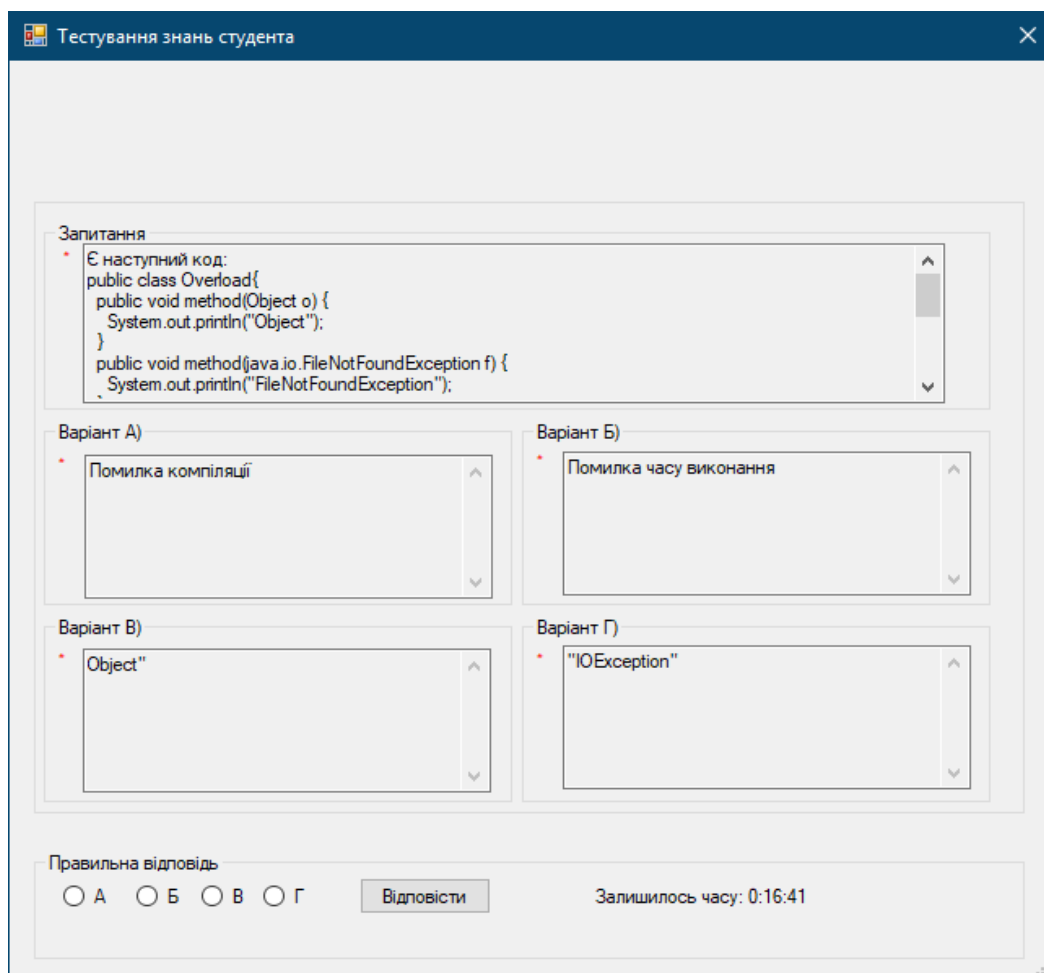


Рисунок 3.8 – Тестування знань

Як можна побачити, зверху вікна система виводить запитання та варіанти відповіді на нього, після вибору варіанту відповіді, користувачу необхідно натиснути кнопку «Відповісти», після чого система виведе наступне запитання. Код методу із перевіркою вибраної відповіді приведений на рисунку 3.9.

```

1 reference
private void AnsverBtn_Click(object sender, EventArgs e) {
    if (_AmountAnsver < _TestList.Count) {
        if (AnswerABool.Checked && AnswerABool.Checked == _SelectTest.AnswerABool) {
            _RightAnsver++;
        } else if (AnswerBBool.Checked && AnswerBBool.Checked == _SelectTest.AnswerBBool) {
            _RightAnsver++;
        } else if (AnswerCBool.Checked && AnswerCBool.Checked == _SelectTest.AnswerCBool) {
            _RightAnsver++;
        } else if (AnswerDBool.Checked && AnswerDBool.Checked == _SelectTest.AnswerDBool) {
            _RightAnsver++;
        }
        GetNextQustion();
    } else {
        StopTest();
    }
}
}

```

Рисунок 3.9 – Код події «AnsverBtn_Click»

Як можна побачити із коду, насамперед в ньому перевіряється кількість вже заданих питань відносно всіх питань системи. Якщо кількість заданих питань менша за кількість всіх запитань, тоді викликається метод «GetNextQustion», інакше метод «StopTest» (рис. 3.10-3.11).

```

2 references
private void GetNextQustion() {
    _SelectTest = _TestProvider.SelectedTestByTestId(_TestList[_AmountAnsver].TestId);
    if (_SelectTest.TestTypeId == 1) {
        QuestionNameTBox.Text = _SelectTest.QuestionName;
        AnswerATBox.Text = _SelectTest.AnswerA;
        AnswerBTBox.Text = _SelectTest.AnswerB;
        AnswerCTBox.Text = _SelectTest.AnswerC;
        AnswerDTBox.Text = _SelectTest.AnswerD;
        AddTextGBox.Visible = true;
        GraphicsGBox.Visible = false;
    } else {
        QuestionImagesPBox.Image = ByteToImage(_SelectTest.QuestionImages);
        AnswerAImagesPBox.Image = ByteToImage(_SelectTest.AnswerAImages);
        AnswerBImagesPBox.Image = ByteToImage(_SelectTest.AnswerBImages);
        AnswerCImagesPBox.Image = ByteToImage(_SelectTest.AnswerCImages);
        AnswerDImagesPBox.Image = ByteToImage(_SelectTest.AnswerDImages);
        AddTextGBox.Visible = false;
        GraphicsGBox.Visible = true;
    }
    _AmountAnsver++;
}
}

```

Рисунок 3.10 – Код методу «GetNextQustion»

```

2 references
private void StopTest() {
    StartTestTimer.Stop();
    TimeSpan timeQuest = DateTime.Now - _StartTest;
    int wrongAnswer = _TestList.Count - _RightAnswer;
    int notAnswer = _TestList.Count - _AmountAnswer;
    int lastStatisticId = 0;

    _StatisticsProvider.InsertStatistics(Convert.ToInt32(DisciplineCBox.SelectedValue), timeQuest.TotalSeconds, LoginForm.CurrentUser.UserId,
        _TestList.Count, _RightAnswer, wrongAnswer, notAnswer);
    lastStatisticId = _StatisticsProvider.GetLastRecords();
    _LogsProvider.InsertLogs(LoginForm.CurrentUser.UserId, "Тести по дисципліні " + _selectedDisciplines.DisciplinesName, DateTime.Now);
    for (int i = 0; i < _RunProgramsList.Count; i++) {
        _RunProgramsList[i].StatisticsId = lastStatisticId;
        _RunProgramsList[i].IsRun = true;
    }
    _ProgramsProvider.InsertBatchPrograms(_RunProgramsList);

    for (int i = 0; i < _StartProgramsList.Count; i++) {
        _StartProgramsList[i].StatisticsId = lastStatisticId;
    }
    _ProgramsProvider.InsertBatchPrograms(_StartProgramsList);

    string message = "Кількість правильних відповідей: " + _RightAnswer.ToString() + " з " + _TestList.Count.ToString() + " \r\n";
    MessageBox.Show(message);
    AddTextBox.Visible = false;
    GraphicsGBox.Visible = false;
    AnswerGBox.Visible = false;
    DisciplineGBox.Visible = true;
    _RightAnswer = 0;
}

