

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення он-лайн опитувань та тестувань»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

підпис студента

Юрченко Станіслава
Сергійовича

Науковий керівник
кандидат педагогічних наук,
доцент

підпис керівника

Котенко Наталія
Олексіївна

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки

Криворучко О.В.

"20" жовтня 2020 р.

Завдання

на випускнуну кваліфікаційну роботу студентіві

Юрченко Станіславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка програмного
забезпечення он-лайн опитувань та тестувань»

Затверджена наказом ректора від "30" жовтня 2020 р. № 3225

2. Строк здачі студентом закінченої роботи 02 травня 2021 р.

3. Цільова установка та вихідні дані до роботи

Мета роботи - аналіз та вирішення існуючих проблем в рамках обраної теми,
результатом якого стане створення відповідного он-лайн сервісу.

Об'єкт дослідження - система для створення он-лайн опитувань та тестувань.

Предмет дослідження - технології розробки програмного забезпечення для
створення он-лайн опитувань та тестувань.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

Розділ 1 ПРОБЛЕМАТИКА ТА ЗАВДАННЯ

1.1 Тестування та опитування в режимі он-лайн

1.2 Технічне завдання

1.3 Висновки до розділу 1

Розділ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розробка бази даних

2.2 ASP.NET Core 5 та Entity Framework Core 5

2.3 Проектування архітектури системи

2.4 Висновки до розділу 2

Розділ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка функціоналу

3.2 Опис інтерфейсу

3.3 Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. ПРОБЛЕМАТИКА ТА ЗАВДАННЯ</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021 – 21.05.2021	21.05.2021
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.06.2021	01.06.2021
11.	<i>Здача прошитої випускної кваліфікаційної роботи на кафедру</i>	02.06.2021	02.06.2021
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020р.

8. Науковий керівник випускної кваліфікаційної роботи

Котенко Н. О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М. О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Юрченко С. С.

(прізвище, ініціали, підпис)

ЗМІСТ

ВСТУП	3
Розділ 1 ПРОБЛЕМАТИКА ТА ЗАВДАННЯ	5
1.1 Тестування та опитування в режимі он-лайн.....	5
1.2 Технічне завдання	9
1.3 Висновки до розділу 1	11
Розділ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
2.1 Розробка бази даних	12
2.2 ASP.NET Core 5 та Entity Framework Core 5	18
2.3 Проектування архітектури системи	19
2.4 Висновки до розділу 2	28
Розділ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	30
3.1 Розробка функціоналу	30
3.2 Опис інтерфейсу.....	32
3.3 Висновки до розділу 3	39
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТКИ	44

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Зміст	2	43
Зав. кафедри		Криворучко О.В.		14.12.2020				
Керівник		Котенко Н.О.		14.12.2020				
Гарант		Цензура М.О.		14.12.2020				
Розроб.		Юрченко С.С.		14.12.2020	<i>Зміст</i>	<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		

ВСТУП

Інтегрування електронних технологій в усі сфери сучасного життя людини щодня набирає обертів. Сфера комунікацій хоча і є найбільш популярною та досить розвиненою, але й досі має низку проблем в різних областях що потребують вирішення. Однією з таких – є область з проведення опитувань та тестувань в рамках певної групи осіб. Найбільшу долю попиту створює сфера навчання та, зокрема, її дистанційна форма. В умовах всесвітнього карантину актуальність цієї теми досить швидко зросла і не збирається зменшуватись. Нова форма процесу здобування знань досить перспективна і користується попитом, через що велика кількість компаній інвестують у її розвиток. Тому обрана тема *випускної кваліфікаційної роботи* «Розробка програмного забезпечення он-лайн опитувань та тестувань». Основною проблематикою є контроль доступів, захищеність даних та системи від спроб впливу на роботу алгоритмів програми, гнучкість і зручність додатку.

Метою дослідження випускної кваліфікаційної роботи є аналіз та вирішення існуючих проблем в рамках обраної теми, результатом якого стане створення відповідного он-лайн сервісу.

Об'єктом дослідження є система для створення он-лайн опитувань та тестувань.

Предметом – технології розробки програмного забезпечення для створення он-лайн опитувань та тестувань.

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		В	3	43
Зав. кафедри		Криворучко О.В.		14.12.2020	<i>Вступ</i>			
Керівник		Котенко Н.О.		14.12.2020				
Гарант		Цензура М.О.		14.12.2020				
Розроб.		Юрченко С.С.		14.12.2020				

Завдання дослідження:

- вирішення проблематики контролю в області он-лайн тестувань та опитувань;
- реалізація анонімності при проведенні опитувань серед визначеної групи осіб.

Методи дослідження: аналіз та моделювання.

Моделювання - універсальний метод наукового пізнання, що припускає вивчення оригіналу об'єкта шляхом створення й дослідження його копії - моделі, що заміщає оригінал з певних сторін, що цікавлять дослідника. Основна перевага моделювання - цілісність подання інформації.

Аналіз передбачає роздроблення цілого на складові елементи, тобто виділення ознак предмету для вивчення їх окремо як частини цілого.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						4
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1

ПРОБЛЕМАТИКА ТА ЗАВДАННЯ

1.1 Тестування та опитування в режимі он-лайн

З розвитком інтернет технологій все більше сфер людського життя переходять в он-лайн. І причина цьому не тільки більша кількість можливостей для вирішення існуючих проблем, а й зручність та доступність інструментарію, адже зі зростанням попиту та, як наслідком, розвитком технологій, розробка та підтримка веб-сервісів стає більш дешевою. Через всесвітню епідемію та карантин попит на он-лайн послуги в сфері навчання почав зростати з неймовірною швидкістю.

Дистанційне навчання досить перспективне. Воно постійно вдосконалюється, але існує ряд проблем, дослідження яких ведеться в даний час. З'являються нові методи та шляхи їх вирішення. В основі технології дистанційного навчання лежить цілеспрямована підготовка респондентів до здачі іспитів з тієї чи іншої теми з використанням різних засобів, в тому числі – тестування в електронній формі.

Сформулюємо основні проблеми, що відносяться до систем дистанційного тестування:

1. Відсутність можливості достовірно визначити, чи респондент пройшов тестування самостійно. Він міг використовувати допоміжні носії інформації або засоби доступу до неї. Також є можливість проходження тестування більш обізнаною людиною.

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		P1	5	43
Зав. кафедри		Криворучко О.В.		19.02.2021	<i>Проблематика та завдання</i>			
Керівник		Котенко Н.О.		19.02.2021				
Гарант		Цензура М.О.		19.02.2021				
Розроб.		Юрченко С.С.		19.02.2021				
					<i>Факультет інформаційних технологій, 4 курс, 7 група</i>			

2. Проблема відкритого доступу до тестування що дає можливість респондент збільшити кількість спроб проходження тесту, пройти тест в анонімному режимі для отримання відповідей якщо це передбачено системою.
3. Проблема захисту програмного забезпечення від модифікації його коду, з метою зміни алгоритму оцінювання результатів тестування або іншого коду.
4. Проблема захисту правильних відповідей у випадку тестування за допомогою кейс-технологій.

