

Київський національний університет імені Тараса Шевченка  
Факультет інформаційних технологій  
Кафедра програмних систем і технологій

УДК 004.942

На правах рукопису

# ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Прототипування навчального ПЗ для вступу абітурієнтів до ВНЗ”  
(назва згідно з наказом ректора)

Спеціальність – 121 “Інженерія програмного забезпечення”

## ПОЯСНЮВАЛЬНА ЗАПИСКА

БР.ПЗ - 31.00.00.000  
(позначення)

### Студент

ПЗ-43 \_\_\_\_\_ /Дмитро УСТИМЕНКО/  
(шифр групи) (підпис) (дата) (розшифровка підпису)

### Науковий керівник

к.т.н. \_\_\_\_\_ /Геннадій ПОРЄВ/  
(посада) (підпис) (дата) (розшифровка підпису)

### Консультант з питань нормконтролю

фахівець \_\_\_\_\_ /Тамара ЧАПОВСЬКА/  
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту

### Завідувач кафедри

д.т.н., проф. \_\_\_\_\_ /Олексій БИЧКОВ/  
(посада) (підпис) (дата) (розшифровка підпису)

Київ – 2021

**Київський національний університет імені Тараса Шевченка**

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

\_\_\_\_\_ ( Олексій БИЧКОВ )

(підпис)

(прізвище та ініціали)

**ЗАВДАННЯ**

**НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ  
БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

**Устименко Дмитру Олеговичу**

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної бакалаврської роботи “**Прототипування навчального ПЗ для вступу абітурієнтів до ВНЗ**”.

керівник проекту (роботи) Порєв Геннадій Володимирович., д.т.н., с.н.с. доцент  
затвержені наказом вищого навчального закладу від “11 ” листопада 2020 р. №

2. Строк подання студентом роботи \_\_\_\_\_

3. **Вихідні дані до проекту (роботи)** Теоретичні концепції та формальні моделі побудови та функціонування інформаційних систем та програмних модулів для освітньої системи та підготовки випускників для вступу до ВНЗ.

4. **Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Побудова функціональної моделі майбутнього програмного модуля

2. Розробка функціональних вимог до розроблюваної програми

3. Розробка програмного модуля для повноцінної роботи інформаційної системи для школярів для підготовки до вступу у ВНЗ

4. Оцінка роботи створеної програми

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

1. Функціональна схема ( рис.1.3, ст. 21)
2. Діаграма прецедентів ( рис. 2.2, ст. 28)
3. Блок-схема ( рис. 2.3, ст. 29).
6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основна частина	Доцент, д.т.н., с.н.с. Порєв Г.В.		

7. Дата видачі завдання 11.11.2020 р.

Керівник \_\_\_\_\_ (Геннадій ПОРЄВ)

Завдання прийняв до виконання \_\_\_\_\_ (Дмитро УСТИМЕНКО)

## КАЛЕНДАРНИЙ ПЛАН

НОМЕР І НАЗВА ЕТАПІВ БАКАЛАВРСЬКОЇ РОБОТИ	ТЕРМІН ВИКОНАННЯ ЕТАПІВ РОБОТИ	ПРИМІТКА
1. Підбір і вивчення літератури	25.11.2020	виконано
2. Аналіз концепцій та алгоритмів вирішення задачі	09.01.2021	виконано
3. Розробка функціональної моделі	20.01.2021	виконано
4. Побудова архітектури програмного модуля	03.02.2021	виконано
5. Моделювання даних	17.02.2021	виконано
6. Вибір та обґрунтування обраних технологій, програмного середовища та мови програмування	10.03.2021	виконано
7. Розробка програмного модуля	31.03.2021	виконано
8. Перевірка виконання функціональних вимог програмного модуля	08.04.2021	виконано
9. Оформлення записки	12.04.2021	виконано
10. Затвердження пояснювальної записки роботи завідувачем кафедри		виконано

Студент – бакалавр \_\_\_\_\_ (Дмитро УСТИМЕНКО)

(підпис) (розшифровка підпису)

Керівник роботи \_\_\_\_\_ (Геннадій ПОРЄВ)

(підпис) (розшифровка підпису)

## АНОТАЦІЯ

Випускна кваліфікаційна бакалаврська робота: 72 сторінки, 25 рисунків, 3 додатки та 37 джерел використаних в роботі;

**Тема:** Прототипування навчального ПЗ для вступу студентів до ВНЗ

**Об'єктом дослідження** виступає організація процесу підготовки абітурієнтів для вступу до ВНЗ.

**Метою** кваліфікаційної роботи є розробка програмного модуля для навчання та підготовки школярів до вступу у ВНЗ.

**Предметом дослідження** є оптимізація та автоматизація роботи інформаційної системи для абітурієнтів для підготовки до вступу у ВНЗ.

**Результат дослідження.** В даній роботі спроектовано та розроблено програмний модуль для підготовки абітурієнтів до вступу у ВНЗ. Створена можливість перегляду повної інформації по дисципліні, перегляду методичних розробок та проходження тестів.

**КЛЮЧОВІ СЛОВА:** ІНФОРМАЦІЙНО-ДОВІДКОВА СИСТЕМА, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОГРАМНИЙ МОДУЛЬ.

## АННОТАЦИЯ

Выпускная квалификационная бакалаврская работа: 72 страницы, 25 рисунков, 3 приложений и 37 источников использованных в работе;

**Тема:** прототипирование учебного ПО для поступления студентов в вузы

**Объектом исследования** выступает организация процесса подготовки абитуриентов для поступления в вузы.

**Целью квалификационной работы** является разработка программного модуля для обучения и подготовки школьников к поступлению в ВУЗ.

**Предметом исследования** является оптимизация и автоматизация работы информационной системы для абитуриентов для подготовки к поступлению в ВУЗ.

**Результат исследования.** В данной работе спроектирован и разработан программный модуль для подготовки абитуриентов к поступлению в ВУЗ. Создана возможность просмотра полной информации по дисциплине, просмотра методических разработок и прохождения тестов.

**КЛЮЧЕВЫЕ СЛОВА:** информационно-справочная система, ИНФОРМАЦИОННАЯ СИСТЕМА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, программных модулей.

## ANNOTATION

Final qualifying bachelor's thesis: 72 pages, 25 drawings, 3 appendices and 37 sources used in the work;

**Topic:** Prototyping of educational software for students entering university

**The object** of the study is the organization of the process of preparing applicants for admission to the university.

**The purpose** of the qualification work is to develop a software module for teaching and preparing students to enter the university.

**The subject** of the study is the optimization and automation of the information system for applicants to prepare for university entrance.

**The result of the study.** In this work, a software module for designing entrants to enter the university is designed and developed. There is an opportunity to view complete information on the discipline, view methodological developments and take tests.

**KEY WORDS:** INFORMATION AND REFERENCE SYSTEM, INFORMATION SYSTEM, SOFTWARE, SOFTWARE MODULE.

## ЗМІСТ

ВСТУП .....	10
<b>РОЗДІЛ 1</b> .....	<b>14</b>
<b>ОГЛЯД ІНФОРМАЦІЙНИХ ДЖЕРЕЛ І СУЧАСНИЙ СТАН ПРОБЛЕМИ</b>	<b>14</b>
1.1. Актуальність інформаційних технологій у навчанні .....	14
1.2. Доцільність прототипування навчального ПЗ для підготовки учнів для вступу у ВНЗ. ....	20
1.3. Функціональне моделювання даних спроектованої системи.....	25
1.4. Функціональні вимоги.....	26
Висновок до розділу 1 .....	28
<b>РОЗДІЛ 2</b> .....	<b>29</b>
<b>РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ</b> .....	<b>29</b>
2.1. Архітектура програмної системи .....	29
2.2. Моделювання даних .....	31
2.3. Розробка інтерфейсу програмного модуля.....	33
2.4. Вибір програмного середовища та мови програмування для розробки графічного інтерфейсу .....	38
2.5. Керівництво користувача .....	47
Висновок до розділу 2 .....	50
<b>РОЗДІЛ 3</b> .....	<b>51</b>
<b>ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>51</b>
3.1. План випробувань.....	51
3.2. Перевірка виконання функціональних вимог програмного модуля .	52
Висновки до розділу 3 .....	55

ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТОК А.....	62
ДОДАТОК Б.....	63
ДОДАТОК В.....	64

## ВСТУП

**Актуальність дослідження.** З кожним роком інформаційні технології все глибше проникають у сучасне суспільство. Передусім, до автоматизації процесів у багатьох галузях людської діяльності, а зокрема і діяльності навчальних закладів, спонукають величезні потоки інформації, операції з ними, наявність інноваційних технологій та стрімкий розвиток сучасного світу загалом. Автоматизація вже давно увійшла в наше життя. Вона супроводжує різні сфери професійної діяльності людини, дозволяючи прискорити, спростити і структурувати виконану роботу.

Інформація як головний фактор виробництва в формі сучасних технологій відкриває великі можливості якісного зростання на базі наступних інструментів і факторів:

- необмеженість комерційних майданчиків в Інтернеті, розвитку Інтернет-торгівлі, фінансових (фондових і валютних) бірж;
- мінімізація розміру організацій для успішної конкуренції на ринках, а також неупинного розвитку систем управління;
- активне використання інноваційних освітніх технологій і сучасних засобів в освітньому процесі;
- активізації наукових досліджень.

Постійний розвиток та неупинне вдосконалення сучасних інформаційно-комунікаційних технологій (ІКТ), їх активна реалізація значною мірою впливає на характер наукових досліджень, виробництва, освіти, соціальні взаємовідносини, культуру тощо. Це, передусім зумовлює як прямий вирішальний вплив на зміст освіти, пов'язаний з сталим рівнем науково-технічних досягнень, її удосконалення в умовах реального навчального процесу, так і опосередкований, що нерозривно пов'язаний із появою на ринку праці нових спеціальностей. Варто зазначити, що на сьогоднішній день, на тлі

глобальних змін в інформаційному суспільстві відбувається активний розвиток та інтенсивне використання ІКТ в усіх сферах життя, зокрема в освіті.

Сучасна система навчання за допомогою електронних та інформаційних технологій змінюють саму концептуальну модель освітньої активності, адже традиційне поняття навчального процесу досить важко поєднується із використанням ІКТ. Основними характерними ознаками, що відтворюють ці напрями виступають: мобільність, фокусування, доступність, універсальність, індивідуальна спрямованість викладацького складу та навчально-методичних матеріалів, результативне пристосування випускника до роботи у сучасних умовах із застосуванням новітніх інформаційних технологій.

Інформаційні технології надають можливість керувати академічними знаннями; формувати актуальні новітні системи управління якістю освіти; ефективно об'єднувати науку, освіту та інноваційні технології на основі створення освітніх кластерів. Інформаційні технології активно застосовуються для практичної реалізації навчальних програм, які полягають не тільки в інструментальних системах технологій ведення навчального процесу, але і в модернізації навчальних дисциплін та учбових планів. Саме інформаційні технології дають можливість створювати модернізовані навчально-методичні матеріали, та генерувати унікальні траєкторії навчання для школярів.

Неможливо не зазначити, що, враховуючи надзвичайний стан в усьому світі, особливо доцільним на сьогоднішній день інформаційні технології у системі освіти постають для абітурієнтів, які через карантинні обмеження втратили можливість для повноцінного відвідування навчальних закладів та комплексної підготовки до фахових іспитів до вступу у ВНЗ.

Актуальність теми обумовлена тим, що розвиток технологій автоматизації процесів, досягло такого рівня, який повинен був відбутися якісний перехід в новий стан. В даний час існує велика кількість різних підходів до вирішення даної проблеми, які базуються або на реляційному, або об'єктно-орієнтованому проектуванні.

**Актуальність теми** полягає в модернізації процесу проходження підготовчого курсу дисциплін для кращого засвоєння матеріалу та підготовки до вступу до ВНЗ, а також для наявності можливостей проходження курсу дисципліни, не маючи можливості бути в навчальному закладі.

Сучасний стан інформаційних технологій демонструє високий зріст наочних розробок. Під наочними розробками маються на увазі створення інформаційно довідкових систем. Створення можливостей перегляду навчальної літератури, методичних розробок, перегляду матеріалів і проходження тестів по засвоєному матеріалу.

**Метою** кваліфікаційної роботи є розробка програмного модуля для навчання та підготовки школярів до вступу у ВНЗ.

Мета дослідження передбачає постановку і вирішення наступних **завдань**:

- комплексний аналіз інформаційно-пошукових та інформаційно-довідкових систем для оптимізації навчального процесу, їх функцій і специфіку структури;
- аналіз закономірностей і перспектив використання моделей зберігання інформації, формалізації представлення даних при побудові інформаційно-довідкових систем;
- побудова функціональної моделі майбутнього програмного модуля;
- окреслення функціональних вимог до розроблюваної програми;
- розробка програмного модуля для повноцінної роботи інформаційної системи для школярів для підготовки до вступу у ВНЗ
- оцінка роботи створеної програми.

**Об'єктом дослідження** виступає організація процесу підготовки абітурієнтів для вступу до ВНЗ.

**Предметом дослідження** є оптимізація та автоматизація роботи інформаційної системи для абітурієнтів для підготовки до вступу у ВНЗ.

В даній роботі **спроектовано та розроблено** програмний модуль для підготовки абітурієнтів до вступу у ВНЗ.

**Методи дослідження.** У дипломній роботі використовувалися синтез, формалізація, аналітичний, порівняльний, графічний методи, а також методи інтелектуального аналізу даних.

**Наукова новизна отриманих результатів.** Досліджено можливості застосування інформаційних технологій для ефективної підготовки випускників до здачі фахового іспиту для підготовки до вступу у ВНЗ.

**Практичне значення одержаних результатів.** Розроблений програмний модуль дозволяє ефективно готуватись та перевіряти отримані знання та навички у процесі підготовки школяра для вступу до ВНЗ. Також розроблений програмний модуль є досить хорошим фундаментом для масштабніших проектних рішень.

# РОЗДІЛ 1

## ОГЛЯД ІНФОРМАЦІЙНИХ ДЖЕРЕЛ І СУЧАНИЙ СТАН ПРОБЛЕМИ

### 1.1. Актуальність інформаційних технологій у навчанні

В умовах сьогодення електронне навчання (*e-learning*) допомагає вирішити ряд завдань, що нерозривно пов'язані із пристосуванням освітньої системи в цілому та навчальних закладів до змін, що відбулися в умовах карантину, такі як дистанційне навчання, індивідуальні навчальні плани тощо. У свою чергу автоматизована система освіти включає в себе уже наявні традиційні підходи до навчання і з застосуванням електронних технологій, але не обмежується ними.

Концепція автоматизації процесу отримання освіти – гнучкий процес, що передбачає наявність досить значного об'єму джерел, різноманітність мультимедіа, можливість легко та швидко налаштуватися під рівень знань та потреби здобувача освіти. Разом з тим змінюється роль викладача, який, у разі використання під час навчального процесу інформаційних-технологій, повинен змінювати підхід до системи контролю[4]. Тенденції переходу інформаційних технологій окреслюють нові вимоги до вчителів: педагоги повинні мати великий багаж знань не тільки у своїй професійній сфері, а й мати широкий світогляд, вміти швидко налаштуватись до роботи у суміжних галузях та реалізовувати різноманітні технології для роботи із учнями. Такі інноваційні підходи вимагають від викладачів створювати сприятливі умови для придбання молоддю власних навичок та досвіду, а не тільки надавати готові знання.

Смарт-освіта – це сучасне освітнє середовище; об'єднання навичок та зусиль фахівців, викладачів, вчителів, студентів та школярів для ефективного застосування всесвітніх знань та переходу до активного контенту. Провівши аналіз на основі досліджень в області автоматизації освіти, варто зазначити, що комплексне дослідження багатьох її елементів фактично не проводилося. На

сьогоднішній день уже досить популярним стає проведення навчальних занять з активним використанням мультимедійних презентацій, зроблених за допомогою усіх відомих програмних пакетів *Microsoft Power Point* або *Macromedia Flash*. Однак, разом зі звичними презентаційними технологіями в освітню сферу проникають нові інформаційні технології, які дають змогу уникнути презентації у вигляді слайд-шоу. Абсолютно новим для сприйняття є те, що на інтерактивних дошках можна демонструвати навчальний матеріал, писати ключові моменти та робити письмові нотатки над зображеннями спеціальним маркером на екрані. При цьому все, що викладено та додатково зазначене на такій дошці транслюється учням, зберігається на магнітних носіях із можливістю роздруковувати матеріал або надіслати його електронною поштою [5].

У глобальній мережі Інтернет можна знайти досить багато різноманітних веб-ресурсів для автоматизованого навчання. Я вважаю доцільним зазначити найефективніші із них:

- *Khan Academy*: некомерційний освітній ресурс, що пропонує у відкритому доступі велику кількість матеріалів із різних навчальних дисциплін, тестові завдання із можливістю моментальної перевірки, а також вікторини для проведення уроків.