```

Рисунок 3.11 – Код методу «StopTest»

Також, програма відслідковує запуснені програми до тестування знань та під час тестування знань. Всі ці дані програма записує у базу даних. Якщо користувач запускає сторонню програму, його буде про це негайно проінформовано відповідним повідомленням (рис. 3.12).

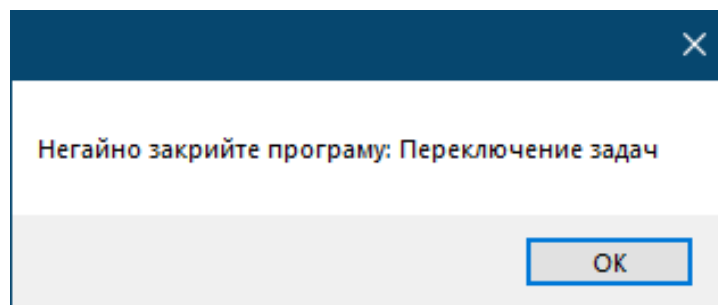


Рисунок 3.12 – Вікно, про запуснену програму

Час, скільки була відкрита програма, теж буде записано. В кінці проходження тестування програма виведе результат у діалоговому вікні (рис. 3.13).

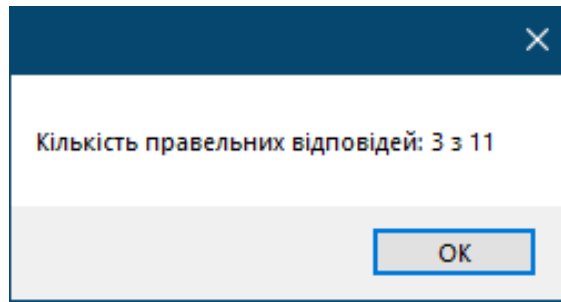


Рисунок 3.13 – Діалогове вікно із результатом тестування знань

В свою чергу користувач із роллю «викладач» має всій набір функціоналу системи, а саме:

- можливість додавання користувачів системи;
- можливість створення дисциплін;
- можливість перегляду логів системи;
- можливість створення тестування для кожної дисципліни;
- можливість формування звітності по пройдених тестах.

На рисунку 3.14 показано можливість створення нової дисципліни, по якій буде проводитись тестування.

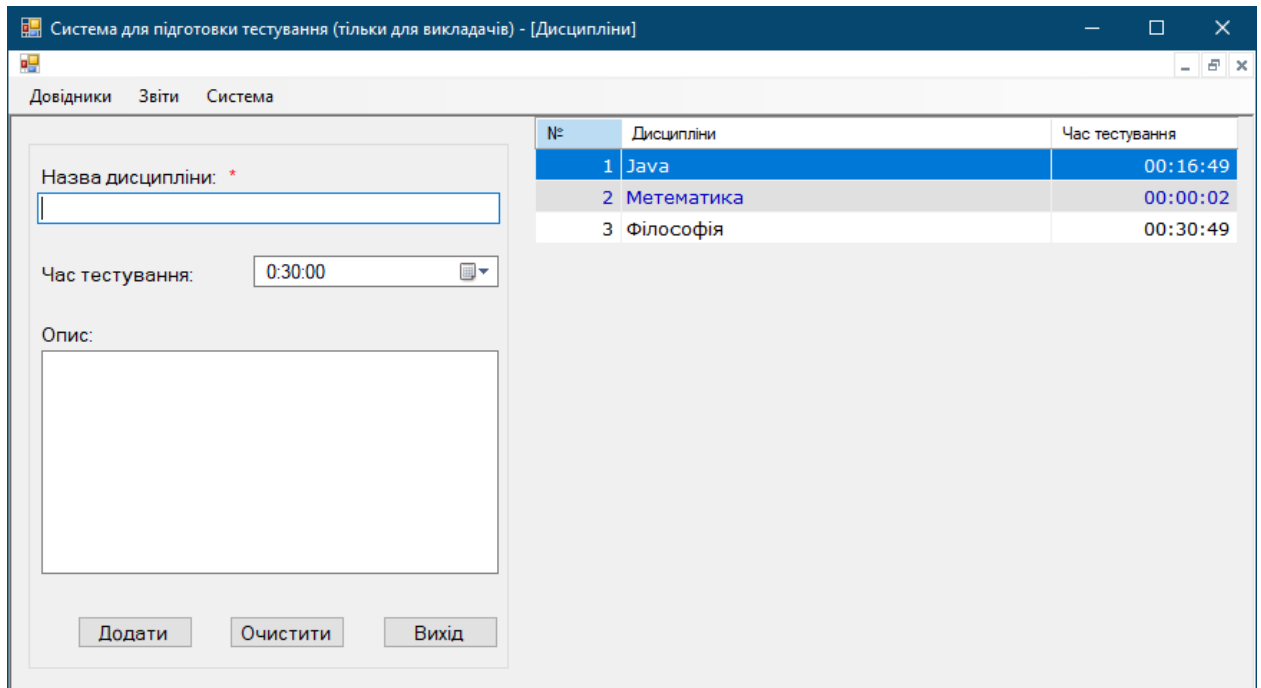


Рисунок 3.14 – Вікно для створення та редагування даних дисципліни

Для створення нової дисципліни було реалізовано подію із викликом методів, що приведені на рисунку 3.15.

```
1 reference
private void AddBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        _DisciplinesProvider.InsertDisciplines(DisciplinesNameTBox.Text, DescriptionTBox.Text, SetDateTimeDTP.Value);
        DataLoad();
        ClearAllControls();
    }
}
```

Рисунок 3.15 – Код події «AddBtn_Click»

Як можна побачити, перед занесенням даних в базу даних, за допомогою методу «IsDataEnteringCorrect» проводиться валідація введених даних користувачем програми. Код методу «IsDataEnteringCorrect» приведені на рисунку 3.16.

```
1 reference
private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (_validation.IsDataEntering(DisciplinesNameTBox.Text)) {
        DisciplinesNameValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        DisciplinesNameValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}
```

Рисунок 3.16 – Код методу «IsDataEnteringCorrect»

Також, якщо дані було введено із помилками, в програмі реалізована можливість їх редагування або видалення. Для цього необхідно в правій частині екрану вибрати відповідну дисципліну, після чого програма відкриє відповідне вікно (рис. 3.17).