Система дистанційного тестування повинна мати достатні засоби захисту від несанкціонованого втручання і підміни реальних респондентів їх дублерами. Практично відсутня можливість достовірно визначити, чи пройшов респондент тестування самостійно. Для цього завдання він цілком міг використовувати іншу людину. При використанні даної системи в аудиторіях, ніяких складнощів не виникає, так як вони знаходяться під контролем екзаменаторів. Але орієнтація освіти на дистанційне навчання вносить свої корективи. Виникає потреба в можливості використання даного програмного забезпечення респондентом на своїй локальній машині. Це, мабуть, найскладніше завдання. Апаратні рішення даної проблеми не розглядаються через їх досить малу поширеність та велику вартість. Один із можливих програмних варіантів може бути використання індивідуальності користувача що проявляється в спілкуванні з комп'ютером. Наприклад звичкою використовувати основну чи додаткову частину клавіатури, характер "здвоєних" і "прибудованих" натискань клавіш, улюблених прийомів роботи з комп'ютером, за допомогою яких можна виділити конкретну людину що працювала на даній машині. Зрозуміло, по двом-трьом, навіть по десяти натиснутим клавішам відрізнити користувача неможливо, потрібна досить велика статистика. При наборі ключової фрази комп'ютер дозволяє зафіксувати багато різних параметрів, але для ідентифікації найбільш зручно використовувати час,

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						6
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

витрачений на введення окремих букв. А повторивши введення фрази кілька разів, в результаті отримаємо безліч тимчасових інтервалів для кожного символу. На базі отриманих значень завжди можна розрахувати середній час введення кожного символу, допустиме відхилення від середнього, і зберігати ці результати як еталони для кожного користувача. Це вимагає розробки складних алгоритмів перевірки та ідентифікації. До того ж це актуально при досить тривалій роботі користувача на комп'ютері. При проходженні тестування респонденту доведеться вводити зовсім небагато текстових даних. Подібна ідентифікація для даної роботи також не прийнятна. Доступне технічне рішення для проблеми стовідсоткової ідентифікації придумати поки не вдалося. Вихід може бути в створенні поступового поетапного контролю знань з метою формування труднощів для підміни особи що проходить тестування.

Ще однією досить складною проблемою тестування в он-лайн режимі є можливість респондентів використовувати допоміжні носії інформації або засоби доступу до неї. Через відсутність повного контролю з боку екзаменатора при дистанційному тестуванні, найпопулярнішим є рішення в створенні обмежень в часі та нетиповій постановці питань що ускладнює пошук відповідей в інтернеті.

Проблему контролю доступу до тестувань вирішує система авторизації користувачів. Класичне рішення в створенні персонального логіну та паролю для кожного користувача дає змогу також контролювати кількість спроб при проходженні тестування та доступ до правильних відповідей.

Найпопулярнішими інструментами для вирішення питань захисту коду від модифікацій та спроб взлому системи у випадку тестування за допомогою веб-сервісів – є використання сесій, валідація даних що надходить та ідентифікація відправника, шифрування трафіку між клієнтською та серверною частинами. Але основа безпеки будь-якої системи - її архітектура, інкапсуляція підсистем та розмежування доступу за ролями.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						7
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Також в межах дистанційного спілкування існує попит на засоби проведення опитувань серед певної групи осіб. Для збільшення об'єктивності результатів та зменшення психологічного впливу на відповіді опитуваних людей через контроль з боку суб'єкта що проводить опитування – анонімність має бути невід'ємною складовою. Основними проблемами, що відносяться до проведення анонімних опитувань можна назвати:

1. Проблема доступу до опитування для людей, відповіді яких не повинні впливати на кінцеву статистику за результатами опитування.
2. Проблема контролю за кількістю голосів серед людей, що мають доступ до опитування.

Для вирішення питання що до проблеми контролю доступу до опитування, зберігаючи при цьому анонімність опитуваних, може бути використана відповідна архітектура системи. В цій системі присутня ідентифікація користувачів, але відсутній зв'язок між опитуваним та його відповідями. Всі відповіді в системі зберігаються без ідентифікаторів автора, що дозволяє досягти анонімності в межах окремої групи опитуваних. Але система також має бути однаково відкритою для всіх користувачів, що дасть змогу переконатися у відсутності додаткових систем стеження та викликати довіру до веб-сервісу з боку опитуваних.

Окрім вище перерахованих проблем при проведенні он-лайн тестувань та опитувань існують проблеми гнучкості налаштувань системи та її зручність у використанні при вирішенні поставлених завдань. Основним показником зручності є простота інтерфейсу, автоматизація усіх можливих підсистем та лояльність системи до помилок користувача.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						8
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1.2 Технічне завдання

Загальні відомості

Повне найменування системи – Surveys and Tests service;

Скорочене найменування системи – Surveys&Tests;

Плановий термін початку робіт – 21.09.2020 року;

Плановий термін закінчення робіт – 14.05.2021 року;

Результатом роботи вважати повністю розроблену, протестовану та розміщену в мережі інтернет систему що відповідає усім пред'явленим вимогам;

Головні бенефіціари – екзаменатори та інтерв'юери;

Потенційні користувачі системи – респонденти, інтерв'юери, екзаменатори;

Мета та призначення створення системи

Призначення системи – проведення тестувань та опитувань за допомогою веб-сервісу, що задовольняє потреби потенційних користувачів та вирішує проблематику в сфері дистанційного контролю знань та анонімності опитувань серед визначеної групи осіб.

Мета системи – об'єднати найкращі програмні рішення для реалізації он-лайн сервісів опитувань та тестувань в єдину систему, що має зручний інтерфейс та задовольняє потреби потенційних користувачів.

Вимоги до системи

Вимоги до архітектури системи:

- система повинна бути розроблена на платформі ASP.NET Core 5, з можливістю розгортання в будь-якій операційній системі;
- архітектура системи повинна відповідати шаблону проектування Model View Controller;
- в якості бази даних використовувати MS SQL та нормалізувати її до 4 форми;

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						9
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

- в якості реалізації моделей та зв'язку з базою даних використовувати Entity Framework Core 5;

- представлення реалізувати за допомогою Razor Pages.

Вимоги до системи доступу:

- система повинна мати засоби підтвердження електронної пошти користувача;

- доступ до тестів та опитувань відкривається тільки для певної групи осіб;

- додавати осіб до групи може тільки творець групи;

- розміщувати свої тести та опитування в групі може кожен її учасник;

- налаштування доступу до тесту повинні включати в себе обмеження часу, обмеження кількості спроб та регулювання відкритості правильних відповідей;

Вимоги до системи тестування:

- тести мають включати в себе 5 типів питань, а саме: вибір однієї правильної відповіді, вибір декількох правильних відповідей, правильне співставлення відповідей, написання відповіді, визначення правильних та неправильних тверджень;

- тести складаються з блоків питань, кожен блок повинен містити питання, картинку за бажанням автора, кількість балів за питання, блок варіантів відповідей;

- перевірка відповідей відбувається на стороні сервера;

- по закінченню часу на тестування, тест має автоматично відправлятися на перевірку;

- налаштування доступу має включати в себе кількість спроб на проходження тестування.

- результати за тестуванням повинні включати всі спроби респондентів;

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						10
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Вимоги до системи опитування:

- опитування мають включати в себе 4 типи питань, а саме: вибір однієї відповіді, вибір декількох відповідей, написання відповіді, погодження або непогодження з твердженням;
- опитування складаються з блоків питань, кожен блок повинен містити питання, картинку за бажанням автора, блок варіантів відповідей, блок коментування;
- результати опитування повинні зберігатись та подаватись у вигляді статистики, без прив'язки до опитуваних.