- *Quora*: соціальний сервіс, що широко використовується для цілого ряду цілей, і його можна сміливо віднести до одних із найкращих технологічних інструментів для педагогів. Бездоганне освітнє середовище, щоб підлітки могли навчитися шляхом обговорення освітніх тем, або, за необхідності, знайти відповіді на питання шляхом розміщення запитань у Quora. Також сервіс надає можливість пошуку питання, що потребує вирішення шляхом звернення до архіву, де необхідні відповіді об'єднані спільною тематикою.

- *Capsules*: ресурс, що надає змогу користувачеві збирати фотографії, відеозаписи, публікації в блозі та важливі документи в одному місці. Викладачі можуть використовувати його для навчання в класі та ведення онлайн проєктів.

- *Google Docs*: безкоштовний хмарний застосунок, що є досить ефективним для викладачів. Педагоги можуть створювати форми зворотного

зв'язку для виконаних учнями проектів. Разом з тим, створення документів, таблиць, презентацій та їх подальший обмін між викладачем та учнями в аудиторії відбувається швидше за допомогою застосунку “Google Docs”.

– *Evernote*: найкращий сервіс для створення нотаток у режимі онлайн. Цей інструмент дозволяє користувачам бути добре організованими; робити, зберігати та синхронізувати фотографії, ідеї, нотатки, коментарі чи щось інше та отримувати безперервний доступ до них будь-де. Цей інструмент доцільно використовувати для планування уроків.

– *Socrative*: Інтернет-інструмент для викладачів. Ця система заохочує дітей до різних вправ і ігор, допомагає педагогам створювати, шукати та поширювати тести, та, безпосередньо, допомагає із проведенням тестувань. Цей інструмент, який доступний на комп'ютерах, у планшетах та на мобільних пристроях, має досить простий та зручний інтерфейс, дозволяє викладачам стежити за прогресом учнів та їх оцінками.

– *YouTube*: досить популярний відеохостинг, що містить у своїх архівах безліч важливих навчальних матеріалів, які згодом доцільно буде використовувати в аудиторії для оптимізації навчального процесу. *YouTube* також має спеціальний розділ для освіти. Завдяки деяким системним обмеженням, школярі мають змогу максимально ефективно використовувати *YouTube*, не відволікаючись на сторонні процеси.

– *Dropbox*: досить сучасний робочий простір, є одним із найкращих технологічних ресурсів для педагогів для оптимізації навчального процесу. *Dropbox* можна використовувати для зберігання даних, отримання доступу до них та обміну будь-якою інформацією. Також у застосунку є можливість надати для обмеженого кола користувачів (наприклад для однієї групи учнів). *Dropbox* є безкоштовним файлообмінником та досить легким у використанні.

– *SlideShare*: сервіс, що дозволяє досить легко завантажувати презентації, відеопроєкти та будь-яких інші документи. *SlideShare* дозволяє користувачеві завантажити документ приватно, або зробити доступним для усіх бажаючих.

Цей перелік не є вичерпним, наведено , на мою думку, найважливіші застосунки для оптимізації навчального процесу.

Передумовами до розробки концепції автоматизації освітнього процесу є[6-7]:

- технологічні фактори, які забезпечують нові технології та засоби для навчального процесу в сучасному інформаційно-телекомунікаційному середовищі;
- соціальні умови, включають вимоги сучасного суспільства до якості освітніх послуг та їх удосконалення;
- економічні чинники, полягають в тому, що освіта завжди робила значний внесок в розвиток макроекономіки.

Ключовою ціллю смарт-навчання виступає створення належних умов для отримання максимальної ефективності у навчальному процесі. Практична реалізація смарт-навчання вимагає всебічного підходу, включаючи організаційний підхід, педагогічний та технологічний. Я вважаю доцільним буде зазначити ключові переваги запровадження до освітнього процесу інформаційних технологій:

- особистісно-орієнтований підхід у навчанні, запровадження персональних освітніх методик, що працюють та показали реальний результат;
- оперативний зворотний зв'язок з педагогами й іншими учнями;
- необмежений доступ до різноманітних джерел навчальної інформації;
- збільшення об'ємів самостійної індивідуальної та групової роботи, активізація процесу мотивування та заохочення школярів;
- запровадження новітніх освітніх технологій, що побудовані на широких можливостях навчальних автоматизованих інформаційних систем;
- реалізація змішаної форми навчального процесу;

– формування комунікативних компетентностей засобами віртуального освітнього середовища, що сприяють професійній соціалізації учнів.

Разом з тим, головним недоліком інформаційних технологій можна назвати відсутність безпосередньої комунікації між школярем та викладачем, а й іноді заформалізованість протоколів спілкування в мережі. Однак, співставивши такі незначні недоліки та вагомні переваги, а також взявши до уваги неупинний інформаційний розвиток технологій, інформаційні-технології у навчальному процесі майбутньому ставатимуть тільки ефективнішими та дієвішими[8].

Необхідність накопичення великих об'ємів професійно-цінних інформації і оперування ними – одна із проблем, з якою зіштовхнулися вчителі та школярі.

Активне розповсюдження інформаційних ресурсів, її подальший збір обробка та трансформація всередині сучасного інформатизованого суспільства відбувається завдяки спеціальним ресурсам. У визначений період ці інформаційні дані збираються в одному місці, за заздалегідь окресленими критеріями структуруються, об'єднуються у зручні спеціальні бази з метою подальшого використання. Інформаційна система виступає ефективним інструментом, за допомогою якого відбувається обробка та трансформація отриманих відомостей. Ключова функція інформаційної системи виступає у забезпеченні актуальними інформаційними даними суспільство. Інакше кажучи інформаційна система задовольняє потребу в отриманні відомостей в межах окресленої області предмета. Кінцевим продуктом активної діяльності інформаційної системи виступають масиви даних, структуровані бази, нормативні документи.

Інформаційна система виступає системою, яка реалізує або в якій відбуваються інформаційні процеси такі як пошук, накопичення, збереження, трансформація та обробка інформаційних даних. Обробка інформації передусім залежить від об'єму та змісту вхідних інформаційних даних, однак в процесі

самої обробки інформаційні потоки не осмислюються, а лише трансформуються відповідно до алгоритмів, розроблених раніше.

Далі окреслимо активні процеси, які відбуваються в інформаційній системі:

- введення інформаційних потоків, які одержані із джерел інформації;
- обробка та трансформація інформаційних даних;
- збереження вхідних та опрацьованих інформаційних;
- виведення призначеної для користувача інформації;
- відправка / отримання інформації мережею.

Розробка інформаційної системи передбачає вирішення наступних задач:

- заповнення інформаційної системи даними специфічної вирізняльної предметної області;
- розробка інтерфейсу активного користувача (переважно графічного) для одержання потрібних інформаційних ресурсів [2].

До автоматизованої системи інформаційні дані поступають від джерела інформації. Такі відомості відправляють на подальше зберігання чи трансформацію у системі, а вже згодом передають активному споживачеві. Цей процес ми можемо простежити на рисунку 1.1.



Рис. 1.1. Процес отримання інформації

## **1.2. Доцільність прототипування навчального ПЗ для підготовки учнів для вступу у ВНЗ**

На тлі збільшення попиту на конкурентоспроможних кваліфікованих кадрів на ринку праці, можемо спостерігати активні зміни освітньої системи України. Це стало першочерговою причиною поступового підвищення вимог до рівня підготовки учнів старших класів та, безпосередньо, абітурієнтів, з метою формування у майбутньому їх вдалої професійної діяльності.

Однак можна відмітити, що учні, особливо випускних класів, зазнають помітних проблем у навчанні. Причинно-наслідковим діянням такого процесу, логічно, виступає низький рівень обізнаності та засвоєння навчального матеріалу, а також проблеми зі вступом до компетентного вищого навчального закладу.

Далі вважаю за доцільне виокремимо ключові переваги використання інформаційних технологій у навчальному процесі, що є досить актуальним для учнів старших класів та вступників:

- організація пізнавальної діяльності шляхом моделювання;
- використання мультимедійних матеріалів з метою імітації типових професійних ситуацій;
- активне використання отриманих під час навчального процесу знань та навичок у наукових дослідженнях;
- ефективне тренування умінь, знань та навичок;
- автоматизований контроль результатів навчання;
- розвиток у учнів творчого мислення;
- візуальне та звукове подання матеріалу у рамках навчальних програмах сприяє кращому засвоєнню пройденого навчального лекційного матеріалу.

Інформаційно-комунікаційні технології та їх активне використання у навчальному процесі надають майже необмежені можливості публікування, перегляду, обробки та доставки інформаційних даних будь-якого змісту та об'єму на необмежені відстані. В цих умовах питання самоосвіти абітурієнтів постає чи не на перший план. Під терміном “самоосвіта” можна тлумачити персональну пізнавальну діяльність школяра з метою досягнення певних особистісно значущих освітніх цілей. Самостійна робота має на увазі різноманітні види навчальної діяльності школярів, що здійснюються безпосередньо під час заняття, вдома дистанційно, під керівництвом вчителя, або і взагалі без його безпосередньої участі.

Тобто, можна відзначити що, ефективність навчального процесу безпосередньо та якість підготовки до вступних іспитів зокрема, у переважній більшості залежить від якісної організації процесу самостійного навчання, належної якості лекційних теоретичних чи мультимедійних матеріалів, що використовуються під час самостійного навчального процесу. Варто відзначити, що самостійна позааудиторна робота вступників сприяє:

- розширенню, закріпленню та поглибленню знань, отриманих у процесі навчання в навчальній аудиторії;
- активному отриманню нових навичок та знань;
- розвитку творчого підходу до вирішення проблемних питань, що може виступити досить ефективним рішенням на фаховому іспиті у стресовій ситуації;
- прояву індивідуальності учня;
- формуванню практичних вмінь та навичок у вирішенні ситуаційних задач.

До самостійної навчальної підготовчої діяльності можна віднести будь-яку діяльність учнів, що реалізується школярем в позааудиторний час в межах навчального процесу, яка в першу чергу сприяє їхньому особистісному розвитку, поглибленню та постійному збільшенню фахових знань. Позааудиторна навчальна діяльність школярів із застосуванням інформаційних технологій

сприяє не тільки якіснішій підготовці випускників до успішного вступу до ВНЗ, а й формуванню інформаційної культури школяра.

Для реалізації автоматизованого інформаційного навчального підготовчого процесу необхідне відповідне комунікаційне забезпечення пристроїв ( безперервний доступ до мережі Інтернет) та технічні засоби. Застосування новітніх технологій навчання спонукало створити програмний модуль для належної самостійної підготовки для вступу до ВНЗ, що вцілому максимізує ефективність навчального процесу.

Концепція автоматизованої освіти у процесі підготовки випускника до вступу до ВНЗ передбачає гнучкість, яка передбачає наявність досить значного об'єму джерел, різноманітність мультимедійних засобів, можливість легко та швидко підлаштуватись під потреби та рівень знань слухача. Така парадигма передбачає постійний обмін ідеями та досвідом, персоніфікацію курсу залежно від компетенцій суб'єктів навчального процесу та його завдань, значну економію часу та ресурсів на доопрацювання вже наявного навчального контенту замість створення його з нуля. Крім цього, смарт-освіта розуміється простою та зручною у застосування, легко керованою, як внутрішньо кожним навчальним закладом через універсальність навчального процесу, так і ззовні, тобто безперервно доповнюється зовнішніми різноманітними джерелами.

Дотримуючись ідеї смарт-освіти, сучасні навчальні підготовчі курси повинні стати якіснішими: окрім забезпечення легкого засвоєння навичок, знань та умінь вони мають також заохочувати до вивчення. Мотивувати сучасного підлітка, що має необмежений доступ до мережі Інтернет та електронних носіїв, звичайними текстовими методичними посібниками вже практично неможливо. Постає необхідність у створенні такого сценарію, який би спонукав та заохочував до наукової та творчої діяльності.

Найпопулярнішими технологіями смарт-освіти, що є досить актуальними для вступника у процесі його підготовки до вступу до ВНЗ можна зазначити:

- дистанційне навчання;
- *BYOD*;

- *Meetoo*;
- *m-learning*;
- хмарні технології;
- СМАРТ-дошки;
- програмні та веб-ресурси.

Варто відзначити, що створення програмної платформи чи веб-ресурсу за допомогою інформаційних технологій з певним набором функціональностей має ряд переваг, що зазначені нижче:

- Гнучкий графік навчання. Школяр має змогу самостійно обирати та коректувати режим вивчення навчальної програми та знаходити комфортний темп, адже його ніхто не буде квапити. Такий варіант є досить зручним для багатьох абітурієнтів.

- Здобувати освіту можна в абсолютно будь-якому комфортному місці: вдома в особистому кабінеті, в громадському транспорті, якщо передбачається тривала поїдка та навіть на свіжому повітрі в парку або лісі. Здатність змінити обстановку в залежності від своїх потреб робить навчальний процес менш обтяжливим та нудним, та й матеріал засвоюється краще в комфортних умовах.

- Простота освоєння інформації. Учні рекомендують різноманітні варіації подачі матеріалу, і є можливість обрати найзручніший. Крім того, якщо інформація не досить зрозуміла, то можна із легкістю переглянути матеріал повторно, причому стільки разів, скільки потрібно для повного розуміння та засвоєння.

- Можливість легко виходити із викладачем на зв'язок у будь-який зручний для користувача час. Для того щоб надіслати на перевірку виконане завдання або уточнити якийсь момент доцільно використовуються різні месенджери, електронну пошту, соціальні мережі, в загальному, всі сучасні методи дистанційних комунікацій, які будуть зручними для обох користувачів.

– Зникає необхідність самостійного підбору необхідних матеріалів у методичній літературі, навчальних посібниках чи друкованих працях вчених тощо. Значний об'єм інформації надається програмою, а решта знаходиться у відкритому доступі у мережі Інтернеті.

– Для вступника передбачається можливість самостійно перевірити свої знання та відслідковувати динаміку, за умови якщо програма електронного навчання передбачає рейтингову систему оцінювання.

Можна впевнено стверджувати, що головним завданням простору для автоматизованої освіти виступає формування положень для реалізації інтелектуального та творчого потенціалу добувачів освіти, оскільки інформаційні технології – це інтегровані засоби для навчання, що включають різноманітні елементи усіх видів інформаційних освітніх технологій, що сприяють організації освітнього процесу для учнів із різними видами діяльності безпосередньо під час заняття та поза його межами навчального процесу[12]. На сьогоднішній день до інформаційних технологій можна віднести такі засоби навчання, як: інтерактивна дошка, програмне забезпечення, персональний комп'ютер та програма контролю вмінь та навичок .

Отже, в умовах сьогодення однією із провідних проблем в Україні постає створення ефективної конкурентоспроможної програми освіти та реалізації підготовчої платформи для ефективної підготовки вступника до вищого навчального закладу. Використання інформаційних технологій – кандидат, що посідає першість у вирішенні цього завдання. Електронне навчання виступає передумовою, що надає можливість учасникам навчального процесу постійно взаємодіяти між собою, обмінюватись досвідом, навичками та професійними здобутками в будь-якій точці світу в режимі реального часу. Усе це неабияк сприяє активізації пізнавальної діяльності учня.

Отже, новітні навчальні інформаційні платформи надають обширний перелік можливостей для професійної самореалізації учня, а також для дистанційного навчання здобувачів освіти[13-14]. Разом з тим, варто зазначити, що у системі освіти досить часто виникають розбіжності між можливостями



## 1.4. Функціональні вимоги

Вирішено розробити інформаційну систему, в якій користувач матиме можливість:

- перегляду навчальної методичної літератури із різних дисциплін для належної підготовки до вступу у ВНЗ;
- виконання практичних завдань після опрацювання наданого лекційного матеріалу;
- проводити тестування на якість проходження курсу дисципліни і визначення рівня засвоєння матеріалу щодо набраних балів.

Майбутній програмний модуль повинен мати простий і зручний графічний інтерфейс користувача. Для розробки програмного модуля необхідно вирішити ряд приватних взаємопов'язаних завдань:

- вибір інструментарію і середовища розробки;
- підготовка до роботи і проектування модуля;
- розробка програмного модуля;
- оцінка ефективності розробленої системи.

Для розробки програмного забезпечення мною було обрано та використано середовище розробки *Visual Studio 2019*, яке знаходиться у вільному доступі.

Для виконання роботи була використана операційна система *Windows 10*, яка є чи не найпоширенішою операційною системою сімейства *Microsoft*.

Ключові причини вибору *Windows 10*:

- стійкість роботи системи;
- зручний користувацький інтерфейс;
- режим мультизадачності роботи;
- популярність використання;

Також необхідно відзначити, що у разі збою в роботі системи можливо повне відновлення програм, документів і інших даних, система є багатофункціональною і при встановленні має стандартний пакет драйверів.