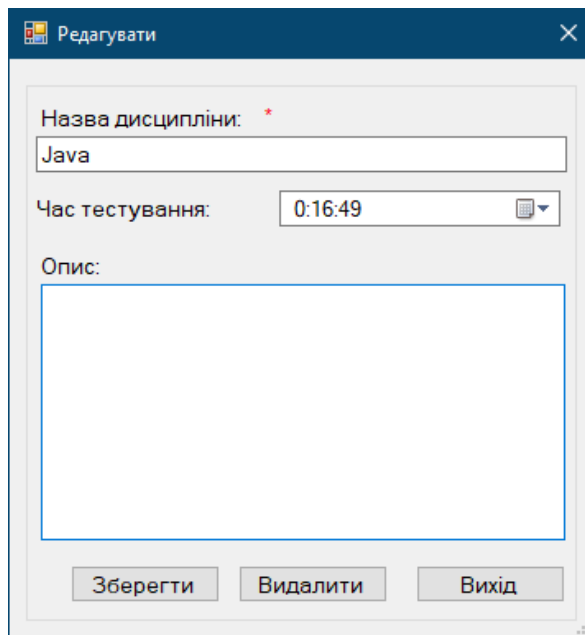


Рисунок 3.17 – Вікно для редагування даних дисципліни

Для додавання тестів по вибраній дисципліні необхідно відкрити відповідне вікно, для цього необхідно перейти по меню програми «Довідники» – > «Тести» (рис. 3.18).

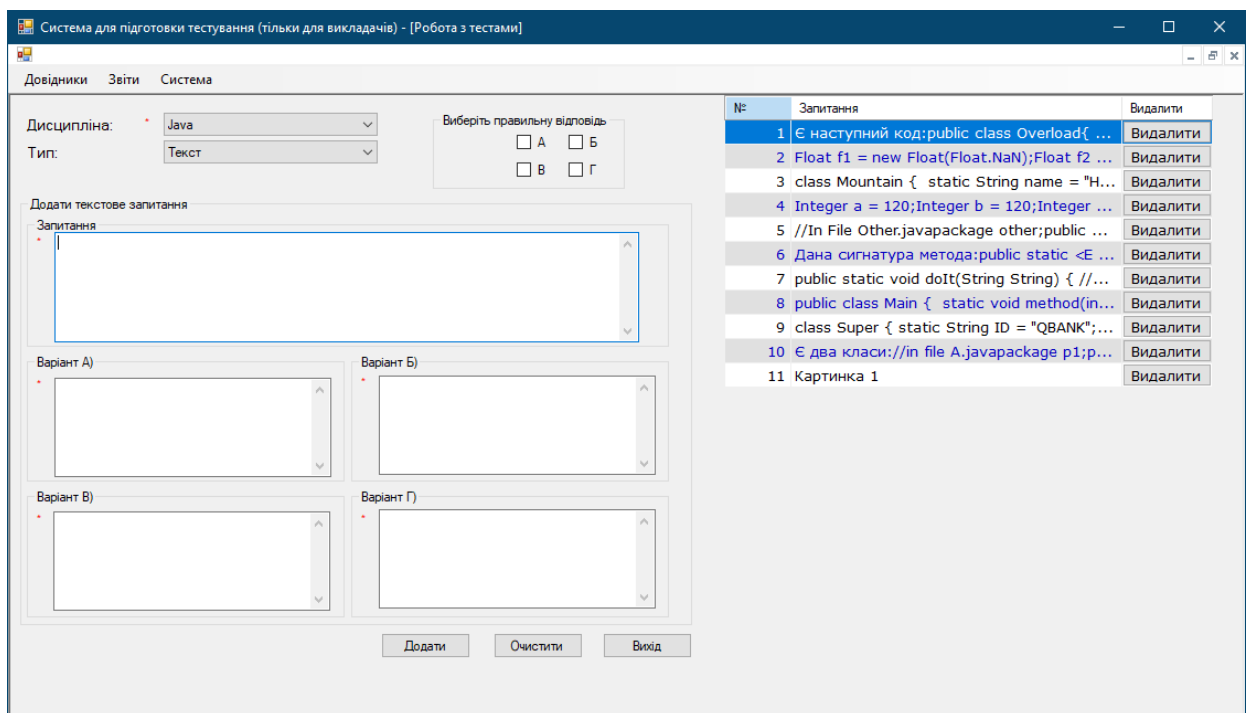


Рисунок 3.18 – Вікно для додавання тестів

Код додавання нового тесту приведений на рисунку 3.19.

```
1 reference
private void AddBtn_Click(object sender, EventArgs e) {
    if (Convert.ToInt32(TestTypeCBox.SelectedValue) == 1) {
        if (IsDataEnteringCorrect()) {
            _TestProvider.InsertTest(QuestionNameTBox.Text, AnswerATBox.Text, AnswerABoolCBox.Checked, AnswerBTBox.Text, AnswerBBoolCBox.Checked,
                AnswerCTBox.Text, AnswerCBoolCBox.Checked, AnswerDTBox.Text, AnswerDBoolCBox.Checked,
                Convert.ToInt32(TestTypeCBox.SelectedValue), Convert.ToInt32(DisciplineCBox.SelectedValue),
                _QuestionImages, _AnswerAImages, _AnswerBImages, _AnswerCImages, _AnswerDImages);
            DataLoad();
            ClearAllControls();
        }
    }
    else {
        if (IsDataEnteringCorrectImages()) {
            _TestProvider.InsertTest(QuestionNameTBox.Text, AnswerATBox.Text, AnswerABoolCBox.Checked, AnswerBTBox.Text, AnswerBBoolCBox.Checked,
                AnswerCTBox.Text, AnswerCBoolCBox.Checked, AnswerDTBox.Text, AnswerDBoolCBox.Checked,
                Convert.ToInt32(TestTypeCBox.SelectedValue), Convert.ToInt32(DisciplineCBox.SelectedValue),
                _QuestionImages, _AnswerAImages, _AnswerBImages, _AnswerCImages, _AnswerDImages);
            DataLoad();
            ClearAllControls();
        }
    }
}
}
```

Рисунок 3.19 – Код події «AddBtn_Click» для додавання нового тесту

Як можна побачити із коду програми: у ній окрім валідації даних на введення, є ще й можливість додавання картинок замість тексту. Ця можливість є досить зручною, оскільки дозволяє додавати окрім тексту ще й картинки.

Також, вважаю доцільним привести код реалізації події таймера (рис. 3.20).

```
1 reference
private void StartTestTimer_Tick(object sender, EventArgs e) {
    GetAllRunPrograms();
    TimeSpan ts = _EndDate - DateTime.Now;
    TimeLbl1.Text = "Залишилось часу: " + ts.Hours + ":" + ts.Minutes + ":" + ts.Seconds;
    if (DateTime.Now > _EndDate) {
        StopTest();
    }
}
}
```

Рисунок 3.20 – Код події «StartTestTimer_Tick»

Стоїть умова перевірки часу виконання тестування. Як тільки час тестування буде вичерпано, тестування буде зупинене.