Вимоги до системи безпеки:

- шифрування всього трафіку між клієнтом та сервером;
- валідація та перевірка авторизації при кожному запиті до серверу;
- використання сесії для обмеження в часі;

1.3 Висновки до розділу 1

У результаті аналізу загальної проблематики з питань проведення тестувань та опитувань в режимі он-лайн встановлено, що основними проблемами з боку тестувань стали: контроль доступу до тестів та правильних відповідей, відсутність контролю за діями учня, що направлені на порушення правил тестування, система безпеки та захисту від спроб впливу на роботу алгоритмів програми. Основною проблемою опитувань стало питання збереження анонімності за умови контролю доступу до опитування. На основі розглянутих проблем та найбільш прийнятних варіантів їх вирішення складено технічне завдання на створення системи для проведення он-лайн опитувань та тестувань, що задовольняло б потреби потенційних користувачів.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						11
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розробка бази даних

Для реалізації системи відповідно до вимог завдання, необхідно розробити базу даних для зберігання інформації використовуючи MS SQL. Центральним аспектом в MS SQL, як і в будь-якій СУБД, є база даних. База даних являє сховище даних, організованих певним способом. Для організації баз даних MS SQL використовує реляційну модель. Реляційна модель передбачає зберігання даних у вигляді таблиць, кожна з яких складається з рядків і стовпців. Кожен рядок зберігає окремий об'єкт, а в стовпчиках розміщуються атрибути цього об'єкта. Для ідентифікації кожного рядка в рамках таблиці застосовується первинний ключ. В якості первинного ключа може виступати один або декілька стовпців. Використовуючи первинний ключ, ми можемо посилатися на певну рядок в таблиці. Відповідно два рядки не можуть мати один і той же первинний ключ. Через ключі одна таблиця може бути пов'язана з іншого. Для взаємодії з базою даних застосовується мова SQL.

Проектування бази даних складається з створення 3 моделей, а саме:

- концептуальна;
- логічна;
- фізична;

Для створення концептуальної моделі необхідно встановити зв'язки між головними об'єктами (Рис. 2.1.).

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		Р2	12	43
Зав. кафедри		Криворучко О.В.		05.03.2021	<i>Проектування архітектури програмного забезпечення</i>	<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		
Керівник		Котенко Н.О.		05.03.2021				
Гарант		Цензура М.О.		05.03.2021				
Розроб.		Юрченко С.С.		05.03.2021				



Рис. 2.1. Концептуальна модель бази даних

Створення логічної моделі бази даних – це етап, на якому на підставі інформаційної моделі предметної області бази даних створюється логічна структура бази даних, незалежна від її реалізації.

Для створення бази даних нам потрібно 16 таблиць які будуть пов'язані між собою, а саме:

- «Profile»

зберігає дані про користувачів;

- «Group»

зберігає дані про групи користувачів;

- «GroupProfile»

проміжна таблиця для створення зв'язку Many-to-Many між таблицями «Group» та «Profile»;

- «Tests»
зберігає дані про тест;

- «Test_parts»
зберігає дані про блоки з яких складається тест;

- «Test_results»
зберігає дані про результати тестування;

- «Tests_access»
зберігає дані про відкриті доступи до тестів;

- «DictionaryDB_tests»
реалізує словник для збереження даних про варіанти відповідей до тестів;

- «Surveys»
зберігає дані про опитування;

- «Survey_parts»
зберігає дані про блоки з яких складається опитування;

- «Surveys_access»
зберігає дані про відкриті доступи до опитувань;

- «Respondents»
зберігає дані за списком людей що пройшли опитування;

- «DictionaryDB_surveys»
реалізує словник для збереження даних про статистику за результатами опитування;

- «ListDB_ints»
реалізує список для збереження масиву даних з числовим типом;

- «ListDB_strings»
реалізує список для збереження масиву даних з строковим типом;

- «List_Answer_surveys»
реалізує список для збереження даних про варіанти відповідей до опитувань;

Провівши всі з'єднання можливо спроектувати логічну модель (Рис. 2.2.).

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						14
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

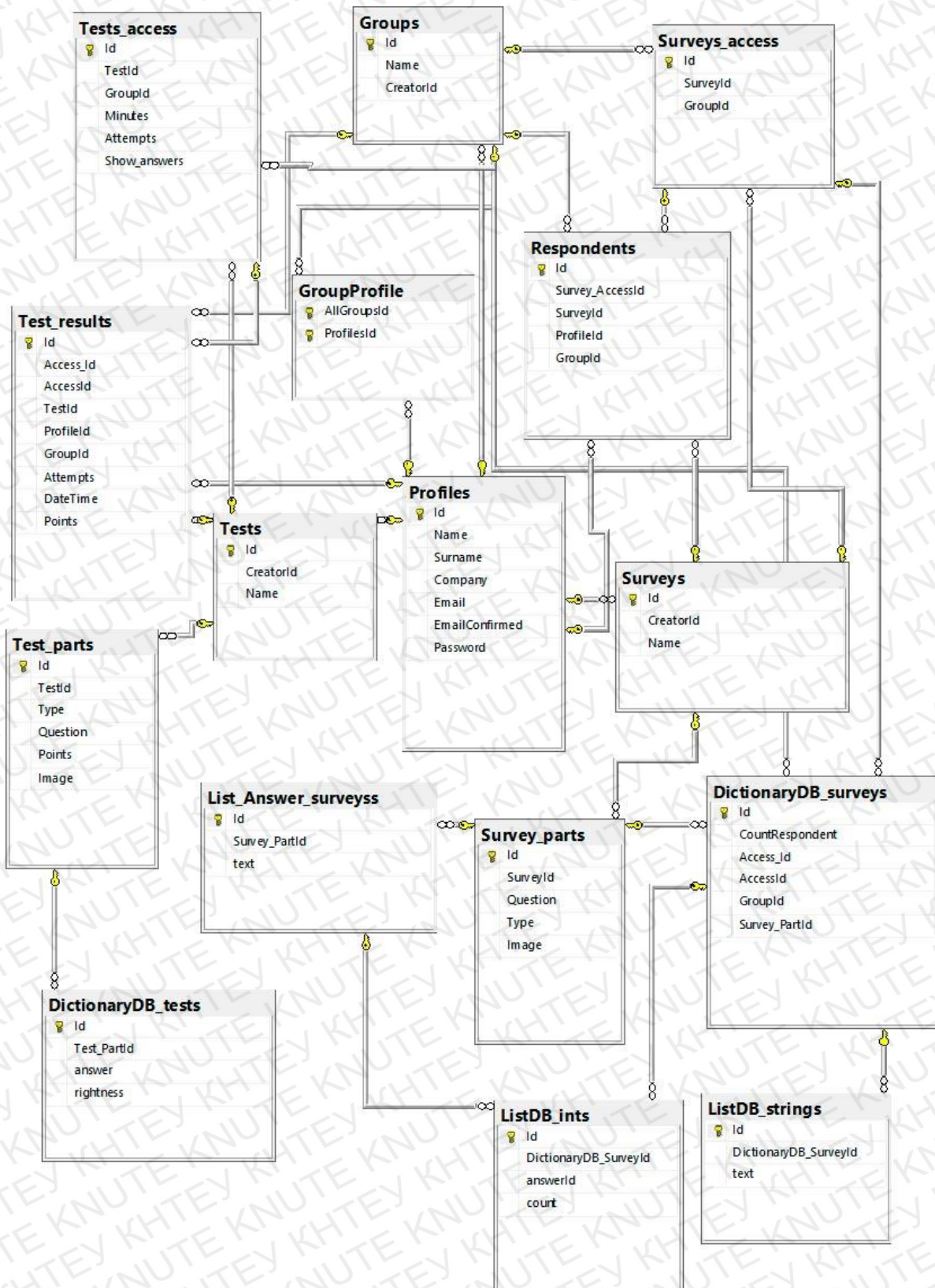


Рис. 2.2 Логічна модель бази даних

<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

Створення фізичної моделі бази даних – це етап, на якому на підставі логічної моделі бази даних створюється фізична структура бази даних, залежна від її реалізації. На цьому етапі виконується перетворення відношень логічної моделі реляційної бази даних у команди створення об'єктів фізичної бази даних, внаслідок чого створюється так звана внутрішня схема бази даних. Для проектування фізичної моделі необхідно вказати тип даних до кожного елемента таблиці. Згрупувавши всі ці дані можливо спроектувати фізичну модель (Рис. 2.3.).