Програмний модуль було розроблено на ПК із такими технічними характеристиками:

- ОС: *Windows 10 Pro*;
- процесор: *Intel I Core I i3-7020U CPU @ 2.30GHz 2.30 GHz*;
- оперативна пам'ять: 8,00ГБ.

Розроблений програмний модуль буде містити в собі наступні класи:

- клас *Database.cs*, для роботи з базою даних;
- клас *Pages.cs*, клас для роботи з робочими формами додатку;
- клас *User.cs* для роботи з користувачами;
- графічні ресурси, які постійно закріплені проекту, які необхідні для графічного інтерфейсу.

Надійне, однак, разом з тим і досить ефективне функціонування майбутньої програми має бути забезпечено виконанням наступних заходів:

- попередження помилок та збоїв шляхом структурного програмування;
- забезпечення постійного живлення техніки;
- обробка виняткових ситуацій та моментальна локалізація похибок, несправностей та збоїв;
- використання у роботі тільки ліцензійного програмного забезпечення;
- періодичне проведення діагностики на рівень захищеності системи вцілому та інформаційних ресурсів зокрема;

Також можна сформулювати такі вимоги до продуктивності роботи програмного модуля:

- час авторизації в систему не повинно перевищувати 10 секунд;
- необмежена кількість користувачів;
- необмежений час зберігання інформаційних даних в базі.

## Висновок до розділу 1

Проведений аналіз показав, що технології інтерактивності і мультимедійності використовуються в інформаційних підсистемах освітнього і наукового напрямку, мають актуальність і практичне застосування в системній організації інформаційних ресурсів, та для навчальних програм для підготовки учнів для вступу до ВНЗ зокрема.

При проведенні аналізу сучасних підходів до створення навчальних програм щодо застосування засобів обчислювальної техніки були виявлені проблеми, які виникають при створенні інформаційних систем; сформульовані вимоги до створення майбутнього програмного модуля, призначеного для підготовки студентів для вступу до ВНЗ. Виконано функціональне моделювання програмного модуля.

Програма призначена для полегшення ефективної роботи та підготовки до фахових іспитів для вступу до вищого навчального закладу майбутніх вступників та абітурієнтів. Також в першому розділі визначені функціональні обов'язки майбутнього спроектованого програмного модуля.

## РОЗДІЛ 2

### РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

#### 2.1. Архітектура програмної системи

Архітектура програмного продукту (*software architecture*) – специфічна структура програмного модуля, яка включає в себе першочергово програмні структурні елементи, зовні видимі характерні властивості згаданих компонентів, а також взаємовідносини між ними. Згаданий процес також стосується процесу документування архітектури програмного забезпечення.

Прикладне програмне забезпечення, призначене для вирішення завдань користувача. Виконання роботи розподілене на окремі блоки, які можуть виконуватися як послідовно, так і паралельно (в окремих потоках програмного продукту). Кожен блок виконує тільки свою специфічну роботу, наприклад, перевіряє дані, введені користувачем.

В архітектурі додатки можна виділити два головних компонента:

- блок графічного інтерфейсу користувача, який представляє зовнішній вигляд фінального програмного результату і виконує взаємодію з користувачем програмного продукту;
- модуль вирішенні завдання, який отримує і аналізує вхідні дані, складає і вирішує завдання, оформляє результат виконання і передає його до графічного інтерфейсу.

Графічний інтерфейс користувача дозволяє активно взаємодіяти із програмою способом, який зручний та комфортний безпосередньо для користувача . Він складається з:

- блоку взаємодії з вхідними даними;
- блоку взаємодії з вихідними даними.

Модуль вирішенні завдання виконує саму роботу програмного продукту. Створюється модуль з таких компонентів:

– блок обробки результату моделювання групує результат виконання блоку вирішенні завдання в набір масивів даних і передає в блок графічного інтерфейсу користувача;

– блок вирішенні задачі розрахунку нормованих показників і показника сумарного питомої нормованого потенціалу [1].

Оскільки програмний продукт написаний на мові, який реалізує принципи об'єктно-орієнтованого програмування, кожен з компонентів програми можна представити у вигляді об'єкта, яких є екземпляром певного класу. Тобто робота і поведінка кожного компонента архітектури описується класом. Це дозволяє створити ефективну і гнучку взаємодію об'єктів.

Для поліпшення швидкодії програмного продукту все трудомісткі операції виконуються в окремих паралельних потоках, що дозволяє мінімізувати час виконання операцій на ПК із багатоядерними процесорами.

У розробці програмного модулю для підготовки школярів до вступу у ВНЗ була обрана клієнт-серверна архітектура. Клієнт-серверна архітектура – це такий спосіб організації структури додатків, в якій визначені для виконання системи завдання, умовним чином диференціюються між двома підкласами: клієнтом і сервером.

До сервера звертаються безліч клієнтів і часто він зберігає дані, тому він виступає “центральною” частиною. З цього випливає, що сервер може існувати без клієнтів, а от клієнти без сервера не можуть. Клієнт – програма, примірників якої зазвичай “більше”, ніж серверів.

Система диференціюється на окремі частини, які мають можливість виконуватися в будь-яких вузлах мережі. Кінцевий користувач, або ж безпосередньо програма активно взаємодіють із клієнтською долею системи, яка в забезпечує міжмережвий інтерфейс. Якщо виникає необхідність, то клієнтська частка системи звертається до серверної частини по мережі.

Архітектура майбутнього програмного модуля зображена на рис. 2.1.

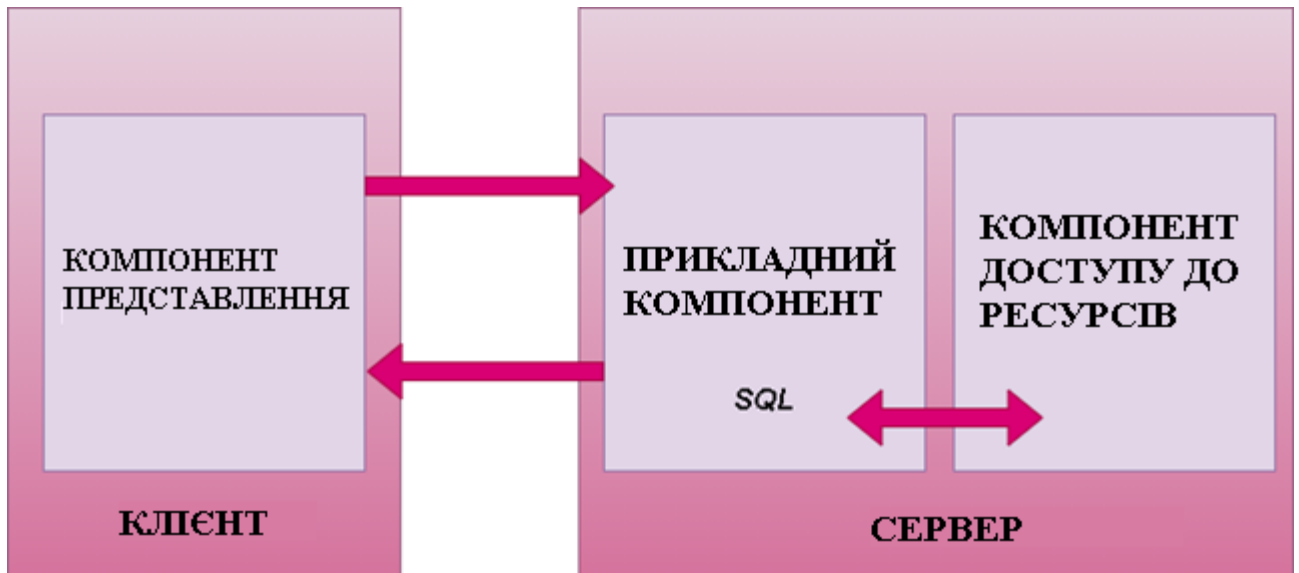


Рис. 2.1. Архітектура програмного модуля

Однією із переваг такої моделі взаємодії є те, що розділений вихідний код клієнтського і серверного додатків. До помітних переваг згаданої архітектури можна віднести зменшені вимоги до пристроїв користувачів, а також використовувана архітектура досить гнучка і надає змогу адміністратору локальної мережі зробити її більш захищеною [10].

Недоліками такої організації виступає той факт, що фінансові затрати на серверне обладнання значною мірою більші, ніж клієнтське. Сервер також повинен обслуговувати лише фахівець, спеціально підготовлений. Також варто відзначити, що у випадку збою сервера у локальної мережі втрачається можливість клієнта працювати у нормальних, звичних для нього умовах.

Однак, ці недоліки не є критичними, саме тому обраний такий варіант

## 2.2. Моделювання даних

В процесі виконання роботи доцільно буде розробити діаграму прецедентів. Цей вид діаграм слугити для того, щоб відобразити відносини між акторами і прецедентами, визначити межі системи, описують функціональні вимоги системи (рис. 2.2).

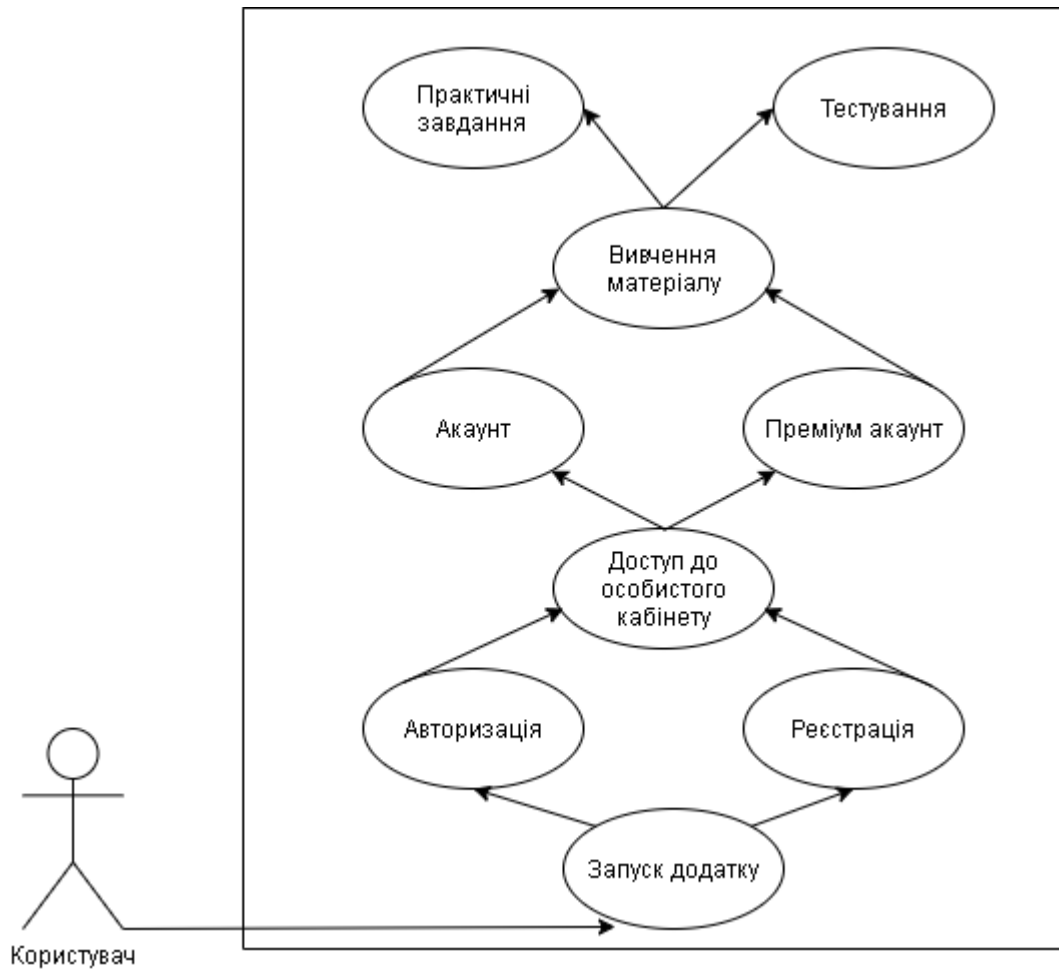


Рис. 2.2. Діаграма прецедентів

На діаграмі прецедентів зображено користувача, у якого є варіант вибору функціоналу. За допомогою цієї діаграми описуються можливості системи, користувачі системи, взаємодія між користувачами і системою. Діаграму прецедентів зручно використовувати при визначенні бачення майбутньої системи для того, щоб зрозуміти хто буде користувачами системи, як ці користувачі будуть використовувати систему.

Для комфортного користування програмним модулем приведена також блок-схему (рис. 2.3).

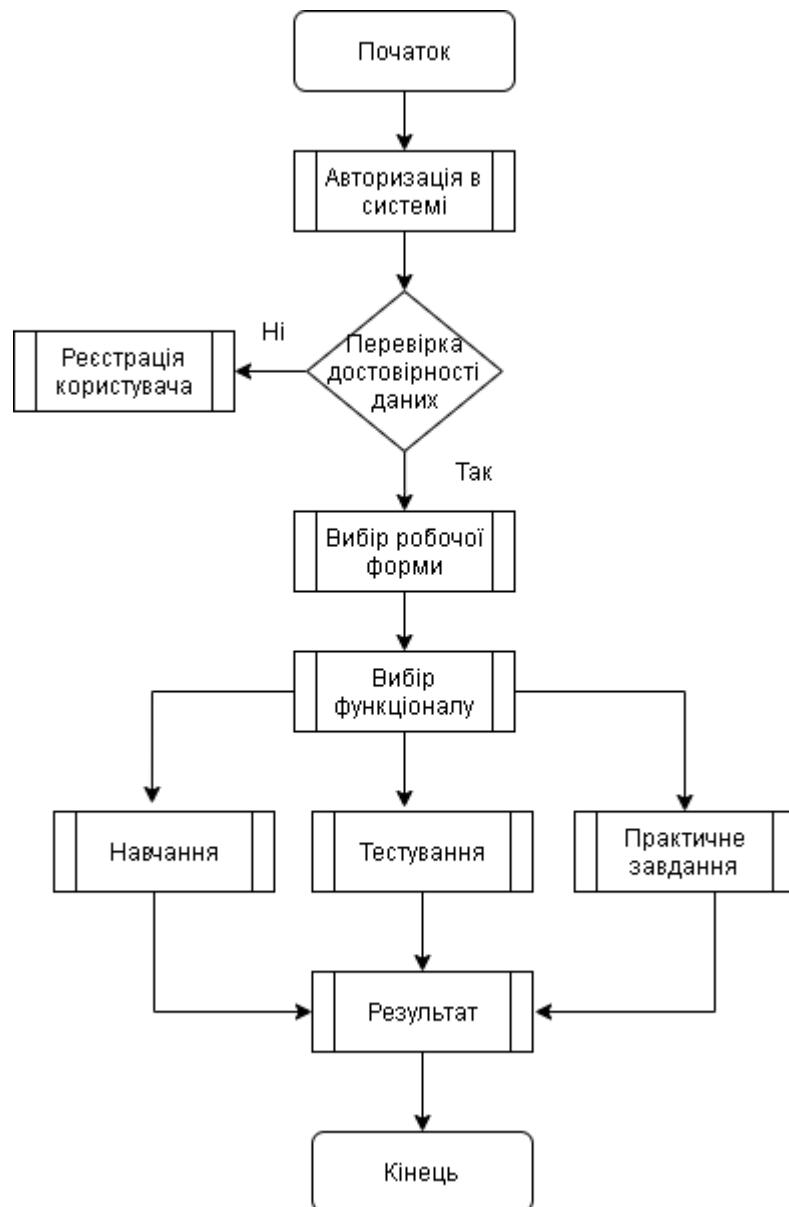


Рис. 2.3. Блок-схема

### 2.3. Розробка інтерфейсу програмного модуля

Для того щоб інтерфейс програмного модуля для активного користувача був зручним, він має проектуватися на базі продуманої реалізації набору типових операцій над даними. У випадку відсутності такої бази, буде спостерігатися негативний відбиток на якості безпосередньо фінішного програмного продукту, час розробки такого продукту помітно збільшиться.

Вимоги до інтерфейсу, призначеного для користувача:

- зменшення часу виконання функцій пошуку та зчитування інформації;
- зменшення термінів вибору команди та навігації;
- збільшення загальної продуктивності користувача, яка полягає в обсязі оброблених вхідних даних за конкретний проміжок часу;
- максимізація тривалості стійкої роботи користувача.

Стильова гнучкість:

- Можливість використовувати з одним і тим самим додатком різні інтерфейси, практично реалізуються із використанням таблиці стилів, разом з тим враховуючи можливість у виборі активним користувачем власних установок інтерфейсу .

Збільшення функціональності:

- Передбачення можливості розвитку додатку без руйнування існуючого інтерфейсу – маєтсья на увазі залишення у рамках уже існуючого спроектованого інтерфейсу.