3.5 Результати функціонального тестування розробленого додатку

Тестування — це процес аналізу та дослідження, який надає змогу виявити інформацію про якість продукту відносно умов, в яких він буде застосовуватись. Методика тестування також включає в себе процес пошуку дефектів, помилок, несправностей. Також це є випробуванням програмних складових з метою оцінити готовність програмного продукту до використання. Результат тестування оцінюється за наступними критеріями:

- відповідність вимогам, які надавалися розробниками та проектувальниками;
- відповідність вихідних даних;
- прийнятний час виконання функцій;
- практичність;
- відповідність вимогам замовника.

Кількість тестів навіть для простих програмних компонентів може бути ледь не нескінченним, тому тактика тестування має полягати в тому, що будуть проведені тільки необхідні тести з урахуванням доступного часу та ресурсів. Як результат, програмні засоби тестуються стандартним виконанням програми з метою виявлення багів, помилок або інших дефектів.

Існує багато видів тестування: одні зазвичай виконують самі розробники, а інші — спеціалізовані групи. В нашому ж випадку буде використовуватись тестування системи.

Тестування системи — це виконання програмного забезпечення в його остаточній конфігурації, інтегрованого з іншими програмними та апаратними системами.

Одним із способів вивчення поставленого питання є дослідження методики «чорної скриньки», Основне місце програми тестів «чорної скриньки» це інтерфейс ПЗ. Ці тести демонструють:

- як виконуються функції програми;
- як приймаються вихідні дані;
- як виробляються результати;
- як зберігається цілісність зовнішньої інформації.

При тестуванні «чорного ящика» враховуються системні характеристики програми та ігнорується внутрішня логічна структура програми. Комплексне тестування зазвичай неможливе. Наприклад, якщо програма мала 10 вхідних значень, кожне з яких приймає 10 значень, кількість варіантів тесту буде 10¹⁰. Багато функцій тестового «чорного ящика» не реагують на помилки програмного забезпечення.

Доречним є тестування програмного забезпечення з використанням методів «чорного ящика».

Вона базується на використанні шаблонів тестування або ж тест-кейсів. Це означає, що буде створено декілька ситуацій у яких перевіряється працездатність додатку, коректності основних функцій.

Таблиця 3.1 – Тест-кейси

Код тест-кейса	Опис тест-кейса		
	Хід тестування		Очікуваний результат
	Дата тестування	Результат	Примітка
001	Перевірка реєстрації користувача програми		
	1. Запустити додаток 2. Провести автентифікацію користувача із роллю «викладач».		Вивід відповідного вікна із функціоналом для роботи користувача
	22.04.2022	Пройдено	–
002	Створення нової дисципліни		

	<ol style="list-style-type: none"> Запустити додаток Провести автентифікацію Додати дисципліну Редагувати дані дисципліни Видалити дисципліну 	Список дисциплін відображається відповідно до введених користувачем даних	
	22.04.2022	Пройдено	—
003	Створення нового тесту, по якому буде проводитись тестування		
	<ol style="list-style-type: none"> Запустити додаток Провести автентифікацію Додати тест Редагувати дані тесту Видалити тест 	Список тестів відображається відповідно до введених користувачем даних	
	22.04.2022	Пройдено	—
004	Перевірка можливості виведення рейтингів тестування		
	<ol style="list-style-type: none"> Запустити додаток Провести автентифікацію Виведення рейтингів тестування користувачів програми 	Успішне виведення рейтингів проведення тестування	
	22.04.2022	Пройдено	—

3.6 Інструкція користувачеві програми

3.6.1 Опис процедури розгортання програмного продукту, створеного на платформі .NET

Важливим фактором, який необхідно врахувати при розробці програмного забезпечення є потреба в ресурсах.

Мінімальні вимоги до апаратних засобів для запуску та функціонування розробленого програмного забезпечення:

- процесор PentiumIV/Xeon2.4ГГц;
- оперативна пам'ять: 1024 Мб вище;
- вільний дисковий простір менше 120Мб;
- миша;
- клавіатура;
- монітор.

Вимоги до програмних засобів:

- Операційна система сімейства Windows: починаючи з Windows XP – до Windows 10;
- наявність бібліотеки класів .NET framework 4.0 або вище.

Розгортання програмного продукту на комп'ютері користувача у вигляді автономного застосунку:

- Створіть на цільовому диску каталог для застосунку, наприклад «Тестування знань студентів».
- Скопіюйте каталог «Debug» із проекту.
- Для перевірки коректності запуску програми виконайте подвійне клацання лівою кнопкою миші на файлі «TestSystem.exe» (операційна система Windows).

3.6.2 Використання програмного продукту

Запуск програми.

Запуск програми в операційній системі сімейства Windows здійснюється одним з стандартних способів:

- подвійним клацанням лівою кнопкою миші на ярлику програми;
- викликом контекстного меню з вибором його пункту "Відкрити";
- натисканням кнопки "Пуск" панелі завдань із подальшим вибором пункту "Усі програми" та подвійним клацанням лівою кнопкою миші на ярлику програми.

Вхід користувача в систему

На початку необхідно запусити додаток «Система для підготовки тестування». ». Після запуску програми буде виведено вікно, де користувачу буде запропоновано ввести ім'я та пароль.

Після успішної автентифікації, користувач в залежності від його ролі відкриється відповідний функціонал програми.

Якщо ж була вибрана роль «викладач», тоді програма такі надає такі можливості:

- додавання користувачів системи;
- створення дисциплін;
- перегляд логів системи;
- створення тестування для кожної дисципліни;
- формування звітності по пройдених тестах.

Якщо ж було вибрано роль «студент», тоді програма виведе вікно для тестування знань. В цьому вікні є можливість вибрати дисципліну, та відповідно до неї прийти тестування. В кінці тестування програма виведе його результат.

Однією із важливих функцій програми є можливість створення рейтингів по результатах тестування. Також програма дозволяє переглянути

всі програми, що були запущені користувачем, який проходив тестування до початку тесту та під час його проходження.

Отже, зайшовши в систему під роллю «викладач», користувачу необхідно для початку заповнити інформаційну систему даними про дисципліни по яких буде створюватись тестування, а також і самі тести, що відносяться до цих дисциплін.

Для того, щоб додати нову дисципліну, користувачу необхідно перейти по меню програми «Довідники» → «Дисципліни» та ввівши назву дисципліни і встановивши «Час тестування» необхідно натиснути кнопку «Додати». Після чого, програма доавить нову дисципліну. На рисунку 3.21 показано можливість додавання нової дисципліни.