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						16
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

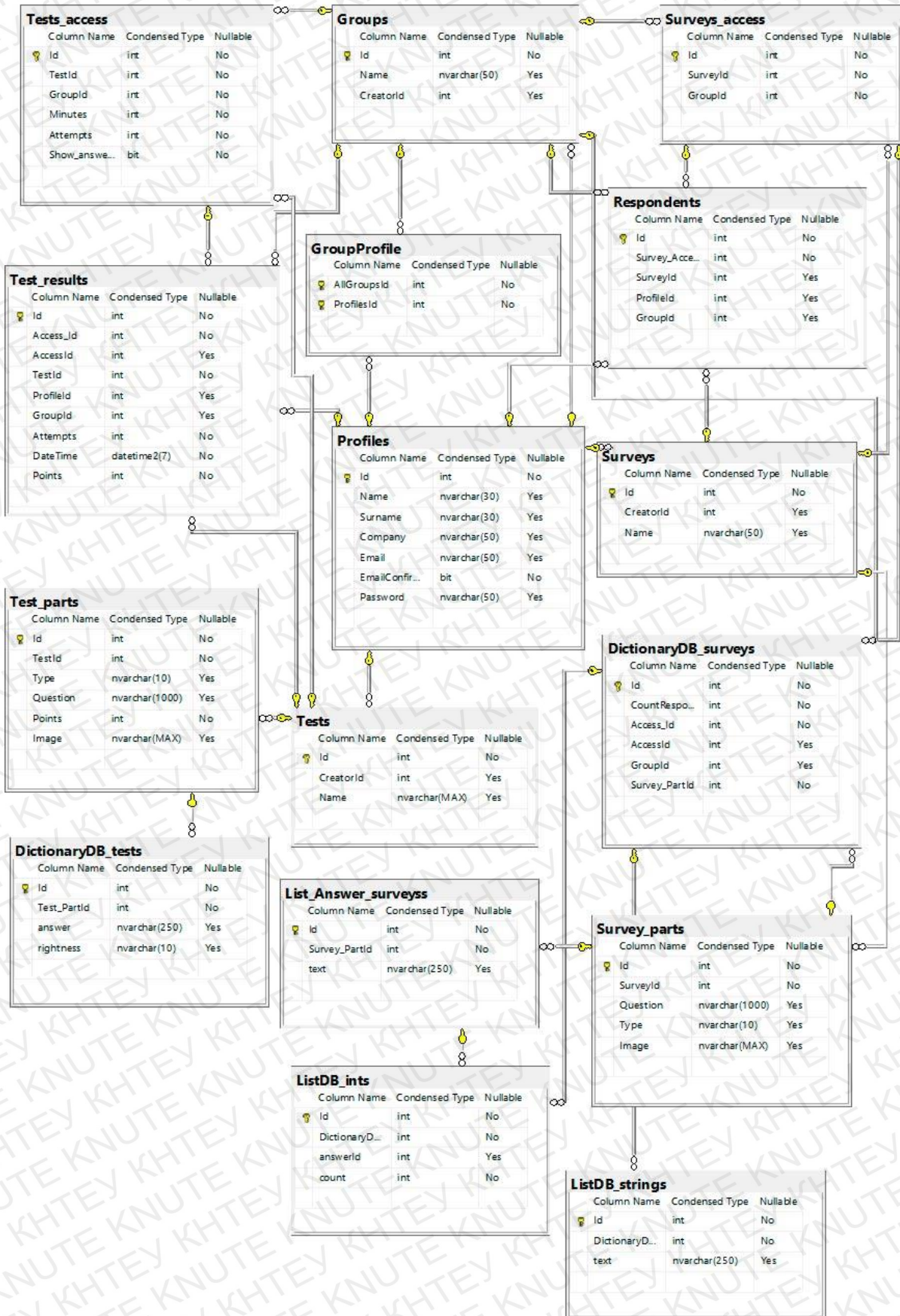


Рис. 2.3. Фізична модель бази даних

<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

2.2 ASP.NET Core 5 та Entity Framework Core 5

Платформа ASP.NET Core 5 являє собою технологію призначену для створення різного роду веб додатків. Весь її початковий код знаходиться у відкритому доступі на GitHub. ASP.NET Core може працювати поверх крос-платформного середовища .NET Core, яке може бути розгорнуте на основних популярних операційних системах: Windows, Mac OS, Linux. Додатковою перевагою є її модульність, що дозволяє окремо завантажувати всі необхідні частини веб-додатку через пакетний менеджер. Окрім цього дана платформа реалізує технологію Razor pages, що дозволяє створювати макети сторінок клієнтської частини комбінуючи мови C#, HTML, CSS та JS. Обробка запитів через конвеєр дозволяє впроваджувати залежності власних модулів або розширювати функціонал вже існуючих.

Entity Framework Core являє собою об'єктно-орієнтовану технологію доступу до даних. Вона є ORM-інструментом (object-relational mapping), тобто дозволяє працювати з базами даних на більш високому рівні абстракції. Центральною концепцією Entity Framework є поняття сутності. Сутність визначає набір даних та властивостей, які пов'язані з певним об'єктом, тому дана технологія передбачає роботу не з таблицями, а з об'єктами і їх колекціями. Також цей інструмент надає можливість оперувати об'єктами та моделями за допомогою запитів LINQ, які потім будуть транслюватися в SQL запити до бази даних. Усі сутності мають низку властивостей. Властивості можуть представляти не лише прості дані, такі як int або string, але і більш комплексні типи даних, на відміну від бази даних MS SQL.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						18
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2.3 Проектування архітектури системи

Розробка архітектури системи починається з її моделювання. Першим етапом моделювання є описання сценаріїв поведінки системи та її модулів при запитах до неї. Таке описання має назву Use Case. Розберемо основні сценарії, такі як реєстрація аккаунту, створення та проходження тесту та інші.

Use Case 1: Реєстрація

Характеристична інформація

Мета в контексті: Користувач надсилає запит на реєстрацію, очікує підтвердження.

Сфера застосування: Сервіс.

Рівень: Non-authorize.

Передумови: Користувач з даним Email не зареєстрований.

Успішна стадія завершення: Користувач має зареєстрований та підтверджений аккаунт. Дані аккаунту збережені в БД.

Невдале завершення стану: Запит на реєстрацію був відхилений, користувач дізнається причину відмови.

Основний актор: Користувач.

Тригер: Надходить запит на реєстрацію аккаунту.

Генеральний сценарій успіху

1. Користувач заповнює та відправляє форму для реєстрації
2. Система перевіряє дані та створює аккаунт.
3. Надсилається лист на Email користувача для його підтвердження.
4. Користувач підтверджує Email перейшовши за посиланням.
5. Система відкриває доступ для аккаунту.