Масштабованість:

- Можливість налаштування та поступового розширення інтерфейсу, а також і самого додатку у випадку зростання числа активних користувачів, об'єму та змісту даних.

Адаптивність до дій користувача:

- Програмний модуль має передбачати можливість введення даних і команд виключно різними варіантами та способами та багатоваріантність доступу до прикладних функцій. Також варто відзначити, що програмне забезпечення має враховувати варіант переходу та повернення від форми до форми та вміти коректно обробляти згадані ситуативні варіації.

Переносимість:

– При переході на іншу апаратну (програмну) платформу, повинен здійснюється автоматично перенесення і призначеного для користувача інтерфейсу, і кінцевого додатки.

Методи оцінки користувальницького інтерфейсу:

– Для оцінки необхідного рівня зручності інтерфейсу використовуються спеціальні експертні анкети, опитувальники, формуляри, *check*-листи.

В якості методів використовують:

– спостереження за користувачами до використання програмного інтерфейсу, в процесі навчання і роботи;

– постановка і протоколювання виконання тестових завдань.

Цілі і критерії оцінки користувальницького інтерфейсу.

– ключове у програмному модулі – реалізація такого інтерфейсу, що призначений для користувача, який зможе виконати поставлене завдання ефективно та продуктивно, а також забезпечить задоволеність користувача від роботи з програмою.

– Ефективність роботи полягає у забезпеченні точності, функціональної повноти і завершеності при вирішенні та практичній реалізації виробничих завдань на автоматизованому робочому місці активного користувача. Ефективність роботи відображає об'єм затрачених ресурсів у процесі вирішення поставленого завдання.

– Розробка інтерфейсу, який призначений для користувача має бути спрямований на показники ефективності автоматизованої інформаційної системи, які доцільно вимірювати як кількісно, так і об'єктивно:

– Продуктивність праці. Визначається середньою кількістю вирішених завдань.

– Точність роботи – має на увазі кількість помилок. Показник точності враховує відсоткове відношення помилок та похибок, які зробив користувач у

процесі роботи: кількісні показники помилок набору, кількість неправильних та некоректних звернень до даних, запитів тощо.

– Функціональна повнота. Окреслюється тим, в якому розмірі розроблений програмний модуль користувачем, відповідає заявленим до нього вимогам; окреслює ступінь застосування первинних і трансформованих даних, переліку необхідних процедур опрацювання звітів, кількість пропущених технологічних операцій або етапів при виконанні поставленого користувачеві завдання.

– Завершеність роботи. Виявляє відповідний рівень виконання виробничого завдання посереднім користувачем за окреслений термін або період, відсоткову частку необроблених заявок, кількість продукції у відсотковому співвідношенні, що знаходиться на проміжній стадії готовності, а також число користувачів, які виконали завдання у фіксований термін.

– Простота освоєння. Визначається часом освоєння інтерфейсу програмного забезпечення.

Вимоги до зручності і комфортності інтерфейсу зростають зі збільшенням складності робіт і відповідальності користувача за кінцевий результат. Значний рівень ефективності та комфортності від роботи програмного продукту досягається у випадку:

– наявності прозорої навігації для користувача і цільової орієнтації в програмному модулі. Досить важливим моментом є визначення питання покроковості виконання;

– чіткості та ясності розуміння користувачем текстів, символів та значення кнопок чи іконок. У програмному модулі повинні бути використані такі символи та графічні образи, які користувач знає або повинен знати за характером його роботи або займаної посади;

– наявності допоміжних методів та способів підтримки користувача, враховуючи і процес прийняття рішення в невизначеній ситуації.

Зручний та простий інтерфейс допомагає користувачеві впоратися з втомою і напругою при роботі в умовах високої відповідальності за результат.

Розробка користувацького інтерфейсу ведеться паралельно розробці програмного продукту в цілому і в основному передує його впровадженню. Процес розробки програмного інтерфейсу умовно можна диференціювати на такі етапи:

- аналіз діяльності потенційних користувачів;
- формулювання вимог до роботи користувача;
- побудова користувацької моделі даних.

Сформуємо інтерфейс форми авторизації та головної форми для відображення даних в наступному вигляді (див. Рис. 2.4-2.5).

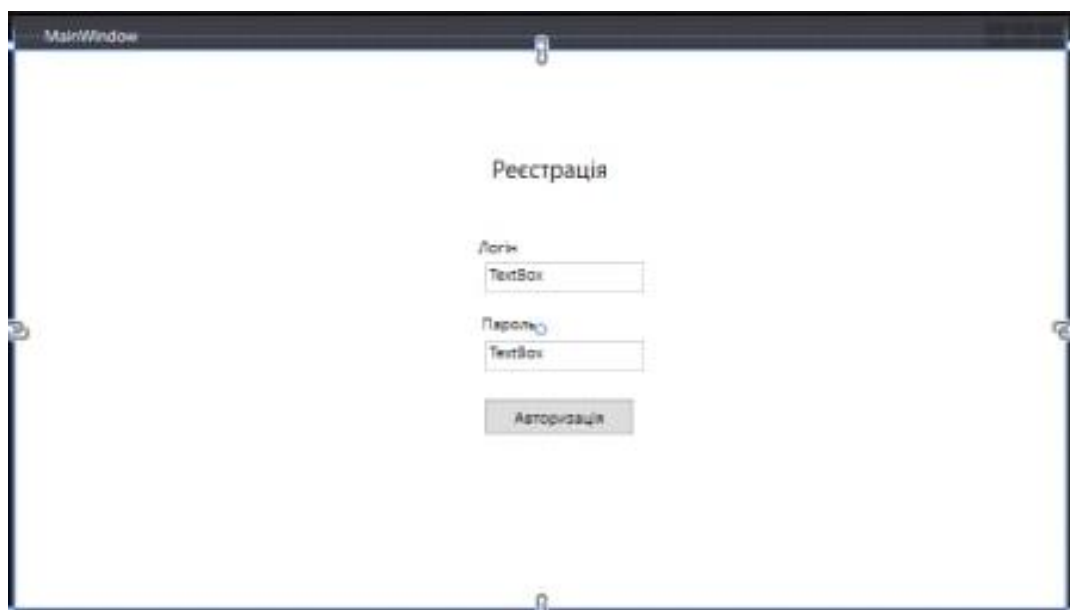


Рис. 2.4. Проект інтерфейсу форми авторизації

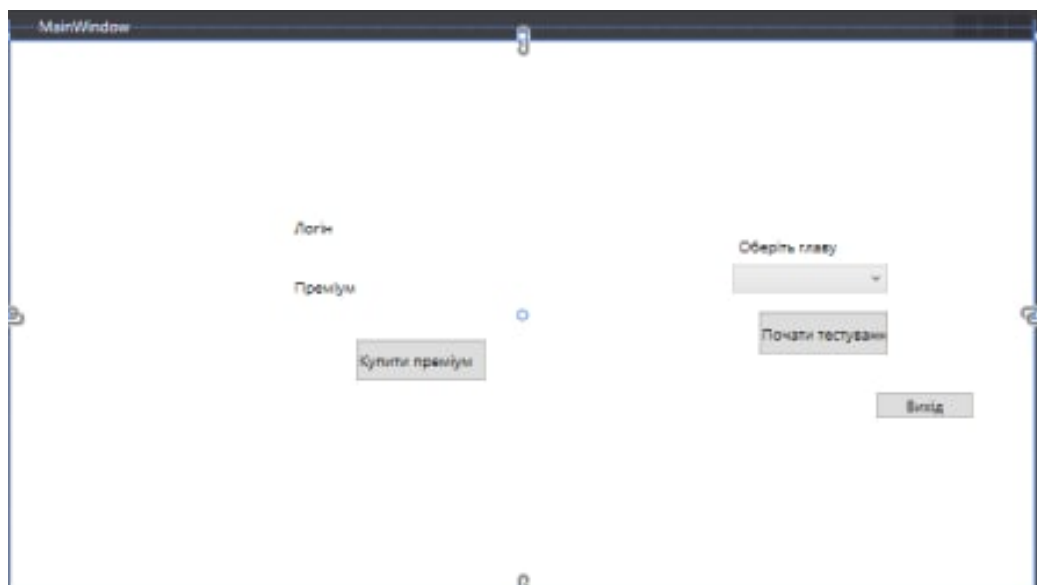


Рис. 2.5. Проект інтерфейсу форми авторизації

## 2.4. Вибір програмного середовища та мови програмування для розробки графічного інтерфейсу

*SQL Server* виступає однією із наймасштабніший популярних систем управління базами даних (СКБД). Згадана СУБД підходить для абсолютно різноманітних проектних рішень: від додатків невеликого розміру до великих високонавантажених масштабних проектів.

*SQL Server* був розроблений компанією *Microsoft*. У поточному програмному модулі буде використовуватися версія 2019, але сама програма бере свій початок у 1987, коли і вийшла перша версія .

*SQL Server* характеризується такими особливостями як:

- висока продуктивність – *SQL Server* працює на практиці дуже швидко.
- *SQL Server* надає шифрування даних – тим самим забезпечуючи належну безпеку та надійність.
- простота – мається на увазі той факт, що із згаданою СУБД відносно легко працювати і вести адміністрування.

Ключовим аспектом в *MS SQL Server* виступає база даних, яка являє собою сховище даних, які організовані специфічним способом і в певному порядку. Для надійного зберігання баз даних та їх ефективного адміністрування застосовуються системи управління базами даних (СУБД) . *MS SQL Server* якраз виступає однією з таких СУБД.

На сьогоднішній день мені відомі такі різновиди мови *SQL*: *PL-SQL* і *T-SQL*, де в *SQL Server* застосовується останній варіант.

Власне тому в рамках поточного програмного забезпечення буде розглядатися саме *T-SQL* (рис. 2.6).



Рисунок 2.6 – Логотип *T-SQL*

В залежності від поставленої задачі, яку вирішує команда *T-SQL*, він може відноситись до одного з наступних типів:

- *DDL* (Мова ідентифікації даних). До такого типу відносяться специфічні команди, які створюють певні таблиці, базу даних, збережені процедури тощо. Взагалі, до *DDL* доцільно віднести такі команди:

- *CREATE*;
- *DROP*
- *ALTER*;
- *TRUNCATE*.

- *DML* (Мова маніпуляції даними). До типу *DML* належать вказівки на вибір даних, їх оновлення, видалення, додавання – тобто такі команди, за допомогою яких ми можемо управляти даними.

До цього типу належать такі команди:

- *SELECT*;
- *INSERT*
- *UPDATE*;
- *DELETE*.

- *DCL* (Мова управління доступу до даних). До цього типу відносять такі команди:

- *GRANT*;
- *REVOKE*.

Основні переваги *Microsoft SQL Server*:

– Повна *Web*-орієнтованість. Реалізація запитів, комплексний аналіз та управління через інформаційні технології. Безпечний та примітивний, у деякій мірі, доступ до ресурсів за допомогою *web*-браузерів, швидкий та легкий пошук необхідних матеріалів та документів.

– Надійність та масштабованість – *SQL Server* забезпечує практично необмежену максимізацію об'ємів зберігання за рахунок зростання надійності і масштабованості системи, застосовуючи усі ключові переваги мультипроцесорної обробки файлів. Це безпечна, надійна, масштабована платформа, яка допомагає у захисті інформації в програмних рішеннях і максимізує рівень її доступності. Разом з тим, оптимізована платформа *SQL Server* відкриває шлях до передбачуваної продуктивності обробки запитів. Інфраструктура *SQL Server* має можливість формувати звіти і аналізувати потоки будь-якого об'єму та складності, одночасно полегшуючи користувачам доступ до ресурсів за рахунок більш тісної інтеграції з *Microsoft Office*.

– Швидкість створення рішень. *SQL Server*, у поєднанні із *.NET Framework* мінімізує час розробки, практичної реалізації та виходу на ринок сучасних додатків, пришвидшує процес пошуку інформації, спрощує для пересічного користувача процес управління, надає змогу застосовувати реалізовані користувачем функції в інших програмних рішеннях, відкриває значні можливості для створення *Web*-додатків.

– Можливість трансформації інформаційних потоків в оперативній пам'яті (*in-memory OLTP*).

– Можливість взаємодії з публічною хмарою *Windows Azure*.

Прогрес комп'ютерних технологій розпочав складний комплексний процес виникнення абсолютно нових рішень знакових систем з метою виконання запису певного набору алгоритмів - мов програмування.

*C#* виступає об'єктно-орієнтованою мовою програмування з безпечною системою типізації для платформи *.NET*(рис. 2.7)



Рис. 2.7. Логотип мови програмування C#

Мова C# розроблена під егідою *Microsoft*. Можна простежити схожість цієї мови з іншими для прикладу із C++ та Java схожий синтаксис C#. Мова наділена строгою статичною типізацією, забезпечує підтримку поліморфізму, а також перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.

Рейтинг використання мов програмування зображено на рис. 2.8. [21]

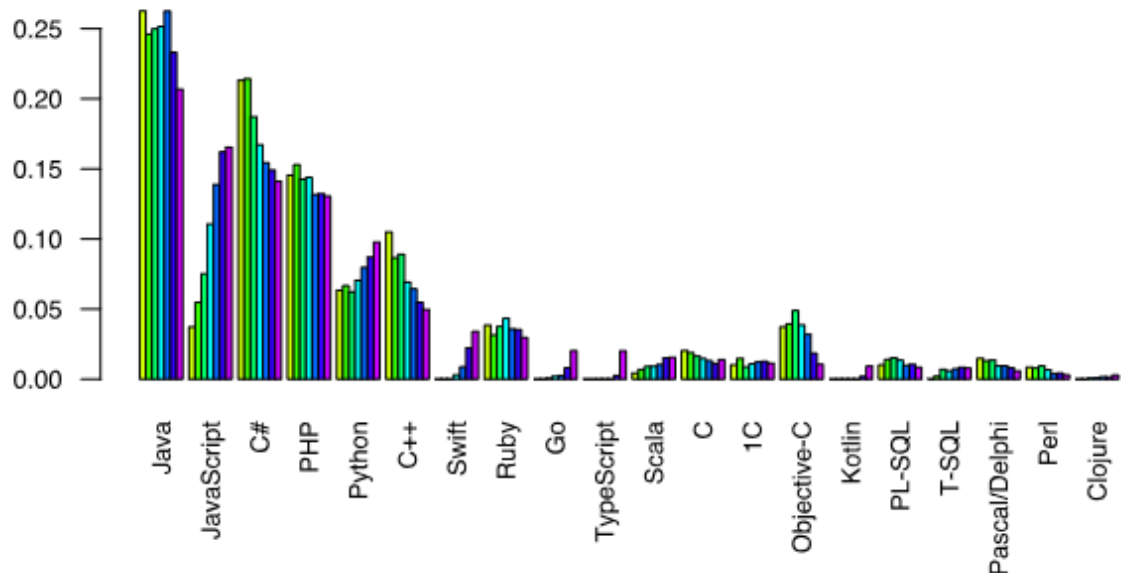


Рис. 2.8. Рейтинг мов програмування

Мова програмування C# переважно використовується для розробки корпоративного програмного забезпечення, фінансових проектів тощо.

На відміну від своїх стратегічних конкурентів, C# легше піддається взаємодії із кодом програм, які написані на інших програмних мовах. Я також хочу відмітити, що на мові C# часто пишуться розширення для інших мов програмування, що застосовуються в якості прошарку між бібліотекою C# і мовою, можливості якої планується розширити.

До переліку принципово важливих рішень, які реалізовані корпорацією *Microsoft* в мові програмування C#, можна віднести наступні:

- комплексний орієнтований підхід до процесу програмування – таке рішення в цілому характерне для ідеології *Microsoft .NET*;
- модифікація подій;
- уніфікована система типізації;
- делегати;
- індексатори;
- перевантажені оператори;
- специфічні механізми *boxing* і *unboxing* для трансформації типів;
- атрибути – специфічний засіб управління та оперування метаданими в *SOM*-моделі.

Можна також зазначити, що мова C# є високорівневою, якщо порівнювати її із іншими мовними рішеннями, це, в свою чергу, означає, що вона має схожі риси, в деякій мірі, із англійською.

C# вже давно підтримує багато корисних функцій. Доцільним вважаю навести ключові функції далі:

- інкапсуляція;
- поліморфізм;
- успадкування;
- статична типізація;
- перевантаження операторів.