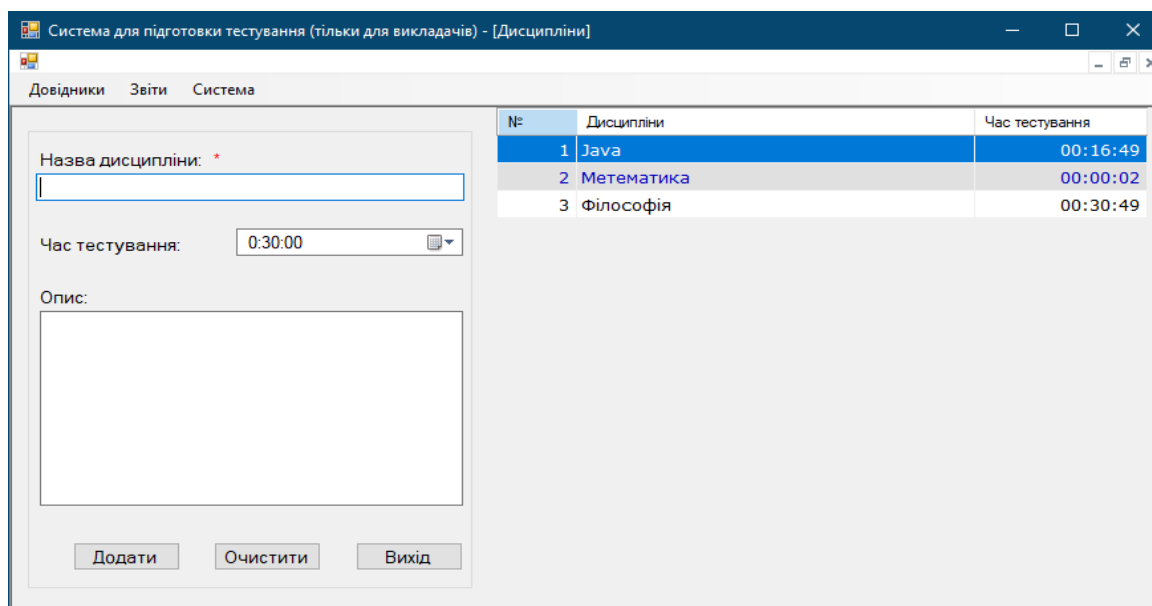


Рисунок 3.21 – Вікно для створення та редагування даних дисципліни

Після того, як дисципліна була додана, користувачу необхідно додати тести для цієї дисципліни. Для того, щоб додати нову дисципліну, користувачу необхідно перейти по меню програми «Довідники» → «Тести». Зверху вікна розташований випадаючий список із назвою дисципліни. В ньому можна вибрати ту дисципліну, до якої будуть додаватись тести. Під час зміни

дисципліни у правому вікні будуть відображатись тільки ті тести, що відносяться до вибраної дисципліни. Запитання до тестів можна додавати, як і в текстовому, так і в графічному виді (у вигляді картинок). На рисунку 21 та 22 зображено праву частину вікна, де можна додати відповідне до дисципліни питання із варіантами відповідей.

The screenshot shows a window titled "Система для підготовки тестування (тільки для викладачів) - [Робота з тестами]". The window has a menu bar with "Довідники", "Звіти", and "Система". Below the menu bar, there are two dropdown menus: "Дисципліна:" with "Java" selected and "Тип:" with "Текст" selected. To the right, there is a section titled "Виберіть правильну відповідь" with four radio buttons labeled "А", "Б", "В", and "Г". Below this, there is a section titled "Додати текстове запитання" containing a large text area for the question, and four smaller text areas for options labeled "Варіант А)", "Варіант Б)", "Варіант В)", and "Варіант Г)". At the bottom of the window, there are three buttons: "Додати", "Очистити", and "Вихід".

Рисунок 3.22 – Вікно для додавання текстового запитання до вибраної дисципліни

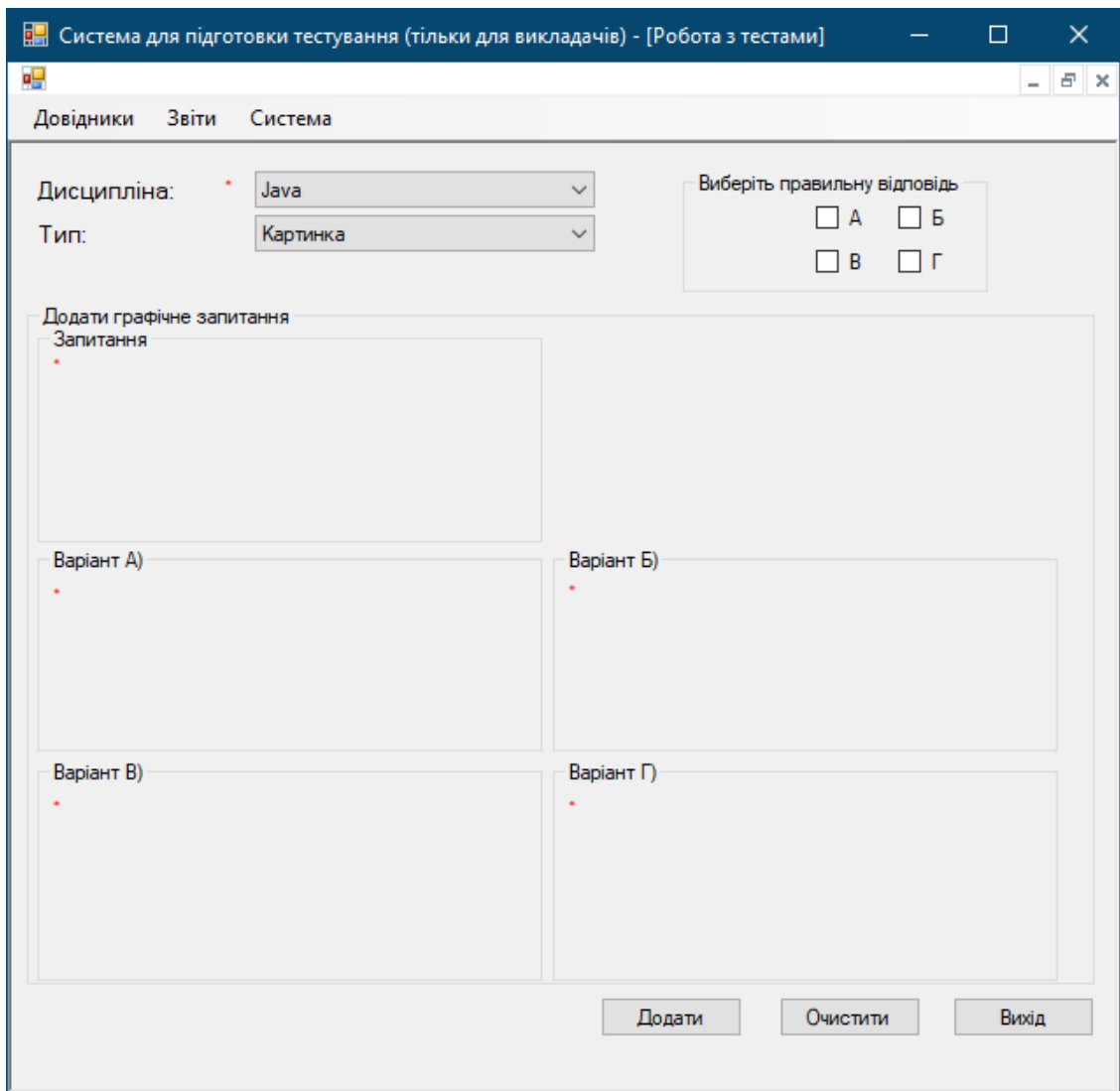


Рисунок 3.23 – Вікно для додавання графічного запитання до вибраної дисципліни

Для користувача з роллю «викладач» реалізована можливість переглядання звітності студентів по пройдених тестах. Для цього йому необхідно перейти по меню програми «Звіти» → «Рейтинги», де відкриється вікно з рейтингами (рис. 3. 24).