Розширення

2а. Відмова в реєстрації:

2а1. Перевірити коректність заповнених даних.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						19
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Відкрити питання

Що станеться, якщо користувач не отримає листа на Email?

Що станеться, якщо даний Email вже зареєстрований?

Use Case 2: Створення тесту

Характеристична інформація

Мета в контексті: Користувач створює тест, очікує збереження в БД.

Сфера застосування: Сервіс.

Рівень: Authorize.

Передумови: Користувач авторизований.

Успішна стадія завершення: Користувач створив та заповнив тест. Дані тесту/опитування оброблені та збережені в БД.

Невдале завершення стану: Валідація тесту проходить невдало.

Користувач дізнається причину відмови в збереженні.

Основний актор: Користувач.

Тригер: Надходить запит на створення тесту.

Генеральний сценарій успіху

1. Користувач заповнює форму на створення тесту.
2. Система створює тест.
3. Користувач заповнює тест даними.
4. Система зберігає тест в БД.

Розширення

4а. Помилка при валідації:

4а1. Перевірити коректність заповнених даних.

Use Case: 3 Створення групи

Характеристична інформація

Мета в контексті: Користувач створює групу, очікує збереження в БД.

Сфера застосування: Сервіс

Рівень: Authorize

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						20
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Передумови: Корисувач авторизований.

Успішна стадія завершення: Користувач створив групу та додав необхідні йому аккаунти до цієї групи. Група збережена в БД.

Невдале завершення стану: Валідація даних проходить невдало. Користувач дізнається причину відмови в збереженні групи.

Основний актор: Користувач.

Тригер: Надходить запит на створення групи.

Генеральний сценарій успіху

1. Користувач заповнює форму на створення групи.
2. Система створює групу.
3. Користувач додає до групи необхідні йому аккаунти.
4. Система зберігає групу в БД.

Розширення

4а. Помилка при валідації:

4а1. Перевірити коректність заповнених даних.

Відкрити питання

Що станеться, якщо доданий аккаунт не підтверджений?

Use Case: 4 Створення доступу до тесту

Характеристична інформація

Мета в контексті: Користувач відкриває доступ до тесту для певної групи на певних умовах, очікує позитивну відповідь від системи.

Сфера застосування: Сервіс

Рівень: Authorize

Передумови: Корисувач авторизований, створена група та тест.

Успішна стадія завершення: Користувач відкрив доступ до тесту/опитування. Учасники групи отримують доступ.

Невдале завершення стану: Валідація даних проходить невдало. Користувач дізнається причину відмови системи.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						21
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Основний актор: Користувач.

Тригер: Надходить запит на відкриття доступу до тесту.

Генеральний сценарій успіху

1. Користувач створює доступ до тесту на певних умовах для певної групи.
2. Система оброблює запит та відкриває доступ.

Розширення

2а. Помилка при валідації:

2а1. Перевірити коректність заповнених даних.

Use Case 5: Проходження тесту

Характеристична інформація

Мета в контексті: Користувач відправляє запит на проходження тесту за доступом.

Сфера застосування: Сервіс.

Рівень: Authorize.

Передумови: Користувач авторизований та має доступ до тесту.

Успішна стадія завершення: Користувач пройшов тест, результат збережено в БД.

Невдале завершення стану: Валідація даних проходить невдало. Користувач дізнається причину відмови в обробці тесту.

Основний актор: Користувач.

Тригер: Надходить запит на проходження тесту.

Генеральний сценарій успіху

1. Користувач відправляє запит на проходження тесту.
2. Система надає тест за доступом.
3. Користувач проходить тест.
4. Система оброблює результати тесту та зберігає їх.

Розширення

4а. Помилка при валідації:

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						22
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

4a1. Перевірити коректність заповнених даних.

Відкрити питання

Що станеться, якщо доданий користувач спробує порушити умови доступу?

Use Case: 6 Перегляд результатів тестування

Характеристична інформація

Мета в контексті: Користувач відправляє запит на отримання результатів за тестом. Система надає дані.

Сфера застосування: Сервіс

Рівень: Authorize

Передумови: Користувач авторизований, результати за тестом існують.

Успішна стадія завершення: Користувач отримав дані за запитом.

Невдале завершення стану: Результати за тестом не знайдені.

Основний актор: Користувач.

Тригер: Надходить запит на отримання результатів за тестом.

Генеральний сценарій успіху

1. Користувач відправляє запит на отримання результатів за тестом/опитуванням.

2. Система оброблює запит та надає дані.

Відкрити питання

Що станеться, якщо результатів за тестом/опитуванням немає?

Ще одним із видів моделювання системи є розробка UML діаграм. Існує достатня кількість їх видів для того, щоб відобразити всі аспекти взаємодії із системою та між її частинами. Одним із них є діаграма послідовності. Вона відображає взаємодію задіяних об'єктів впорядкованих за часом та послідовність надісланих повідомлень. Відповідно до етапів взаємодії з системою, першою розглянемо реєстрацію (Рис. 2.4.).

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						23
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