При цьому *C #* досить активно розвивається, і разом із кожною наступною версією ми можемо спостерігати все більший перелік функцій ( для прикладу асинхронні методи, лямбда, динамічне зв'язування тощо)

Варто виділити ряд переваг *C #*:

- підтримка переважної більшості продуктів *Microsoft*;
- безкоштовність та доступність ряду інструментів для деяких індивідуальних розробників та невеликих компаній , таких як *Visual Studio, Azure, Windows Server, Parallels Desktop* для *Mac Pro* і ін.;
- типи даних мають фіксований розмір (32-бітний *int* і 64-бітний *long*), що значно збільшує “мобільність” мови та значно спрощує процес програмування;
- автоматичне “прибирання сміття” – це має на увазі процес самостійного звільнення пам'яті, ця програма сама викличе збирач сміття та ретельно очистить пам'ять від усього зайвого;
- низький поріг входження – розуміємо, що *C #* має досить багато схожого з іншими мовами програмування, що дозволяє значно полегшити перехід для програмістів. Досить часто мову *C #* визнають найбільш комфортною та зрозумілою для нових користувачів;
- за допомогою *Xamarin* на *C #* можна писати програми та додатки для таких операційних систем, як *iOS, Android, MacOS i Linux*;
- сьогодні в будь-якому регіоні України є чимало вакантних місць на посаду *C #*-програміста.

Але є у *C #* і деякі недоліки:

- пріоритетна орієнтованість на платформу *Windows*;
- мова безкоштовна тільки для невеликих фірм, індивідуальних програмістів, стартапів і учнів. Для великої компанії покупка ліцензійної версії цієї мови обійдеться досить дорого [22].

Однак, зазначені недоліки не є критичними для вирішення поставленого завдання.

Виділення і об'єднання кращих ідей сучасних мов програмування робить мову C # не просто сумою їх достоїнств, а мовою програмування нового покоління.

В результаті проведеного аналізу і того факту, що деякі навички з написання програм на мові програмування C # у нас вже є, можна зробити висновок про те, що найбільш підходящим мовою програмування є C #.

*Microsoft Visual Studio* (рис. 2.9) – партія продуктів глобальної організації *Microsoft*. Ця серія включає в себе інтегроване середовище проектування та створення програмного модулю та список наступних необхідних інструментальних методів та засобів. Згадані продукти надають змогу створювати консольні програми, програмні рішення, наділені специфічним графічним інтерфейсом, в тому числі з підтримкою технології *Windows Forms*, а також Веб-сайти [20].



Рис. 2.9. Програмне середовище *Visual Studio*

За допомогою вбудованих засобів була створена база даних програмного модулю “*Database*” (рис. 2.10).

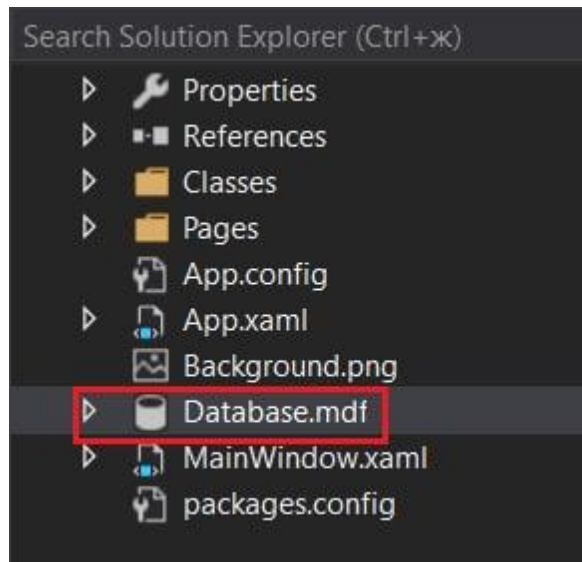


Рис. 2.10. База даних програмного модулю

За допомогою *Visual Studio* та мови *T-SQL* розроблена таблиця даних для роботи з користувачами ( рис. 2.11 – 2.12). Таблиця містить наступні атрибути: *login* (логіни), *password* (паролі), *premium*(дані про преміум-акаунт).

Login	nvarchar(50)	■		
Password	nvarchar(50)	■		
Premium	int	■		
		■		

Рис. 2.11. Графічне відображення таблиці

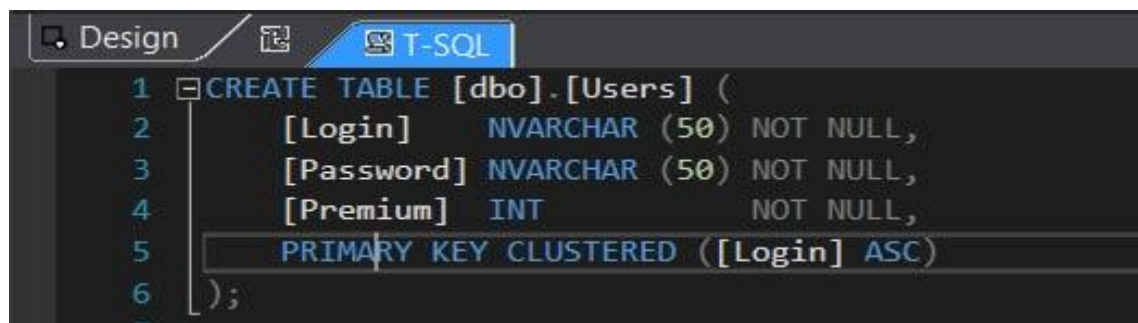


Рис. 2.12. Таблиця даних для роботи з користувачами

Для реалізації графічних інтерфейсів найбільш зручною та найпростішою платформою виступає *WPF*.

Технологія *WPF (Windows Presentation Foundation)* виступає структурною частиною платформи *.NET* і виступає підсистемою для розробки графічних інтерфейсів.

Якщо під час реалізації традиційних програмних рішень на основі *WinForms* за розробку елементів управління і графіки відповідали такі частини

операційної системи *Windows*, як *User32* і *GDI +*, то додатки *WPF* засновані на *DirectX*. У цьому і полягає головна особливість графіки в *WPF*: Використовуючи у роботі *WPF*, значний об'єм роботи по відображенні графіки лягає на графічний процесор на відеокарті, що також дозволяє скористатися апаратним прискоренням графіки.

Далі відзначимо ключові переваги *WPF*:

- Практичне застосування традиційних мов *.NET*-платформи – *C #* і *VB.NET* для створення логічної структури програмного рішення;
- Можливість декларативної ідентифікації графічного інтерфейсу за допомогою специфічної мови розмітки *XAML*, що представляє альтернативу програмної реалізації графіки та структурних елементів управління, а також можливість поєднувати *XAML* і *C # / VB.NET*
- Незалежність від дозволу екрану: у методикі *WPF* наявні структурні елементи вимірюються в одиницях, незалежних від пристрою, додатки на *WPF* із легкістю масштабуються під різні екрани із різноманітним дозволом та розширенням.
- Нові можливості та технології, яких досить складно було досягти в *WinForms*, для прикладу – реалізація тривимірних моделей, прив'язка даних, застосування у роботі таких специфічних елементів, як стилі, шаблони, теми і ін.
- Належну взаємодію з *WinForms*, завдяки чому, для прикладу – в додатках *WPF* передбачається можливість застосовувати традиційні елементи управління з *WinForms*.
- Широкі можливості по реалізації різноманітних додатків та програмних рішень, такі як мультимедіа, багатовимірна графіка тощо.
- Розробка програмних рішень під значну кількість ОС сімейства *Windows* .

У той же час *WPF* має певні обмеження.

Однак такі обмеження стосуються, переважно розробки ігрових платформ, або програють деяким рішенням, якщо їх співставляти. Для прикладу, якщо

порівнювати згадану технологію з іншими рішеннями, то у нашого варіанту у процесі роботи спостерігається в середньому значно вищий обсяг споживання пам'яті.

Однак, такі обмеження не є критичними. Я вважаю, що це з лишком компенсується більш широкими графічними можливостями і підвищеною продуктивністю при відображенні графіки.

## 2.5. Керівництво користувача

З самого початку завантаження з'являється вікно входу в систему. Якщо користувач вже був створений, то він вводить логін і пароль у відповідні поля і якщо введені дані у є таблиці логінів та паролів – виконується вхід в систему з виведення відповідного інформаційного вікна (рис. 2.13). Якщо дані внесені некоректно, чи користувач не створений, програма не буде реагувати.

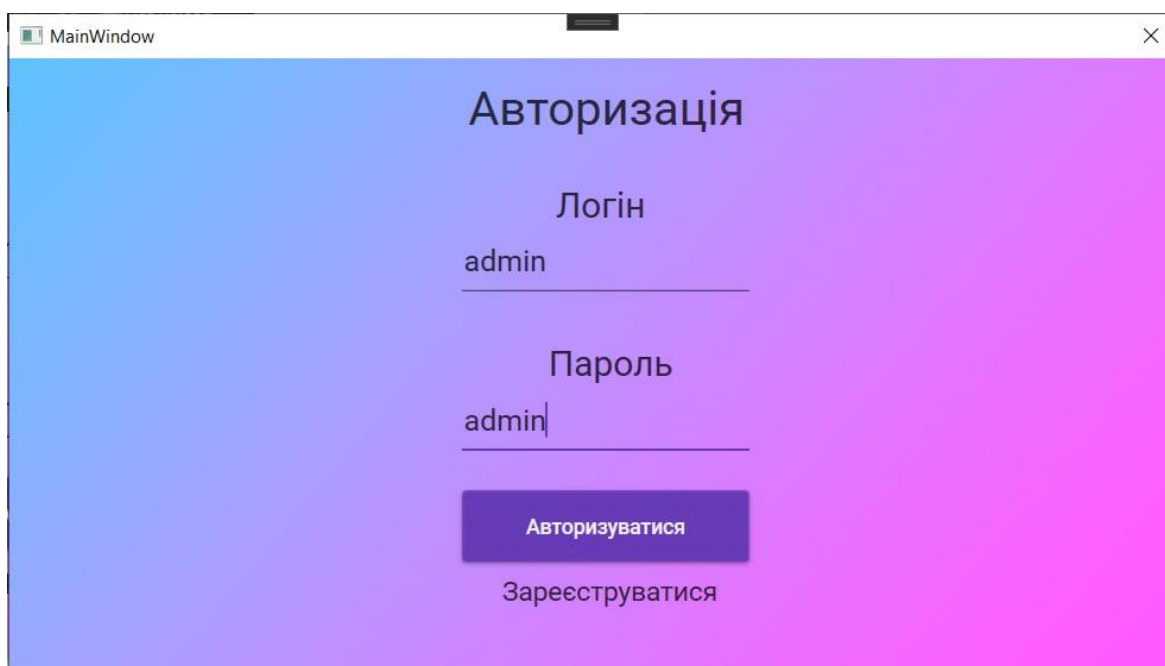


Рис. 2.13. Авторизація в системі

Якщо користувач не був створений, то він переходить на вікно реєстрації. Після введення логіна і пароля виконується команда, яка додає створеного користувача до бази даних (рис. 2.14).

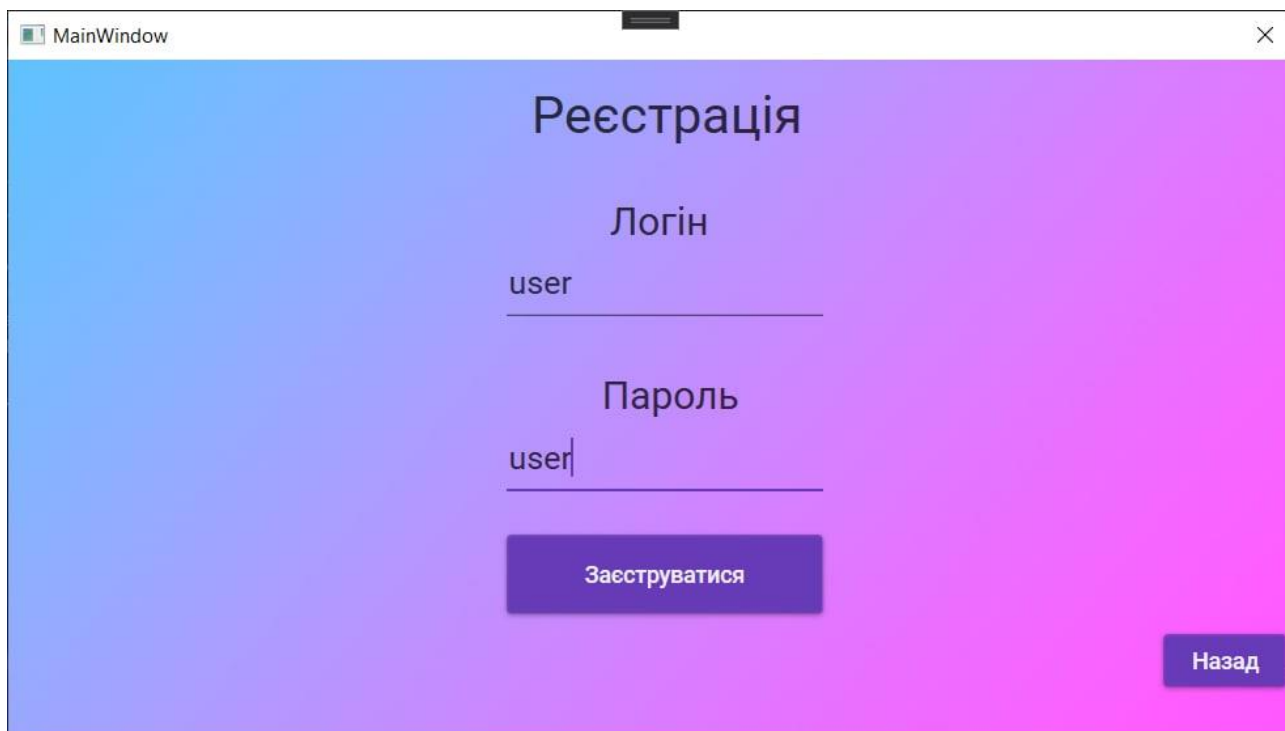


Рис. 2.14. Реєстрація в системі

Після вдалої авторизації користувач потрапляє до головної форми ( рис. 2.15).

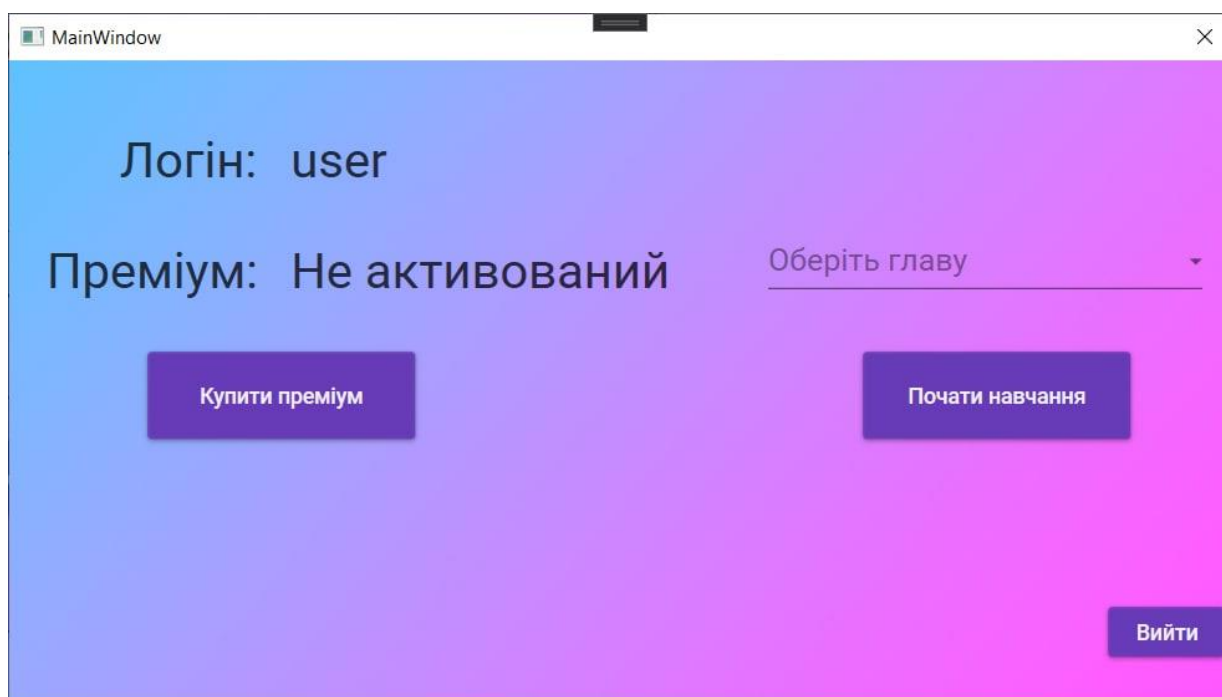


Рис. 2.15. Головна форма

Для користувача передбачена можливість навчання у звичайному режимі або покупка преміум-акаунта. Якщо користувач бажає продовжити роботу із

програмою у звичайному режимі, то у нього є три дисципліни, які він може вивчити та перевірити свої знання ( рис. 2.16).