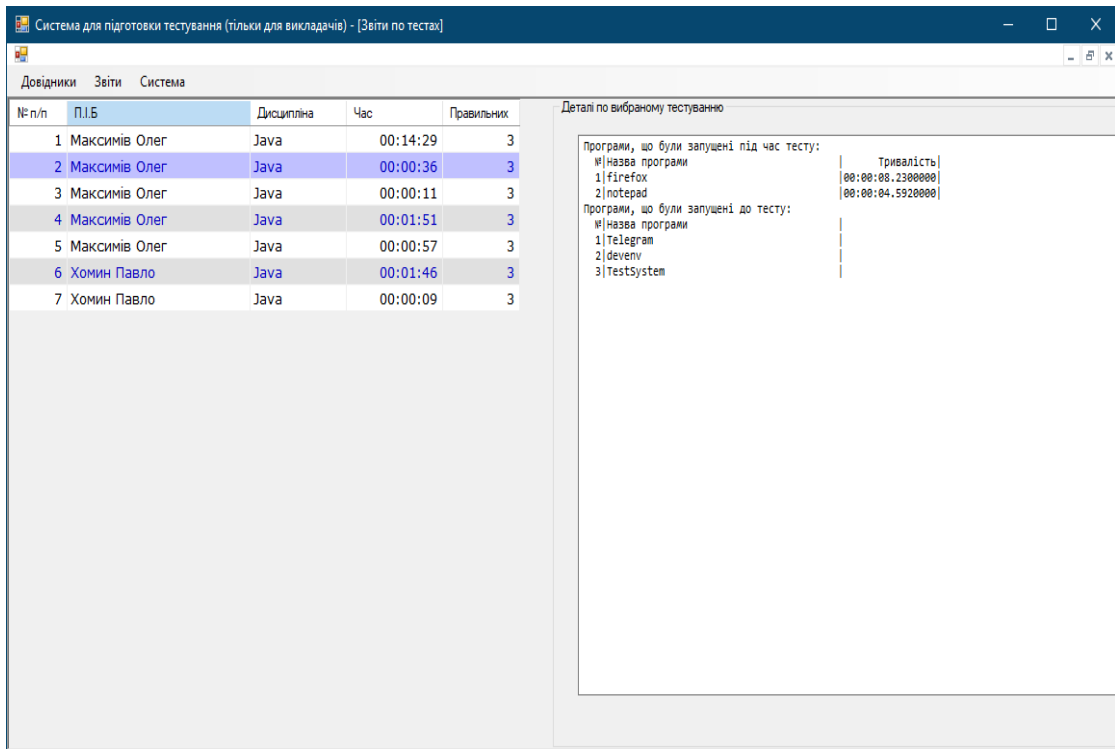


Рисунок 3.24 – Вікно для виведення рейтингів проходження тестування

Вибираючи в лівій частині вікна необхідний запис, методом натискання лівою кнопкою миші по необхідному запису, в правій можна побачити деталі тестування. В деталях по вибраному тестуванню можна побачити, які саме програми були запущені до початку тестування, а також програми, що були запущені під час тестування. Також, відображається час, скільки пропрацювали ці програми. Треба сказати, що розроблена система не фіксує запуск таких програм, як: Visual Studio, Microsoft Word та Microsoft Excel. Це зроблено з метою, щоб дозволити студенту користуватись цими інструментами.

Користувач із роллю «викладач» з правами адміністратора може створювати користувачів системи і надавати їм роль в системі тестування. Для цього необхідно перейти по головному меню «Система» → «Користувачі». Щоб додати нового користувача системи необхідно заповнити всі обов'язкові поля, а саме: прізвище, ім'я, логін та пароль (рис. 3.25). Дуже важлива, щоб поля пароля та підтвердження пароля співпадали, якщо ж цього не буде –

програма не дозволить додати нового користувача в систему.

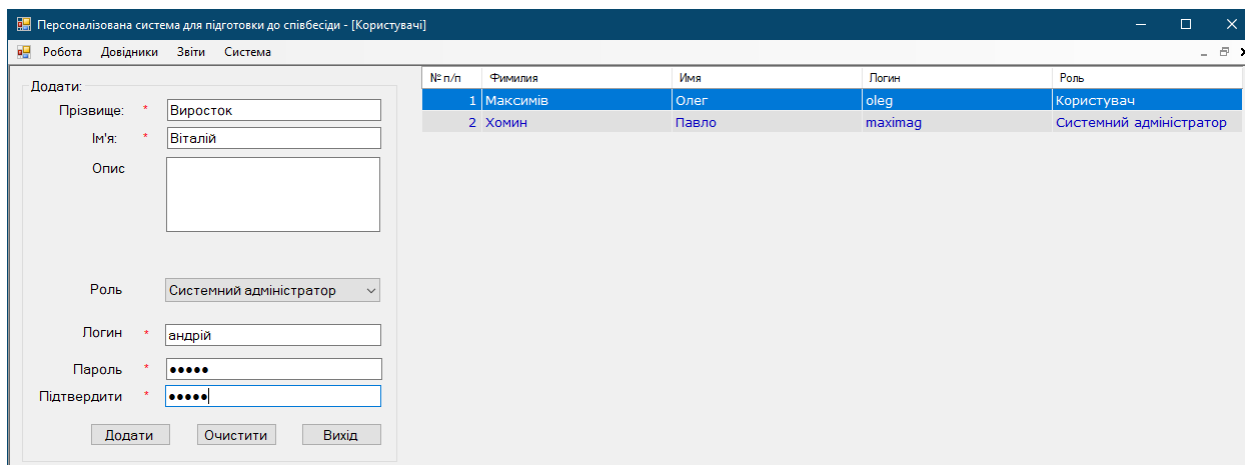


Рисунок 3.25 – Додавання нового користувача в систему

Також дані можна редагувати в разі необхідності. Для цього можна вибрати необхідний із списку обліковий запис на натиснути на нього. Після чого відкриється відповідне вікно для редагування даних (рис. 3.26).