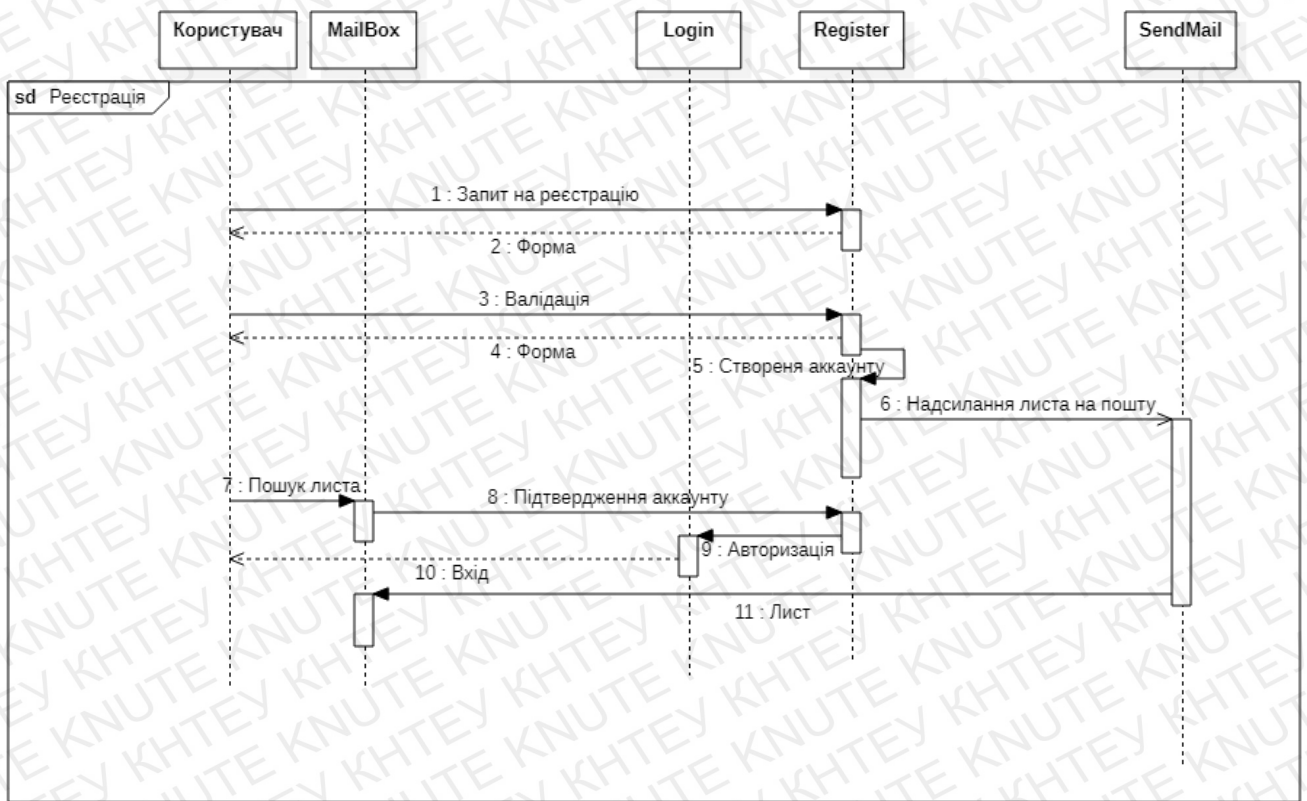


Рис. 2.4. Діаграма послідовності реєстрації

Продовжимо моделювати сценарій створення тестування. Основними об'єктами взаємодії є користувач та класи, що відповідають за створення, редагування та відображення тесту. Змоделювавши їх взаємодію відповідно до вимог технічного завдання - отримаємо діаграму послідовності створення тестування (Рис. 2.5.). Після створення тестування та відкриття доступу до нього – потрібно змоделювати діаграму послідовності проходження тесту (Рис. 2.6.).