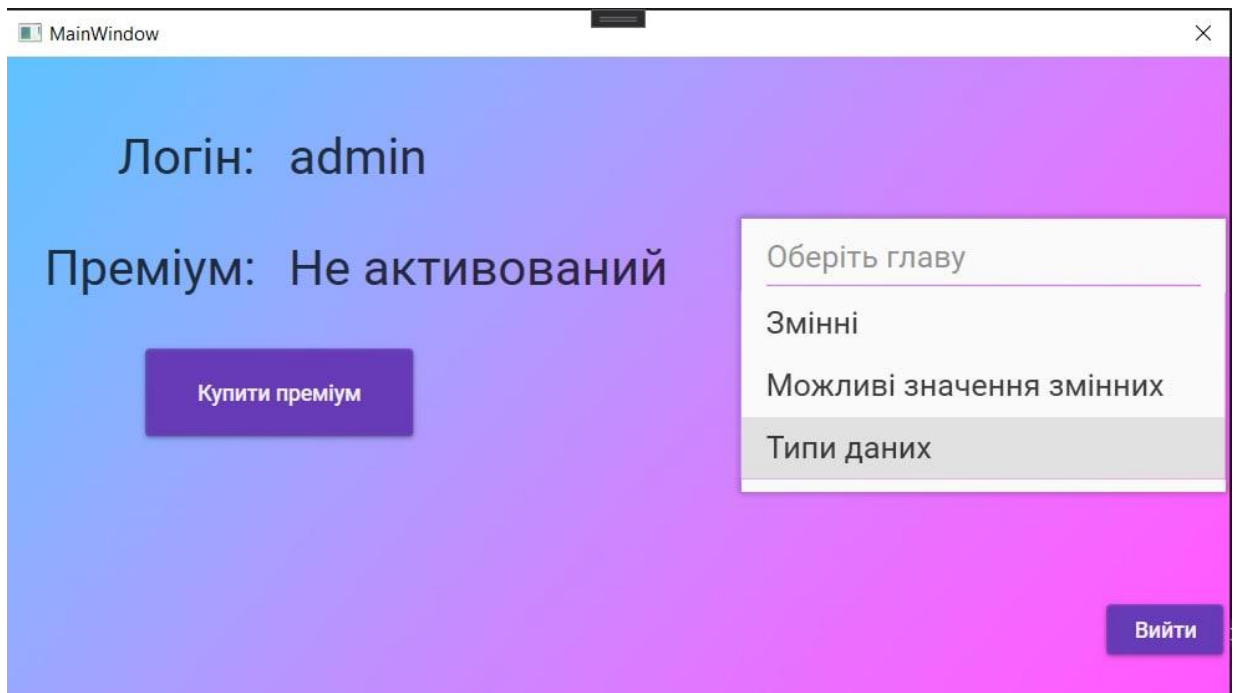


Рис. 2.16. Робота із програмою у звичайному режимі

Якщо користувач хоче збільшити перелік дисциплін, необхідно придбати преміум акаунт. Для цього користувач переходить по клавіші “Купити преміум” та потрапляє на форму, де необхідно ввести дані із кредитної карти ( рис.2.17 ).

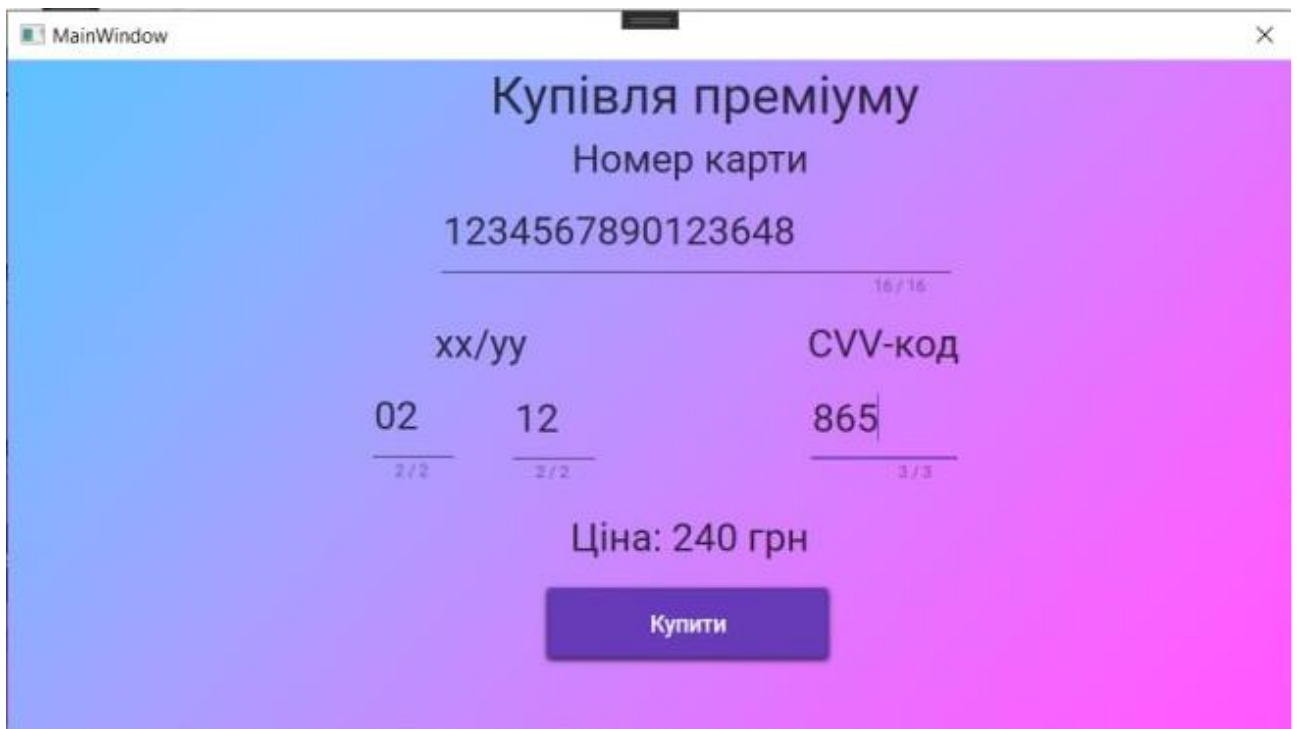


Рис. 2.17. Купівля преміум-акаунта

Якщо користувач успішно придбав преміум акаунт, у нього з'являється значно більший перелік дисциплін для навчання та самоконтролю ( рис. 2.18).

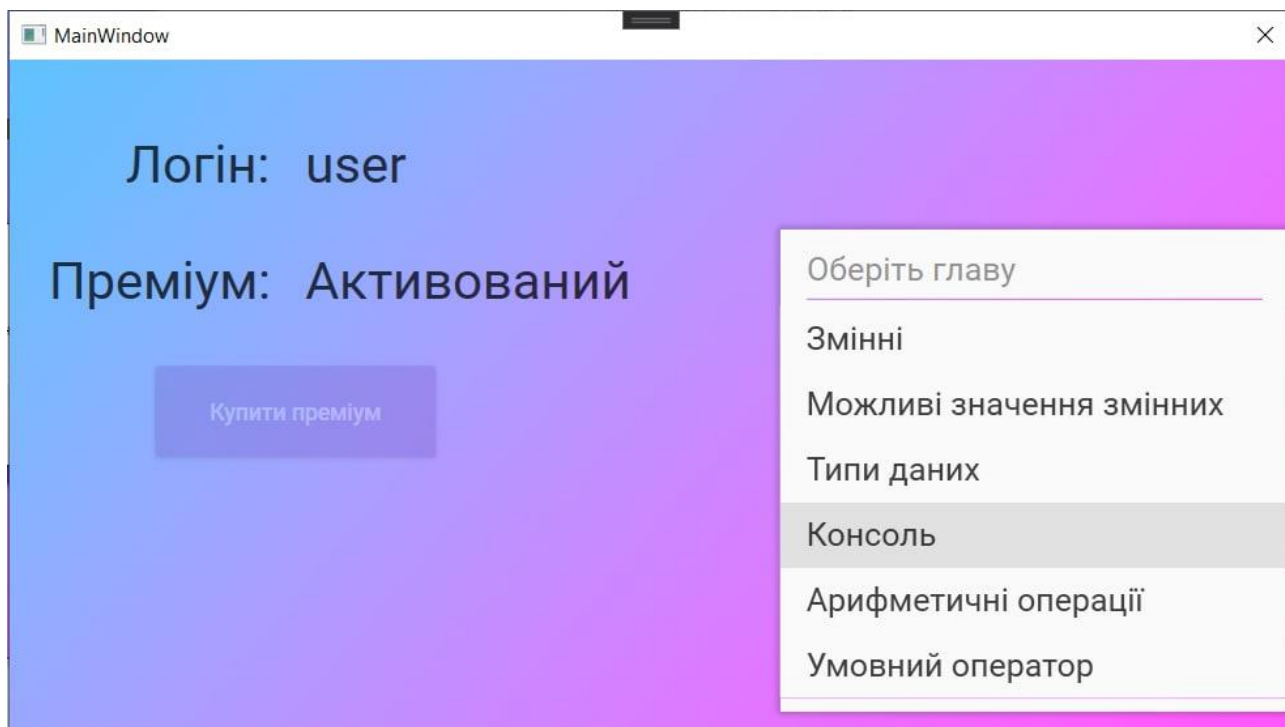


Рис. 2.18. Робота у режимі преміум-акаунта

## Висновок до розділу 2

У третьому розділі розглянуто методологію побудови архітектури програмного модуля.

Обрано необхідний інструментарій, мову програмування і технології розробки, аргументовано доцільність їхнього використання.

Виконано моделювання програмного середовища, а саме побудована діаграма прецедентів та блок-схема. Спроектровано інтерфейс програмного модуля та окреслено ключові вимоги до нього.

## РОЗДІЛ 3

### ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1. План випробувань

Для досягнення максимальної якості тестування, я вважаю за доцільне обрати найоптимальніший та найзручніший метод або спосіб, який у повній мірі забезпечить потрібний результат. Разом з тим варто обрати балансове співвідношення між тим, що буде витрачено на тестування і якістю тестування. Це означає, що процес тестування має проходити найменш збитково, однак охоплювати якнайповніше функціональність системи [10].

Тестування програмного модулю можна ідентифікувати як:

- процес дослідження програмного забезпечення з метою одержання інформаційних даних щодо якості фінального продукту;
- процес перевірки відповідності заявлених до продукту вимог і реально реалізованої функціональності, яка відбувається, в штучно створених ситуаціях і на обмеженому наборі тестів, шляхом спостереження за його роботою.

Для того, щоб провести повне і всебічне тестування відповідності програмного продукту поставленим цілям, складений план тестування:

- Перевірка системи на належну авторизацію;
- Перевірка системи на можливість реєстрації нового користувача ;
- Перевірка системи на перегляд навчальної літератури;
- Перевірка програмного модуля на проходження тестування після засвоєння навчального матеріалу
- Тестування програмного модуля на можливість роботи у двох режимах ( звичайний та преміум).

### 3.2. Перевірка виконання функціональних вимог програмного модуля

Тестування реалізованого програмного модуля проводиться на ПК з операційною системою *Windows 10 Pro*, об'ємом оперативної пам'яті 8,00 ГБ, і на базі процесора *Intel (R) Core (TM) i3-7020U CPU @ 2.30GHz 2.30 GHz*.

Одним із наявних функціоналів розробленої інформаційної системи є навчання. Процес навчання представлено на рис. 3.1.

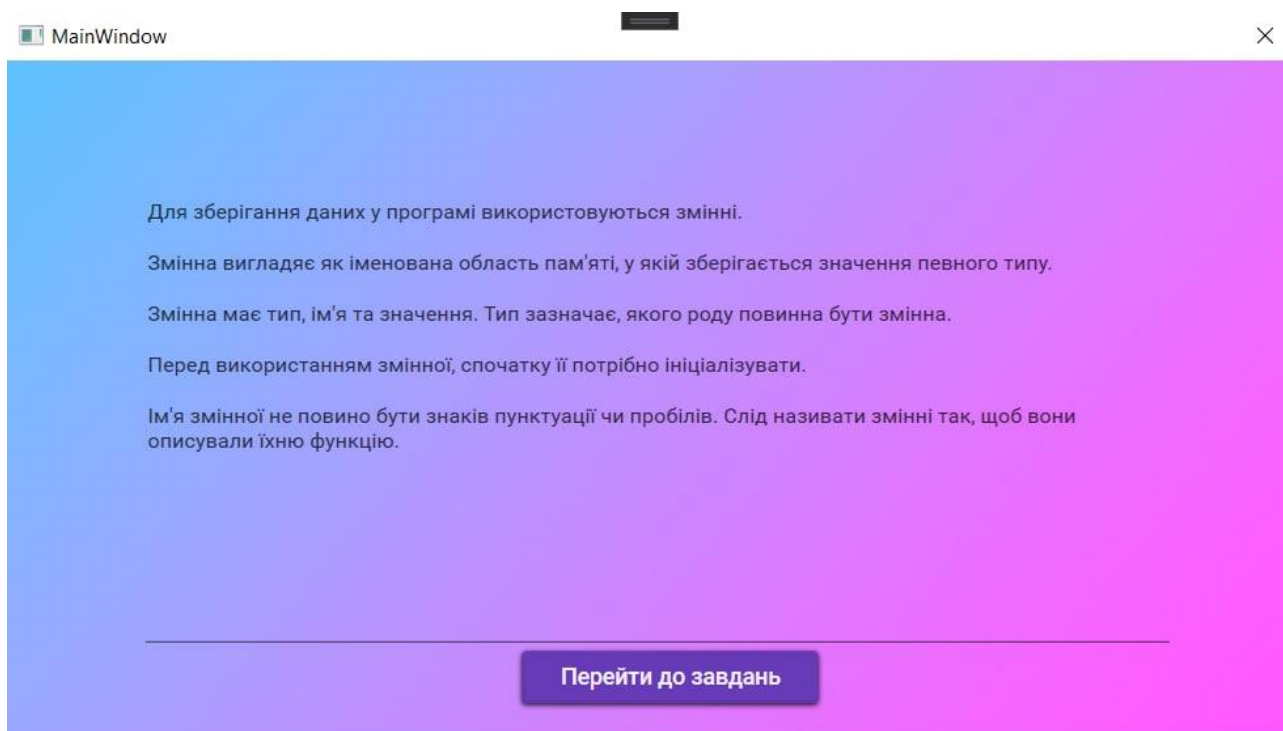


Рис. 3.1. Процес навчання

Після кожної завершені лекції по кожній дисципліні у користувача з'являється можливість самоконтролю у вигляді проходження тестування( рис. 3.2) та виконання практичного завдання ( рис. 3.3).