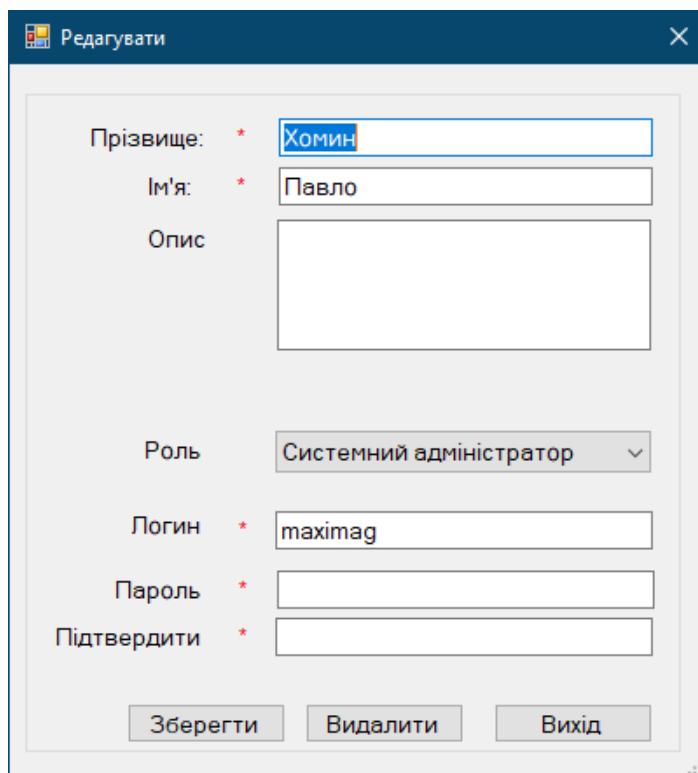


Рисунок 3.26 – Редагування даних користувача системи

Якщо ж поля «Пароль» та «Підтвердити» не співпадають, програма виведе повідомлення про це (рис. 3.27).

Редагувати

Прізвище: * Максимів

Ім'я: * Олег

Опис

Роль: Студент

Поля пароль і підтвердити не співпадають

Логин * oleg

Пароль * ●

Підтвердити !!!

Зберегти Видалити Вихід

Рисунок 3.27 – Пароль» та «Підтвердити» не співпадають

Також в системі можна бачити активність користувачів, та те, що вони робили в системі. Це можливо зробити за допомогою облікового запису, який має права системного адміністратора. Для цього перейдемо по меню «Система» → «Системний журнал». В цьому вікні виводяться всі події системи.

Система для підготовки тестування (тільки для викладачів) - [Системний журнал]			
Довідники Звіти Система			
№	Користувач	Подія	Дата
1	maximag	Користувач ввійшов в систему	22.04.2022 13:00
2	maximag	Користувач вийшов із системи	17.04.2022 9:48
3	maximag	Користувач ввійшов в систему	17.04.2022 9:38
4	maximag	Користувач ввійшов в систему	17.04.2022 9:29
5	oleg	Користувач вийшов із системи	17.04.2022 9:29
6	oleg	Тести по дисципліні Java	17.04.2022 9:23
7	oleg	Користувач ввійшов в систему	17.04.2022 9:06
8	oleg	Користувач вийшов із системи	17.04.2022 8:47
9	oleg	Користувач ввійшов в систему	17.04.2022 8:47
10	maximag	Користувач вийшов із системи	17.04.2022 8:47
11	maximag	Користувач ввійшов в систему	17.04.2022 8:47
12	maximag	Користувач вийшов із системи	17.04.2022 8:46
13	maximag	Користувач ввійшов в систему	17.04.2022 8:46
14	maximag	Користувач вийшов із системи	17.04.2022 8:45
15	maximag	Користувач ввійшов в систему	17.04.2022 8:44
16	maximag	Користувач вийшов із системи	16.04.2022 16:42
17	maximag	Користувач ввійшов в систему	16.04.2022 16:42
18	oleg	Користувач вийшов із системи	16.04.2022 16:42
19	oleg	Тести по дисципліні Java	16.04.2022 16:42
20	oleg	Користувач ввійшов в систему	16.04.2022 16:41
21	maximag	Користувач вийшов із системи	16.04.2022 16:41
22	maximag	Користувач ввійшов в систему	16.04.2022 16:41
23	maximag	Користувач вийшов із системи	16.04.2022 16:39
24	maximag	Користувач ввійшов в систему	16.04.2022 16:38
25	oleg	Користувач вийшов із системи	16.04.2022 16:38
26	oleg	Тести по дисципліні Java	16.04.2022 16:38
27	oleg	Користувач ввійшов в систему	16.04.2022 16:38
28	oleg	Користувач вийшов із системи	16.04.2022 16:37
29	oleg	Користувач ввійшов в систему	16.04.2022 16:37
30	oleg	Користувач вийшов із системи	16.04.2022 16:37
31	oleg	Користувач ввійшов в систему	16.04.2022 16:37
32	maximag	Користувач вийшов із системи	16.04.2022 15:23
33	maximag	Користувач ввійшов в систему	16.04.2022 15:23
34	maximag	Користувач вийшов із системи	16.04.2022 15:23
35	maximag	Користувач ввійшов в систему	16.04.2022 15:22
36	maximag	Користувач вийшов із системи	16.04.2022 15:22
37	maximag	Користувач ввійшов в систему	16.04.2022 15:21
38	maximag	Користувач вийшов із системи	16.04.2022 15:21
39	maximag	Користувач ввійшов в систему	16.04.2022 15:21
40	maximag	Користувач ввійшов в систему	16.04.2022 15:19
41	maximag	Користувач вийшов із системи	16.04.2022 15:19
42	maximag	Користувач ввійшов в систему	16.04.2022 15:18
43	maximag	Користувач вийшов із системи	16.04.2022 15:18

Рисунок 3.28 – Вікно «Системний журнал»

Треба сказати, що дана система є не сильно функціональною, але вона є досить простою в користуванні. Її інтерфейс є інтуїтивно зрозумілим для користувача.

Тестування програми успішно проведено та не було виявлено жодних помилок системи.

3.7 Висновок до розділу

В ході виконання роботи мовою C# в середовищі Visual Studio 2019 реалізовано персоналізовану систему для тестування знань студентів.

Реалізовано такі функціональні вимоги:

- можливість додавати, редагувати та видаляти облікові записи користувачів;
- можливість додавати, редагувати та видаляти дані про дисципліни, по яких будуть проводитись тестування;
- можливість додавати, редагувати та видаляти дані про запитання та варіанти відповіді на них (у вигляді тексту та картинок);
- можливість проходження тестування студентами;
- фіксується список програм, які були запущені до та під час тестування знань студента;
- фіксування активності користувачів системи. Ведуться логи.
- можливість формування звітності про успішність проходження тестування знань студентами.

ВИСНОВКИ

Дана дипломна робота присвячена розробці автоматизованої системи тестування знань та вмінь студентів для різних дисциплін.. В роботі було проведено розгляд аналогічних системи тестування знань. Вибір фреймворків було аргументовано тим, що .NET 4.7 має достатньо обширні можливості у створенні віконних додатків. Так як .NET працює на мовах C, то вибір мови програмування було завершено на C#.