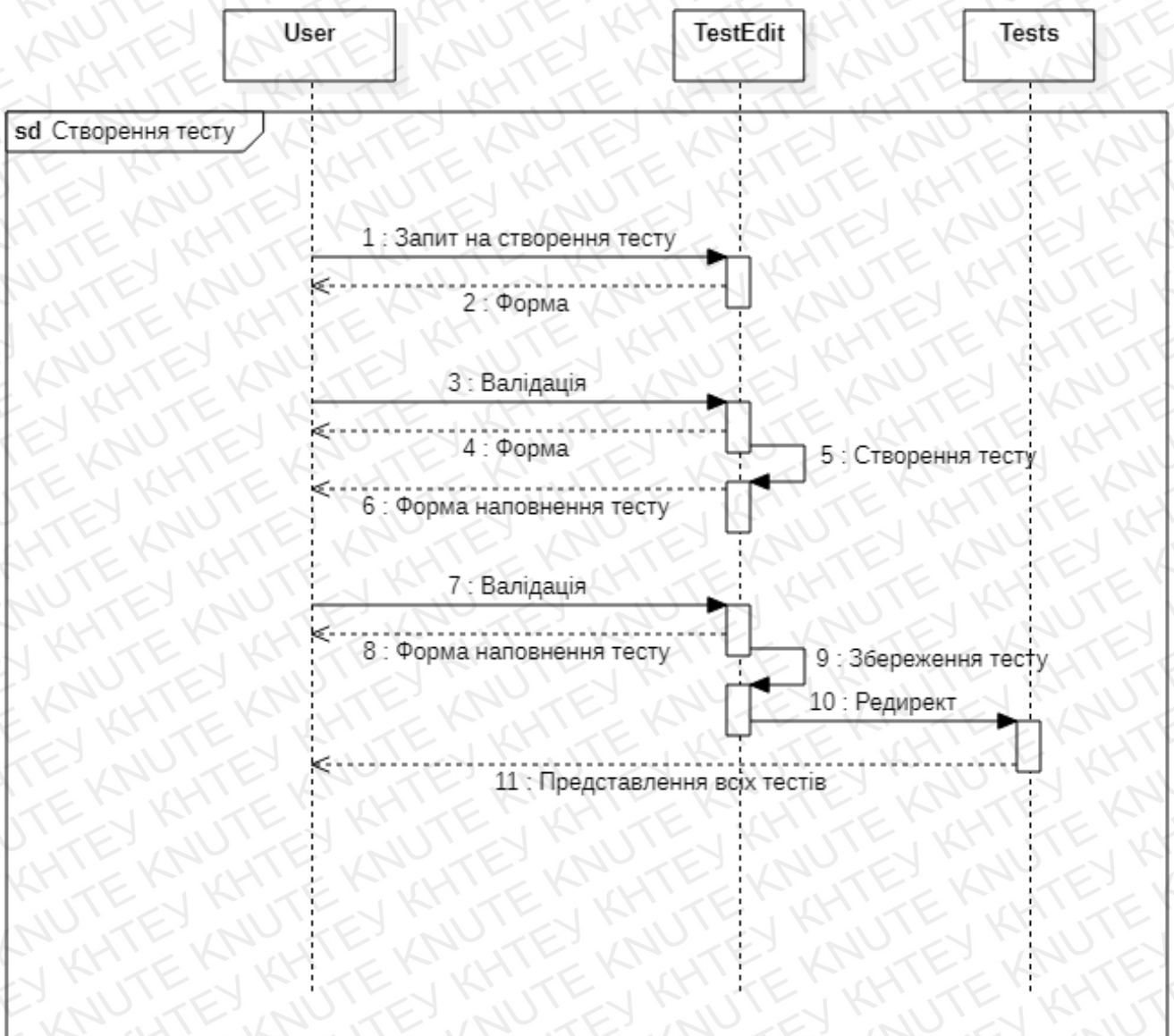


Рис. 2.5. Діаграма послідовності створення тесту

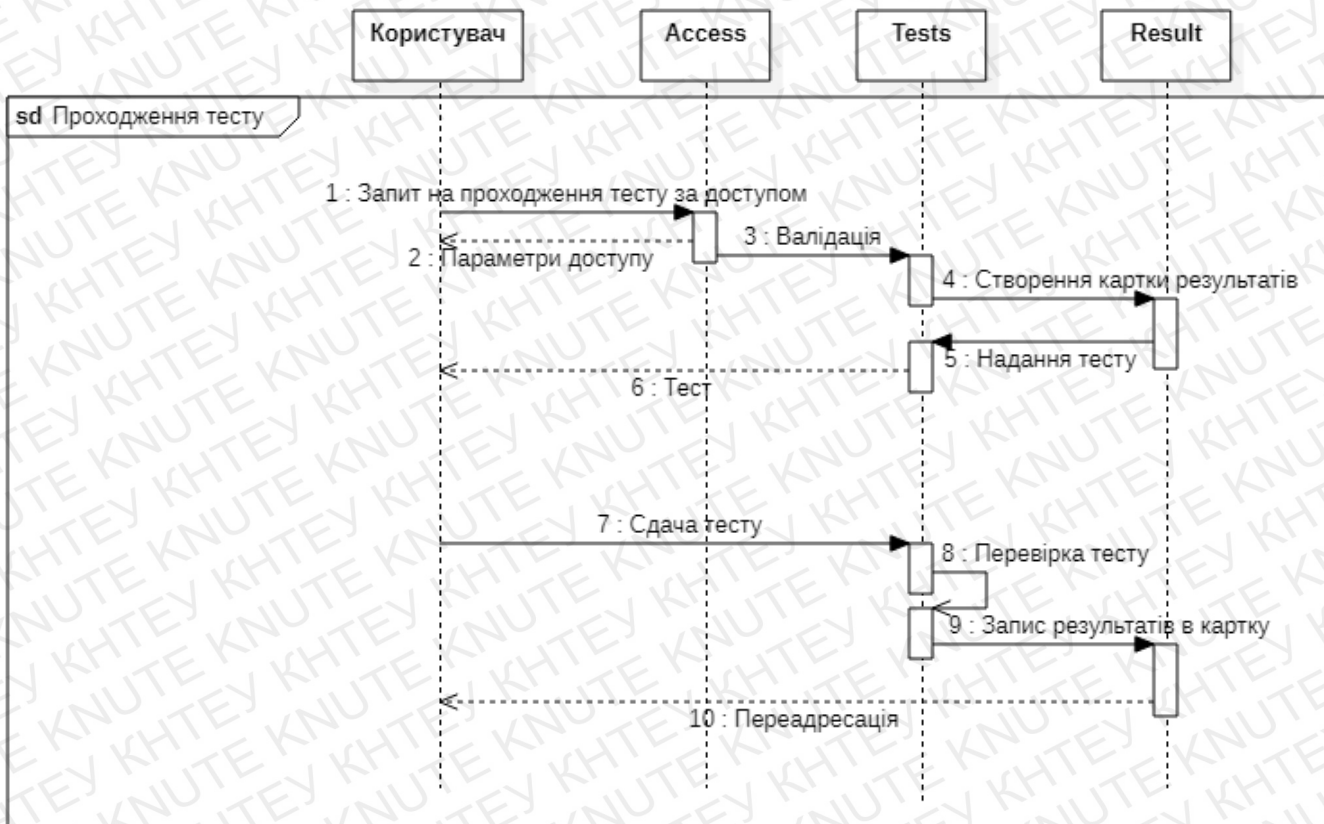


Рис. 2.6. Діаграма послідовності проходження тесту

Наступним етапом моделювання підсистеми тестування буде створення діаграми діяльності (Рис. 2.7.). Діаграма діяльності відображає послідовність кроків в потоці управління системи.

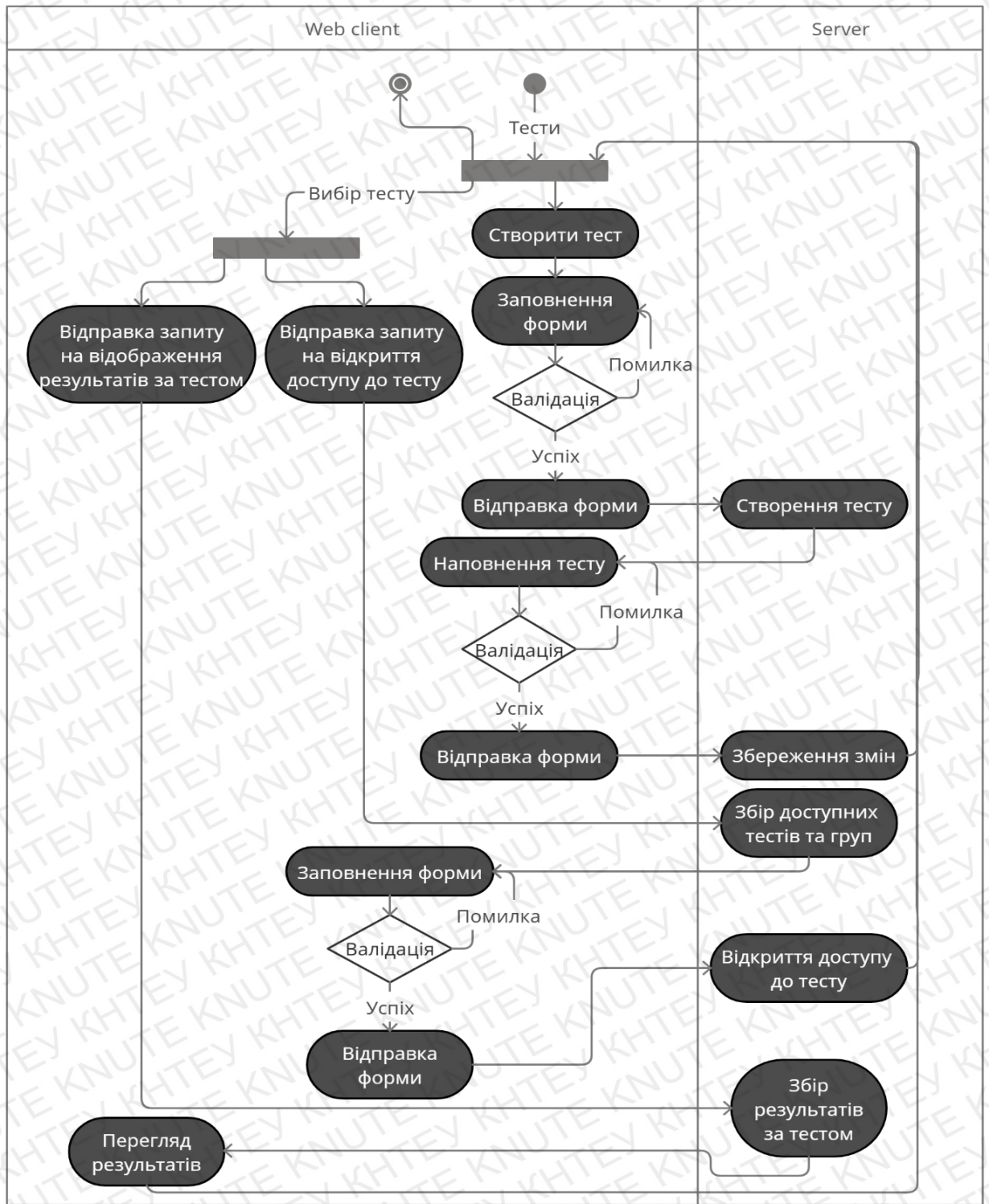


Рис. 2.7. Діаграма діяльності підсистеми тестування

На останньому етапі моделювання підсистеми тестування потрібно відобразити також діаграму станів (Рис. 2.8.).