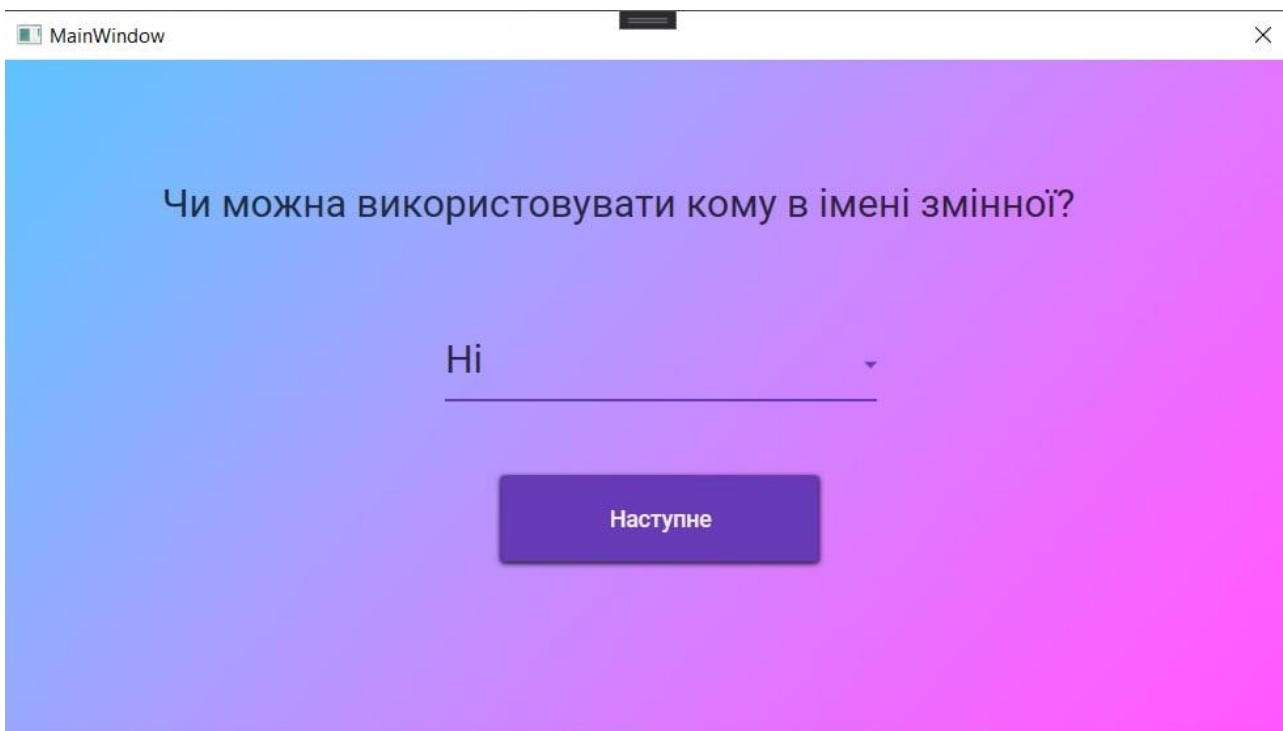


Рис. 3.2. Тестування

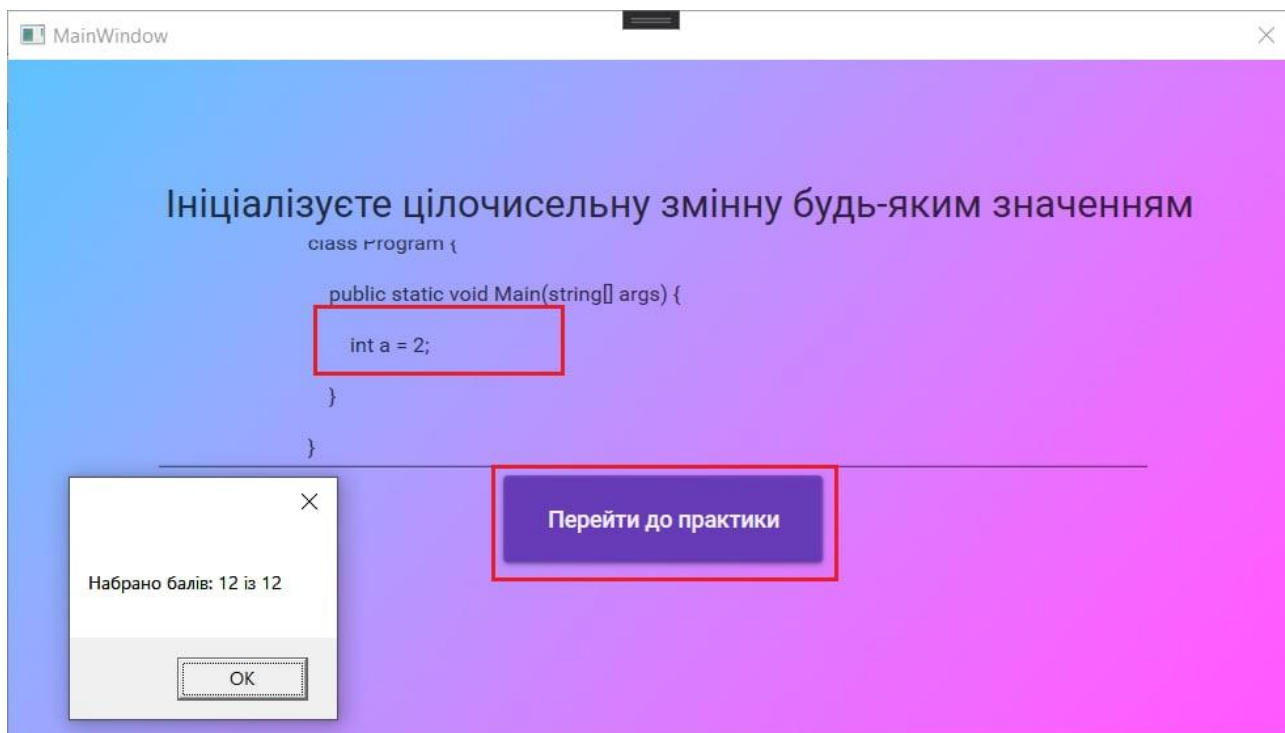


Рис. 3.3 Виконання практичного завдання

По завершенню кожного курсу користувач отримує підсумкову оцінку (рис. 3.4).

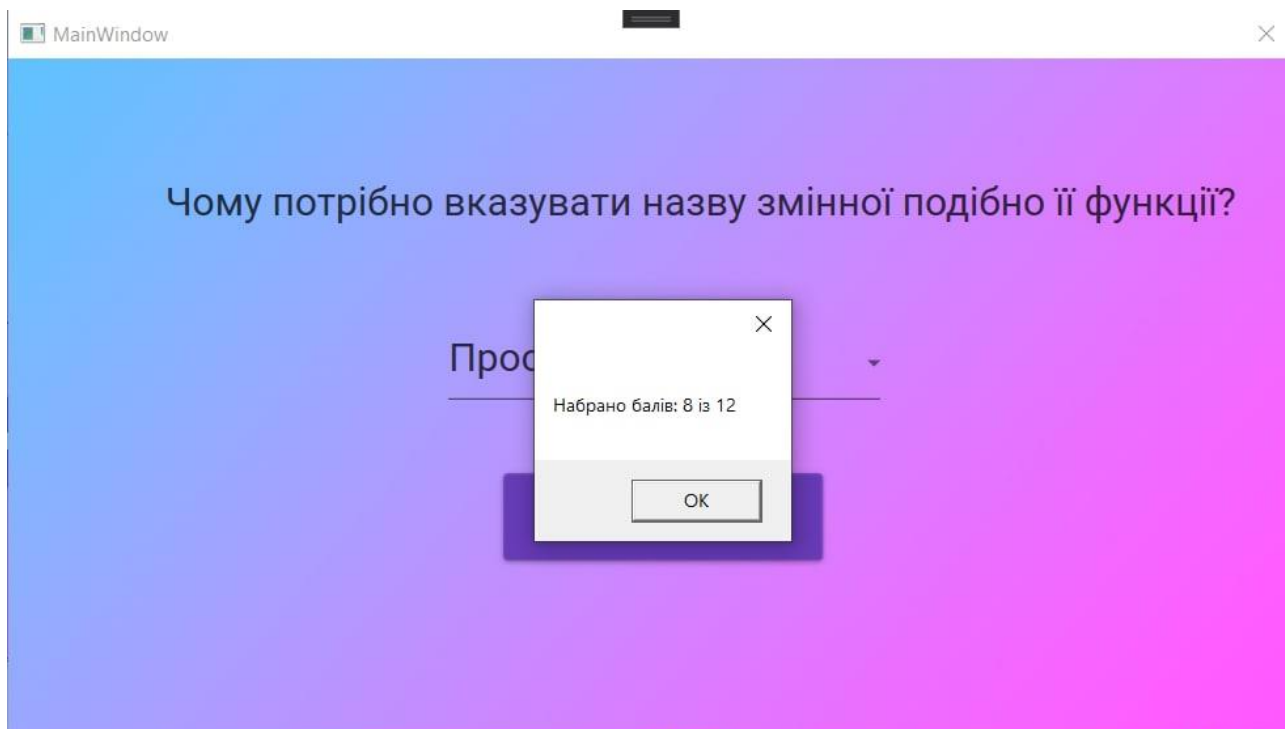


Рис. 3.4. Оцінювання по завершенню тестування

Перевірка програмного рішення в екстремальних умовах передбачала оцінку належної задовільно працездатності розробленої системи для гранично допустимих ситуацій. У процесі такого тестування автоматизована система працювала із вбудованою базою даних. При роботі з такою базою даних автоматизована система не показала явних вповільнень в роботі, а також будь-яких помилок.

Випробування працездатності системи у виняткових ситуаціях ґрунтувалося на застосуванні некоректних, неповних або неточних вихідних даних, перевірці працездатності в разі збоїв.

Для початку було проведено експерименти з введенням некоректних даних в програмі. Для прикладу, при спробі входу в систему під неіснуючим акаунтом або некоректно введеним логіном чи паролем програма просто не пропускала користувача до головної форми та залишалася без змін.

Також під час навчання, коли користувач не обрав коректну дисципліну, а одразу перейшов до навчання отримав пусте вікно ( рис. 3.5).

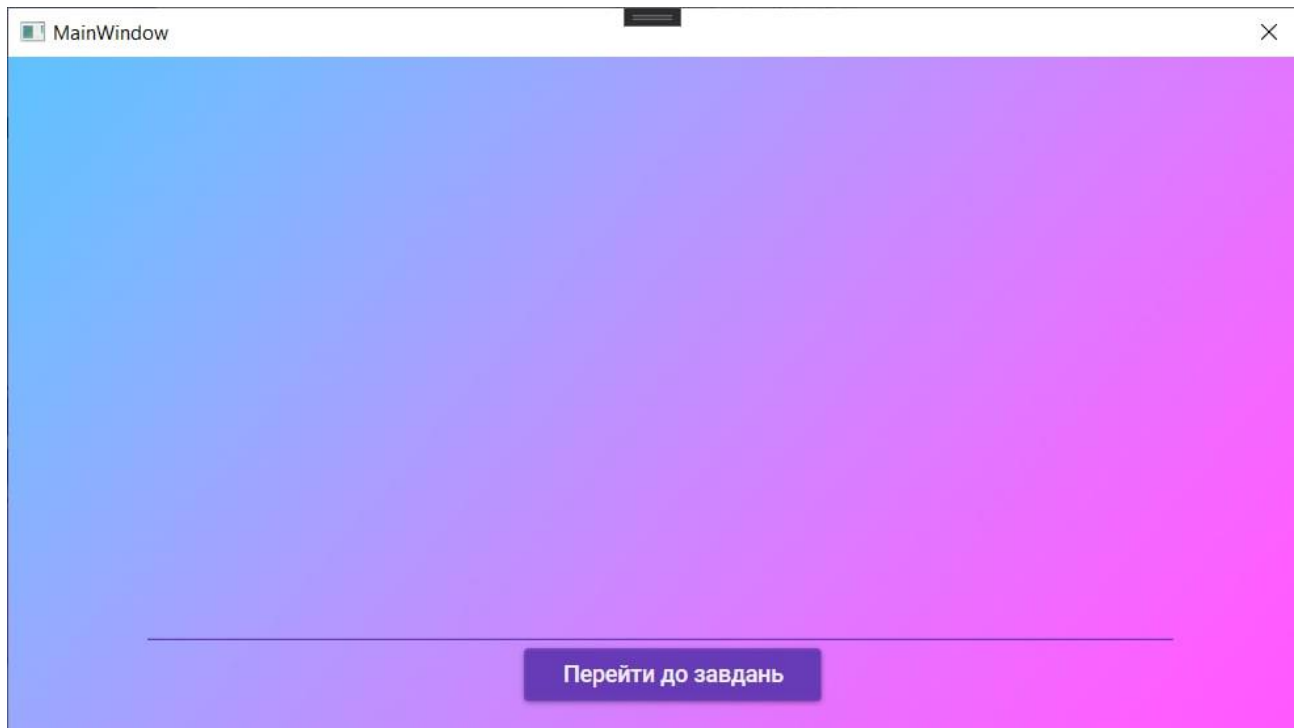


Рис. 3.5. Невірна послідовність дій при роботі з програмою

Отже, результати проведення експерименту показують, що авторизація пройшла успішно, здійснюється відкриття форм і забезпечена робота усіх ключових функцій програмного модулю.

### Висновки до розділу 3

Таким чином, розроблений програмний модуль для підготовки школярів для вступу до ВНЗ повністю відповідає висунутим вимогам. Дана інформаційна система значно полегшує навчальний процес школярам та допомагає у кращому та ефективнішому засвоєнні пройденого матеріалу.

На підставі результатів тестування, можна зробити висновок, що система повністю відповідає вимогам технічного завдання, стабільна і коректно реагує на всі дії користувача.

## ВИСНОВКИ

На сучасному етапі існує необхідність подальшого впровадження інформатизації в вітчизняну освіту. Потрібно розробляти і впроваджувати більшу кількість педагогічних програмних засобів навчання з усіх дисциплін. На сьогоднішній день надзвичайно актуальним постає питання розробки автоматизованих інформаційних систем для організації навчального процесу та належної підготовки абітурієнтів для вступу до ВНЗ, аргументуючи це тим, що в умовах сьогодення учні не отримують достатнього рівня знань для належної підготовки до фахових іспитів

Дипломна робота присвячена вивченню питань програмного модуля для підготовки абітурієнтів для вступу до ВНЗ. Інформаційна система складена на основі тематичного плану курсу дисциплін за навчальний період і створена для поліпшення якості засвоєння курсу дисципліни та належної підготовки до фахових іспитів з метою успішного вступу до ВНЗ.

У роботі вирішено питання, чому саме ця тема актуальна та заслуговує бути розглянутою, також проведений аналіз актуальності інформаційних технологій у сучасному світі, у навчальному процесі та підготовці абітурієнтів, зокрема та доцільність створення програмного модуля за допомогою для підготовки абітурієнта до вступу та для дистанційного навчання.

Проводився аналіз технологій розробки, а саме доцільних мов програмування, аналіз навчальних методів, а також обґрунтування вибору технології розробки програмного модуля. Набуті знання допомогли програмний модуль для підготовки учнів до вступу до ВНЗ. Головним результатом роботи, за допомогою потужної мови програмування третього покоління – *C#* та найефективнішої СКБД *MSSQL*, виступає реалізація програмного модуля для виконання маніпуляцій з даними.

Отже, цілі і завдання моєї кваліфікаційної роботи було чітко досліджені і виконано. Розроблений програмний модуль має зрозумілий і зручний

функціонал і зрозуміло кожному. Я вважаю, що дана програма зменшить час, витрачений вручну на пошук інформації, необхідної для проходження належної підготовки учня до вступу до ВНЗ.

Розроблений програмний продукт задовольняє технічним вимогам споживачів, а його впровадження на ринку є економічно доцільним.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абдрахманова, Б.А. Інформаційні технології в освіті / [Електронний ресурс]. - Режим доступу: <http://www.zkoipk.kz/b2/369-conf.html>
2. Агальцов, В.П. Базы данных. В 2-х т.Т. 1. Локальные базы данных: Учебник / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 352 с.
3. Бази даних та види архітектур інформаційних систем. [Електронний ресурс]. — Режим доступу URL: [http://libraryno.ru/5-bazy-dannyh-i-vidy-arhitektur-ekonomicheskikh-informacionnyh-sistem-2015\\_bd/](http://libraryno.ru/5-bazy-dannyh-i-vidy-arhitektur-ekonomicheskikh-informacionnyh-sistem-2015_bd/)
4. Басюк Т.М. Принципи побудови системи аналізу та просування інтернет-ресурсів / Т.М. Басюк // Вісник Нац. ун-ту “Львівська політехніка” “Комп’ютерні науки та інформаційні технології”. – 2012.
5. Безкорвайный М. М., Костогрызов А. И., Львов В. М. Инструментально-моделирующий комплекс для оценки качества функционирования информационных систем. М.: ”Вооружение, политика, конверсия, 2001, 303 с.
6. Бойко В.В., Савінков В.М. Проектування баз даних інформаційних систем. 1989. - 351 с.
7. Бонч-Бруєвич Г.Ф., Абрамов В.О., Методика застосування Смарт-технології у навчальному процесі : навчальний посібник / Г.Ф. Бонч-Бруєвич, В.О. Абрамов, Т.І. Косенко. Ё К., 2016.
8. Брауде Э.Технология разработки программного обеспечения. – СПб.: Питер, 2004. – 655 с.
9. Вигерс К. Разработка требований к программному обеспечению/ К. Вигерс – Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 2004. – 576с.
10. Голицына, О.Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2004. - 352 с.

11. . Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2012. - 400 с.
12. Дементієвська Н.П., Морзе Н.В. Комп'ютерні технології для розвитку учнів та вчителів // Інформаційні технології і засоби навчання: Зб. наук. праць / За ред. В.Ю.Бикова, Ю.О.Жука / Інститут засобів навчання АПН України. — К.: Аттіка, 2016.
13. Джексон Г. Проекування реляційних баз даних для використання з мікроЕОМ., 1991. - 252 с.
14. Зандстра М. А. Объекты, шаблоны и методики программирования, 3-е издание / М. А. Зандстра. – СПб.: Издательский дом «Питер», 2011.
15. Карпова Т.С. Базы данных: моделі, розробка, реалізація., 2002. - 304 с.
16. Кирилов В.В. Структуризованние мову запитів (SQL)., 1994. - 80 с.
17. Кириллов, В.В. Введение в реляционные базы данных. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. - СПб.: БХВ-Петербург, 2012. - 464 с.
18. Коровайченко Ю., Васильев А.. Дистанційне навчання – це сучасно [Електронний ресурс] – Режим доступу: <http://mon.gov.ua/ua/comments/17415-distantsiyne-navchannya--tsesuchasno.-gazeta-osvita-ukrayini--24-vid>.
19. Корнеев И.К., Машурцов В.А. Информационные технологии в управлении. - М.: ИНФРА-М, 2001. – 158 с.
20. Мартін Дж. Планування розвитку автоматизованих систем. 1984. - 196 с.
21. Лебеденко М. С. Вебметричний ранг як показник ефективності електронного ресурсу підприємства / М. С. Лебеденко // Економічний вісник Національного технічного університету України "Київський політехнічний інститут". - 2014. - № 11
22. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В.Ю. Пирогов. - СПб.: БХВ-Петербург, 2009. - 528 с.