У ході виконання роботи було проаналізовано процес проходження тестування. Були розглянуті можливі проблеми та помилки проведення тестування, причини виникнення та способи їхнього відстеження.

У першій частині роботи було проведено аналітичний огляд існуючих програм для тестування знань студентів. Виявлення їх основних переваг та недоліків для подальшої розробки власної програми, яка буде відповідати усім вимогам.

У другій частині роботи описано архітектуру системи проведення тестування знань студентів. Зроблено аналіз вимог до програмного забезпечення та побудовано use-case діаграми основних прецедентів.

У третій частині було обрано технології, на основі яких було розроблено програмний застосунок для тестування знань студентів. Проведено тестування розробленого продукту та аналіз отриманих результатів. Також розроблено інструкцію користувача системи, за якою можна використовувати всі основні функції програмного застосунку. Результати тестування були успішними, помилок не було виявлено.

У процесі вирішення завдання розроблено інженерну методику автоматизованої процедури проведення тестування знань студентів, і таким чином поставленої мети досягнуто. У ході досліджень отримано такі основні наукові та практичні результати:

					IA82.010BAK.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		66

1. Розроблено гнучку систему, призначену для тестування знань студентів.

2. Розроблені методи проведення тестування знань студентів можуть бути використані для широкого класу завдань, тому можливий подальший розвиток розробленого програмного забезпечення.

3. Дана система не є вимогливою і її можна переносити на будь-який комп'ютер із операційною системою сімейства Windows.

4. Розроблена система є безкоштовною.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		67

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тестування. Тести, їхня класифікація. URL:
<https://www.psychologos.ru/articles/view/testirovanie>
2. Композиція тестових завдань. URL:
https://dspace.nau.edu.ua/bitstream/NAU/48976/1/Калініченко_стаття
3. Програма для тестування SunRay. URL:
<https://www.softportal.com/devsoft-717-1.html>
4. Програма для створення тестів та онлайн тестування. URL:
<https://indigotech.ru>
5. Конструктор тестів. URL:
<https://www.keepsoft.ru/simulator/download.php>
6. Pro C# 7: With .NET and .NET Core 8-th edition / Andrew Troelse, Philip Japikse. –2017
7. Об'єктно-орієнтоване програмування. URL:
http://ruslan.rv.ua/python-essential/oop/oop_basis/
8. Об'єктно-орієнтований підхід до програмування. URL:
<http://www.znannya.org/?view=csharp-oop>
9. Офіційна документація .NET. URL:
<https://docs.microsoft.com/ru-ru/dotnet/>
10. Мова програмування C# і платформа . URL:
<https://metanit.com/sharp/tutorial/1.1.php>
11. What's new in .NET 4.7. URL:
<https://docs.microsoft.com/en-us/dotnet/core/dotnet-five>
12. .NET Core/5+ vs. .NET Framework for server apps. URL:
<https://docs.microsoft.com/en-us/dotnet/standard/choosing-core-framework-server>
13. .NET 4.7. URL: <https://temofeev.ru/info/articles/predstavlyaem-net-5/>
14. Джон Скит, C# для професіоналів: тонкощі програмування, 608 ст. ISBN 978-5-8459-1909-0, 978-1-617-29134-0; 2014, Вільямс.

					ІА82.010БАК.003 ПЗ	Арку
Літ	Арку	№ докум.	Підп	Дата		68

15. Литвинов О.А., Карпенко Н.В. Тестування інформаційних систем: модульне, інтеграційне, системне [Текст] – Д.: Ліра, 2016. – 283 с.

16. Литвинов О.А., Герасимов В.В., Карпенко Н.В. Об'єктно-орієнтована розробка інформаційних систем [Текст] – Д.: Ліра, 2018. – 448 с.

17. Програма для роботи з базами даних Microsoft Access. URL:
<https://www.microsoft.com/uk-ua/microsoft-365/access>.

18. Побудова ER-діаграм. URL:
<https://app.diagrams.net/>.

19. Архітектурні шаблони. URL:
<https://devzone.org.ua/post/nayvazhlivishi-arkhitekturni-shablони-yaki-neobkhidno-znati>.

20. Тестування. Тести, їхня класифікація. URL:
<https://www.psychologos.ru/articles/view/testirovanie>.

ДОДАТОК А

Лістинг програми

Лістинг класу «RoleApp»

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace TestSystem.AppCode {
    class RoleApp {
        public List<Role> GetRoleList() {
            List<Role> RoleList = new List<Role>();
            RoleList.Add(new Role(1, "Викладач"));
            RoleList.Add(new Role(2, "Студент"));
            return RoleList;
        }
    }
}
public class Role {
    private int _RoleId;
    private string _RoleName;
    public Role() {
        _RoleId = 0;
        _RoleName = String.Empty;
    }
    public Role(int RoleId, string RoleName) {
        _RoleId = RoleId;
        _RoleName = RoleName;
    }
    public int RoleId {
        set { _RoleId = value; }
        get { return _RoleId; }
    }
    public string RoleName {
        set { _RoleName = value; }
        get { return _RoleName; }
    }
}
```

Лістинг класу «StatisticBLL»

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using TestSystem.Providers;
namespace TestSystem.BLL {
class StatisticBLL {
    private StatisticsProvider _StatisticProvider = new StatisticsProvider();
    private DisciplinesProvider _DisciplinesProvider = new DisciplinesProvider();
    private UsersProvider _UsersProvider = new UsersProvider();
    public List<Statistics> GetAllStatistic() {
        List<Users> usersList = new List<Users>();
        List<Disciplines> disciplinesList = new List<Disciplines>();
        usersList = _UsersProvider.GetAllUsers();
        disciplinesList = _DisciplinesProvider.GetAllDisciplines();
        List<Statistics> statisticList = new List<Statistics>();
        statisticList = _StatisticProvider.GetAllStatistic();
        for (int i = 0; i < statisticList.Count; i++) {
            statisticList[i].FIO = GetFIO(statisticList[i].UsersId, usersList);
            statisticList[i].DisciplineName = GetDisciplineName(statisticList[i].DisciplineId, disciplinesList);
            statisticList[i].RunTime = TimeSpan.FromSeconds(statisticList[i].Seconds);
        }
        return statisticList;
    }
    private string GetFIO(int UsersId, List<Users> UsersList) {
        for (int i = 0; i < UsersList.Count; i++) {
            if (UsersId == UsersList[i].UsersId) {
                return UsersList[i].FIO;
            }
        }
        return "";
    }
    private string GetDisciplineName(int DisciplineId, List<Disciplines> DisciplinesList) {
        for (int i = 0; i < DisciplinesList.Count; i++) {
            if (DisciplineId == DisciplinesList[i].DisciplinesId) {
                return DisciplinesList[i].DisciplinesName;
            }
        }
        return "";
    }
}
}

```