23. Планирование обеспечения надежности информационных систем / Г. Коваль, Т. Коротун, Т. Яблокова, Л. Кузаченко – Проблемы программирования. – 2001. – №3-4. – С. 40 – 47.
24. Про впровадження пілотного проекту «Learnin – SMART навчання»/ Наказ МОН №812 від 12.07.12 року //[Електронний ресурс] Режим доступу: <http://osvita.ua>.
25. Раскин Д. Интерфейс: Новые направления в проектировании компьютерных систем / Д. Раскин. – М.: Символ-Плюс, 2014.
26. Семеніхіна О.В. Нові парадигми у сфері освіти в умовах переходу до SMARTсуспільства [Електронний ресурс] Режим доступу:<http://irbis-nbuv.gov.ua>
27. Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М.: Юрайт, 2013.
28. Тхір І.Л., Галушка В.П., Юзьків А.В. Посібник користувача ПК. — Тернопіль: СМП «Астон», 2017.
29. Фаулер М. Архитектура корпоративных программных приложений (пер. с англ.). – К.; М.; СПб.: Вильямс, 2004. – 544 с.
30. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: ИЦ Академия, 2012. - 320 с.
31. Хаббард Дж. Автоматизоване проектування баз даних. 1984. - 294 с
32. Яцишин В. Технологія забезпечення якості процесу розробки вимог до програмного забезпечення / В. Яцишин // Матеріали науково-технічної конференції «Обчислювальні методи і системи перетворення інформації» Фізико-механічного інституту ім. Г.В. Карпенка НАН України: 7-8 жовтня 2010 р. – Львів: ФМІ НАНУ – 2010 – с. 275 – 278
33. Яцишин В. Визначення якості програмного забезпечення та відповідність його міжнародним стандартам / В. Яцишин // Матеріали десятої 100 наукової конференції Тернопільського державного технічного університету імені Івана Пулюя: 17-18 травня 2006 р – Тернопіль: ТДТУ – 2006. – с. 93.

34. ISO 9126 Software Quality Characteristics [Електронний ресурс].– Режим доступу URL: <http://www.sqa.net/iso9126.html>
35. Matloff N. The Art of R Programming: A Tour of Statistical Software Design. - No Starch Press, 2011. – 154 p.
36. Smart-технології у вищій освіті / [Електронний ресурс]. - Режим доступу: <http://library.fa.ru/exhib.asp?id=199> (дата звернення: 24.02.2021).
37. Smart-технології змінять систему освіти / [Електронний ресурс]. - Режим доступу: [http://www.trainings.ru/library/education\\_experience/?id=14024](http://www.trainings.ru/library/education_experience/?id=14024)

## ДОДАТОК А

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

namespace Knowledge {
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window {
        public static Database DB { get; set; }
        public static User U { get; set; }
        public static int Type { get; set; }
        public MainWindow() {
            InitializeComponent();
            string T = System.IO.Path.GetFullPath(@"../../Database.mdf");
            DB = new Database("Data Source = (LocalDB)\MSSQLLocalDB; AttachDbFilename =" + T + "; Integrated
Security = True");
            F.Navigate(new Login(F));
        }
    }
}
```

## ДОДАТОК Б

```

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

namespace Knowledge.Pages {
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Page {
        public Frame F { get; set; }
        public Login(Frame F) {
            InitializeComponent();
            this.F = F;
        }

        private void Label_PreviewMouseLeftButtonDown(object sender, MouseButtonEventArgs e) {
            Classes.Pages.SetRegisterPage(F);
            F.Navigate(Classes.Pages.RegisterPage);
        }

        private void Button_Click(object sender, RoutedEventArgs e) {
            if (textBox1.Text.Length != 0 && textBox2.Text.Length != 0) {
                List<User> L = MainWindow.DB.Users.ToList();
                for (int i = 0; i < L.Count; i++) {
                    if (L[i].Login == textBox1.Text && L[i].Password == textBox2.Text) {
                        MainWindow.U = L[i];
                        Classes.Pages.SetMainPage(F);
                        F.Navigate(Classes.Pages.MainPage);
                        return;
                    }
                }
            }
        }
    }
}

```

## ДОДАТОК В

```

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

namespace Knowledge.Pages {
    /// <summary>
    /// Interaction logic for Control.xaml
    /// </summary>
    public partial class Control : Page {
        public Frame F { get; set; }
        private int Type { get; set; }
        private int Count { get; set; } = 0;
        private int Points { get; set; } = 0;
        public Control(Frame F, int Type) {
            InitializeComponent();
            this.F = F;
            this.Type = Type;
            if(Type == 0) {
                label1.Content = "Чи можна використовувати кому в імені змінної?";
                combo1.Items.Clear();
                combo1.Items.Add("Так");
                combo1.Items.Add("Ні");
                combo1.Items.Add("Частично");
            }
            if(Type == 1) {
                label1.Content = "Чи можна використовувати для дробового числа текст?";
                combo1.Items.Add("Так");
                combo1.Items.Add("Ні");
                combo1.Items.Add("За бажанням");
            }
            if(Type == 2) {
                label1.Content = "Як виглядає тип цілого числа?";
                combo1.Items.Add("double");
                combo1.Items.Add("string");
                combo1.Items.Add("int");
            }
            if(Type == 3) {
                label1.Content = "Що таке консоль?";
                combo1.Items.Add("Клас");
                combo1.Items.Add("Текст");
                combo1.Items.Add("Об'єкт");
            }
            if (Type == 4) {

```

```

        label1.Content = "Чи можна проводити арифметичні операції?";
        combo1.Items.Add("Так");
        combo1.Items.Add("Ні");
        combo1.Items.Add("Частично");
    }
    if (Type == 5) {
        label1.Content = "Чи є в C# оператор умови?";
        combo1.Items.Add("Так");
        combo1.Items.Add("Ні");
        combo1.Items.Add("Частично");
    }
}
private void button1_Click(object sender, RoutedEventArgs e) {
    if (combo1.SelectedIndex != -1) {
        if (Type == 0) {
            if (Count == 0) {
                if (combo1.SelectedIndex == 1) {
                    Points++;
                }
                combo1.Items.Clear();
                label1.Content = "Чи потрібно вказувати тип змінної?";
                combo1.Items.Clear();
                combo1.Items.Add("Так");
                combo1.Items.Add("Ні");
                combo1.Items.Add("За бажанням");
                Count++;
                return;
            }
            if (Count == 1) {
                if (combo1.SelectedIndex == 0) {
                    Points++;
                }
                combo1.Items.Clear();
                label1.Content = "Чому потрібно вказувати назву змінної подібно її функції?";
                combo1.Items.Clear();
                combo1.Items.Add("Просто так");
                combo1.Items.Add("Для кращої компіляції");
                combo1.Items.Add("Для зручнішого користування");
                Count++;
                button1.Content = "Завершити";
                return;
            }
            if (Count == 2) {
                if (combo1.SelectedIndex == 2) {
                    Points++;
                }
                MessageBox.Show("Набрано балів: " + Points * 4 + " із 12");
                Classes.Pages.SetMainPage(F);
                F.Navigate(Classes.Pages.MainPage);
                return;
            }
        }
        if (Type == 1) {
            if (Count == 0) {
                if (combo1.SelectedIndex == 1) {
                    Points++;
                }
                combo1.Items.Clear();
                label1.Content = "Чи потрібно ініціалізувати змінну?";
                combo1.Items.Clear();
                combo1.Items.Add("Так");
                combo1.Items.Add("Ні");
            }
        }
    }
}

```

```

        combo1.Items.Add("Такого не можна робити");
        Count++;
        return;
    }
    if (Count == 1) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Через який знак відбувається ініціалізація?";
        combo1.Items.Clear();
        combo1.Items.Add("=");
        combo1.Items.Add("(");
        combo1.Items.Add("!=");
        Count++;
        button1.Content = "Завершити";
        return;
    }
    if (Count == 2) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        MessageBox.Show("Набрано балів: " + Points * 4 + " із 12");
        Classes.Pages.SetMainPage(F);
        F.Navigate(Classes.Pages.MainPage);
        return;
    }
}
if (Type == 2) {
    if (Count == 0) {
        if (combo1.SelectedIndex == 2) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Як виглядає тип даних тексту?";
        combo1.Items.Clear();
        combo1.Items.Add("string");
        combo1.Items.Add("char");
        combo1.Items.Add("object");
        Count++;
        return;
    }
}
if (Count == 1) {
    if (combo1.SelectedIndex == 0) {
        Points++;
    }
    combo1.Items.Clear();
    label1.Content = "Куди можна записати значення 3.14?";
    combo1.Items.Clear();
    combo1.Items.Add("char");
    combo1.Items.Add("double");
    combo1.Items.Add("int");
    Count++;
    button1.Content = "Перейти до практики";
    return;
}
if (Count == 2) {
    if (combo1.SelectedIndex == 1) {
        Points++;
    }
    combo1.Visibility = Visibility.Hidden;
    richTextBox.Visibility = Visibility.Visible;
}

```

```

richTextBox.AppendText(@"
    using System.Linq;
    using System;
    using System.Text;
    class Program {
        public static void Main(string[] args) {

            }
        }");
label1.Content = "Ініціалізуєте цілочисельну змінну будь-яким значенням";
Count++;
return;
}
if(Count == 3) {
var csc = new CSharpCodeProvider(new Dictionary<string, string>() { { "CompilerVersion", "v3.5" }
});

var parameters = new CompilerParameters(new[] { "mscorlib.dll", "System.Core.dll" }, "foo.exe", true);
parameters.GenerateExecutable = true;
CompilerResults results = csc.CompileAssemblyFromSource(parameters, new
TextRange(richTextBox.Document.ContentStart, richTextBox.Document.ContentEnd).Text);
if(results.Errors.Count <= 6) {
    Points = Points * 2 + 5 - results.Errors.Count;
    if(new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("int")) {
        Points++;
    }
}
else {
    Points *= 2;
}
MessageBox.Show("Набрано балів: " + Points + " із 12");
Classes.Pages.SetMainPage(F);
F.Navigate(Classes.Pages.MainPage);
return;
}
}
if (Type == 3) {
    if (Count == 0) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Як вивести у консоль текст?";
        combo1.Items.Clear();
        combo1.Items.Add("Console.WriteLine()");
        combo1.Items.Add("Console.ReadLine()");
        combo1.Items.Add("Console()");
        Count++;
        return;
    }
    if (Count == 1) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Чи можна зчитати дані з консолі?";
        combo1.Items.Clear();
        combo1.Items.Add("Ні");
        combo1.Items.Add("Так");
        combo1.Items.Add("Частично");
        Count++;
        button1.Content = "Перейти до практики";
    }
}

```

```

        return;
    }
    if (Count == 2) {
        if (combo1.SelectedIndex == 1) {
            Points++;
        }
        combo1.Visibility = Visibility.Hidden;
        richTextBox.Visibility = Visibility.Visible;
        richTextBox.AppendText(@"
            using System.Linq;
            using System;
            using System.Text;
            class Program {
                public static void Main(string[] args) {

                    }
                }");
        label1.Content = "Виведіть текст: \"Dog\"";
        Count++;
        return;
    }
    if (Count == 3) {
        var csc = new CSharpCodeProvider(new Dictionary<string, string>() { { "CompilerVersion", "v3.5" }
});
        var parameters = new CompilerParameters(new[] { "mscorlib.dll", "System.Core.dll" }, "foo.exe", true);
        parameters.GenerateExecutable = true;
        CompilerResults results = csc.CompileAssemblyFromSource(parameters, new
TextRange(richTextBox.Document.ContentStart, richTextBox.Document.ContentEnd).Text);
        if (results.Errors.Count <= 6) {
            Points = Points * 2 + 4 - results.Errors.Count;
            if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("Console.Write")) {
                Points++;
            }
            if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("Dog")) {
                Points++;
            }
        }
        else {
            Points *= 2;
        }
        MessageBox.Show("Набрано балів: " + Points + " із 12");
        Classes.Pages.SetMainPage(F);
        F.Navigate(Classes.Pages.MainPage);
        return;
    }
}
if (Type == 4) {
    if (Count == 0) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Чи можна додавати текст до числа?";
        combo1.Items.Clear();
        combo1.Items.Add("Так");
        combo1.Items.Add("Ні");
        combo1.Items.Add("Частично");
        Count++;
        return;
    }
}

```

```

if (Count == 1) {
    if (combo1.SelectedIndex == 1) {
        Points++;
    }
    combo1.Items.Clear();
    label1.Content = "Як реалізувати додавання двох цілих чисел?";
    combo1.Items.Clear();
    combo1.Items.Add("5 + 9");
    combo1.Items.Add("\5\" + "\9\");
    combo1.Items.Add("Таке неможливо");
    Count++;
    button1.Content = "Перейти до практики";
    return;
}
if (Count == 2) {
    if (combo1.SelectedIndex == 0) {
        Points++;
    }
    combo1.Visibility = Visibility.Hidden;
    richTextBox.Visibility = Visibility.Visible;
    richTextBox.AppendText(@"
        using System.Linq;
        using System;
        using System.Text;
class Program {
    public static void Main(string[] args) {

        }
    });
    label1.Content = "Додайте два цілих числа та запишіть їх у змінну";
    Count++;
    return;
}
if (Count == 3) {
    var csc = new CSharpCodeProvider(new Dictionary<string, string>() { { "CompilerVersion", "v3.5" }
});
    var parameters = new CompilerParameters(new[] { "mscorlib.dll", "System.Core.dll" }, "foo.exe", true);
    parameters.GenerateExecutable = true;
    CompilerResults results = csc.CompileAssemblyFromSource(parameters, new
TextRange(richTextBox.Document.ContentStart, richTextBox.Document.ContentEnd).Text);
    if (results.Errors.Count <= 6) {
        Points = Points * 2 + 3 - results.Errors.Count;
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("int")) {
            Points++;
        }
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("=")) {
            Points++;
        }
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("+")) {
            Points++;
        }
    }
    else {
        Points *= 2;
    }
    MessageBox.Show("Набрано балів: " + Points + " із 12");
    Classes.Pages.SetMainPage(F);
    F.Navigate(Classes.Pages.MainPage);
    return;
}

```

```

    }
}
if (Type == 5) {
    if (Count == 0) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Як позначається головне слово?";
        combo1.Items.Clear();
        combo1.Items.Add("switch");
        combo1.Items.Add("Console");
        combo1.Items.Add("if");
        Count++;
        return;
    }
    if (Count == 1) {
        if (combo1.SelectedIndex == 2) {
            Points++;
        }
        combo1.Items.Clear();
        label1.Content = "Як реалізувати перевірку умови";
        combo1.Items.Clear();
        combo1.Items.Add("a == 7");
        combo1.Items.Add("a ++ b");
        combo1.Items.Add("Таке неможливо");
        Count++;
        button1.Content = "Перейти до практики";
        return;
    }
    if (Count == 2) {
        if (combo1.SelectedIndex == 0) {
            Points++;
        }
        combo1.Visibility = Visibility.Hidden;
        richTextBox.Visibility = Visibility.Visible;
        richTextBox.AppendText(@"
            using System.Linq;
            using System;
            using System.Text;
            class Program {
                public static void Main(string[] args) {

                    }
                }");
        label1.Content = "Створіть цілочисельну змінну та перевірте, чи вона дорівнює 9";
        Count++;
        return;
    }
    if (Count == 3) {
        var csc = new CSharpCodeProvider(new Dictionary<string, string>() { { "CompilerVersion", "v3.5" }
});
        var parameters = new CompilerParameters(new[] { "mscorlib.dll", "System.Core.dll" }, "foo.exe", true);
        parameters.GenerateExecutable = true;
        CompilerResults results = csc.CompileAssemblyFromSource(parameters, new
        TextRange(richTextBox.Document.ContentStart, richTextBox.Document.ContentEnd).Text);
        if (results.Errors.Count <= 6) {
            Points = Points * 2 + 1 - results.Errors.Count;
            if (new TextRange(richTextBox.Document.ContentStart,
            richTextBox.Document.ContentEnd).Text.Contains("if")) {
                Points++;
            }
        }
    }
}

```

```
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("==")) {
            Points++;
        }
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("7")) {
            Points++;
        }
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("int")) {
            Points++;
        }
        if (new TextRange(richTextBox.Document.ContentStart,
richTextBox.Document.ContentEnd).Text.Contains("=")) {
            Points++;
        }
    }
    else {
        Points *= 2;
    }
    MessageBox.Show("Набрано балів: " + Points + " із 12");
    Classes.Pages.SetMainPage(F);
    F.Navigate(Classes.Pages.MainPage);
    return;
}
}
}
}
}
```