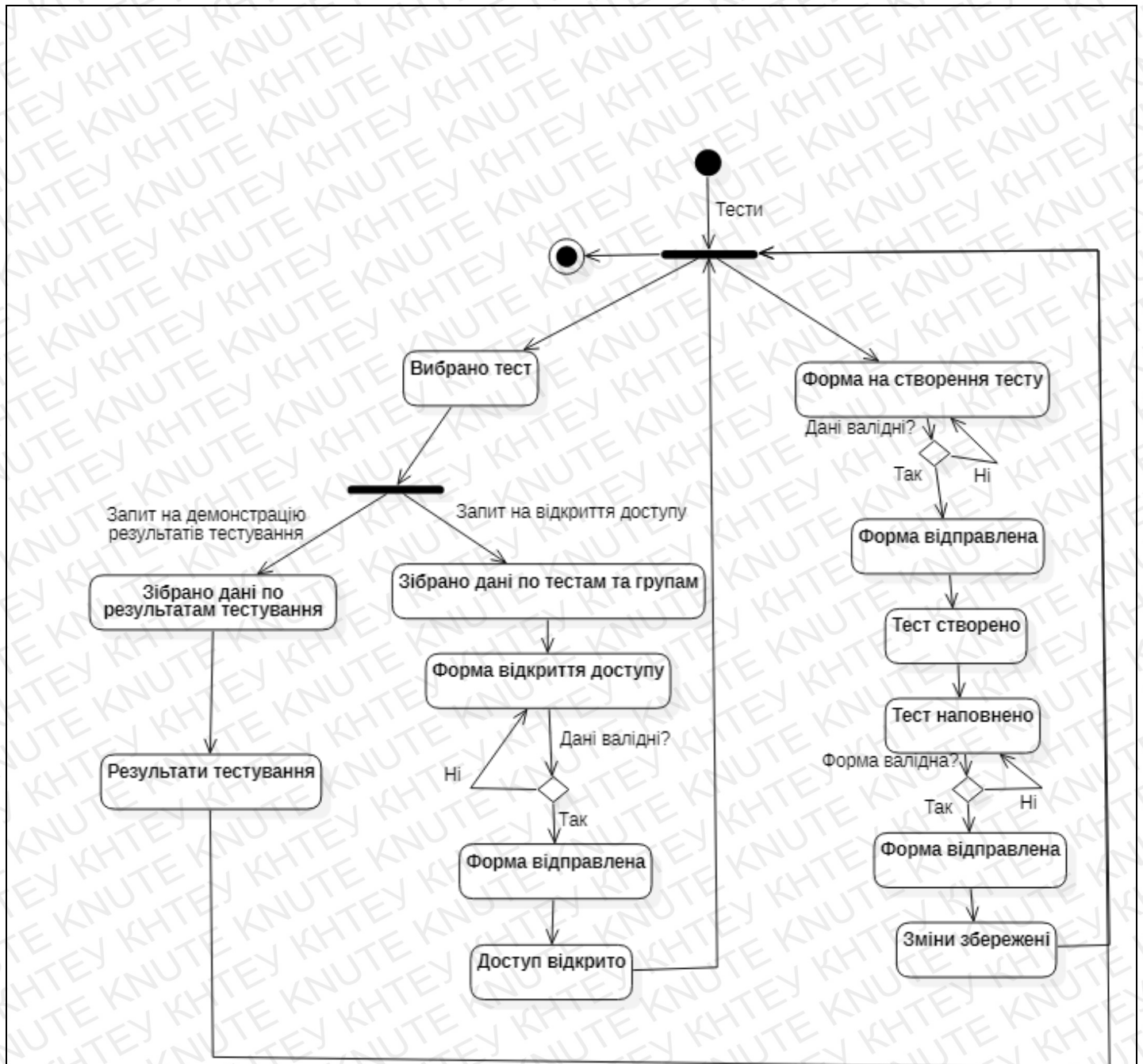


Рис. 2.8. Діаграма станів підсистеми тестування

2.4 Висновки до розділу 2

У цьому розділі спроектовано та розроблено концептуальну, логічну, фізичну моделі бази даних, відповідно до яких створена база даних,

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						28
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

що складається з 16 таблиць. Вона реалізована за допомогою MS SQL. В якості платформи для створення проекту виступає ASP.NET Core 5. Для взаємодії з базами даних використовується технологія Entity Framework Core 5. Після розбору переваг обраної платформи для системи та технологій що будуть використовуватися разом із нею була змодельована архітектура системи та частково представлена у вигляді описання Use Case, діаграм послідовності, діяльності та станів.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		29

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Розробка функціоналу

Наступним етапом після розробки архітектури програмного забезпечення є написання коду та тестування модулів системи. Відповідно до вимог технічного завдання доступ до підсистем тестування та опитування надається тільки після авторизації, тому перш за все потрібно розробити підсистему авторизації, реєстрації з підтвердженням електронної пошти та редагуванням даних профілю. Згідно шаблону проектування Model View Controller всі операції відбуватимуться в класах контролерів. Взаємодію цих об'єктів можна відобразити за допомогою діаграми класів (Рис. 3.1.).

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		РЗ	30	43
Зав. кафедри		Криворучко О.В.		10.04.2021	<i>Розробка програмного забезпечення</i>	<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		
Керівник		Котенко Н.О.		10.04.2021				
Гарант		Цензура М.О.		10.04.2021				
Розроб.		Юрченко С.С.		10.04.2021				

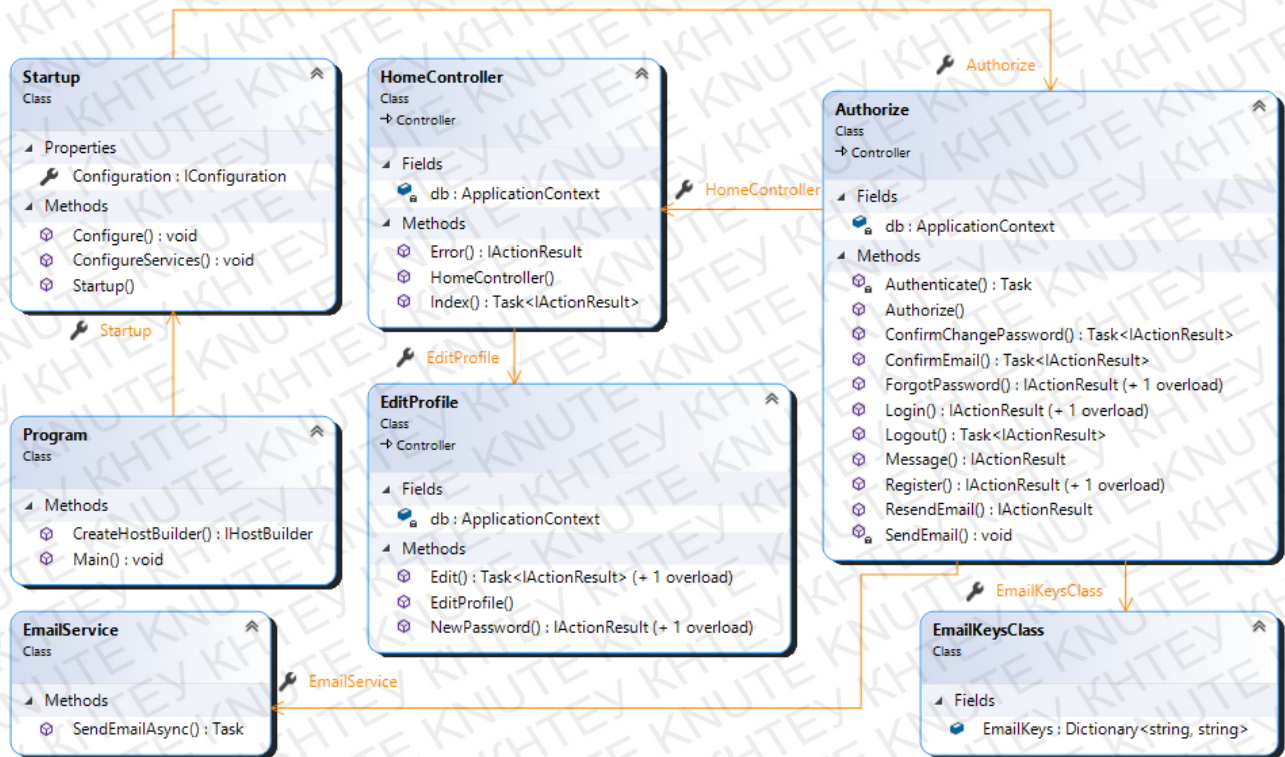


Рис. 3.1. Діаграма класів контролерів авторизації

Діаграма класів контролерів авторизації також включає в себе класи обробки запитів до системи. Аутентифікація відбувається через зашифровані соокіє які відправляються на сторону клієнта. При кожному запиті до серверної частини – вони валідуються та розшифровуються. На основі отриманих даних відбувається авторизація користувача. Якщо користувач підтвердив свою електронну пошту після реєстрації і аутентифікація пройшла успішно – він отримує доступ до підсистем тестування та опитування.

Підсистема тестування, як і опитування, включає в себе засоби для створення та редагування груп користувачів, які необхідні для реалізації функцій підсистем доступу до тестувань та опитувань. Контролери та їх функції можна відобразити за допомогою діаграми класів (Рис. 3.2.).

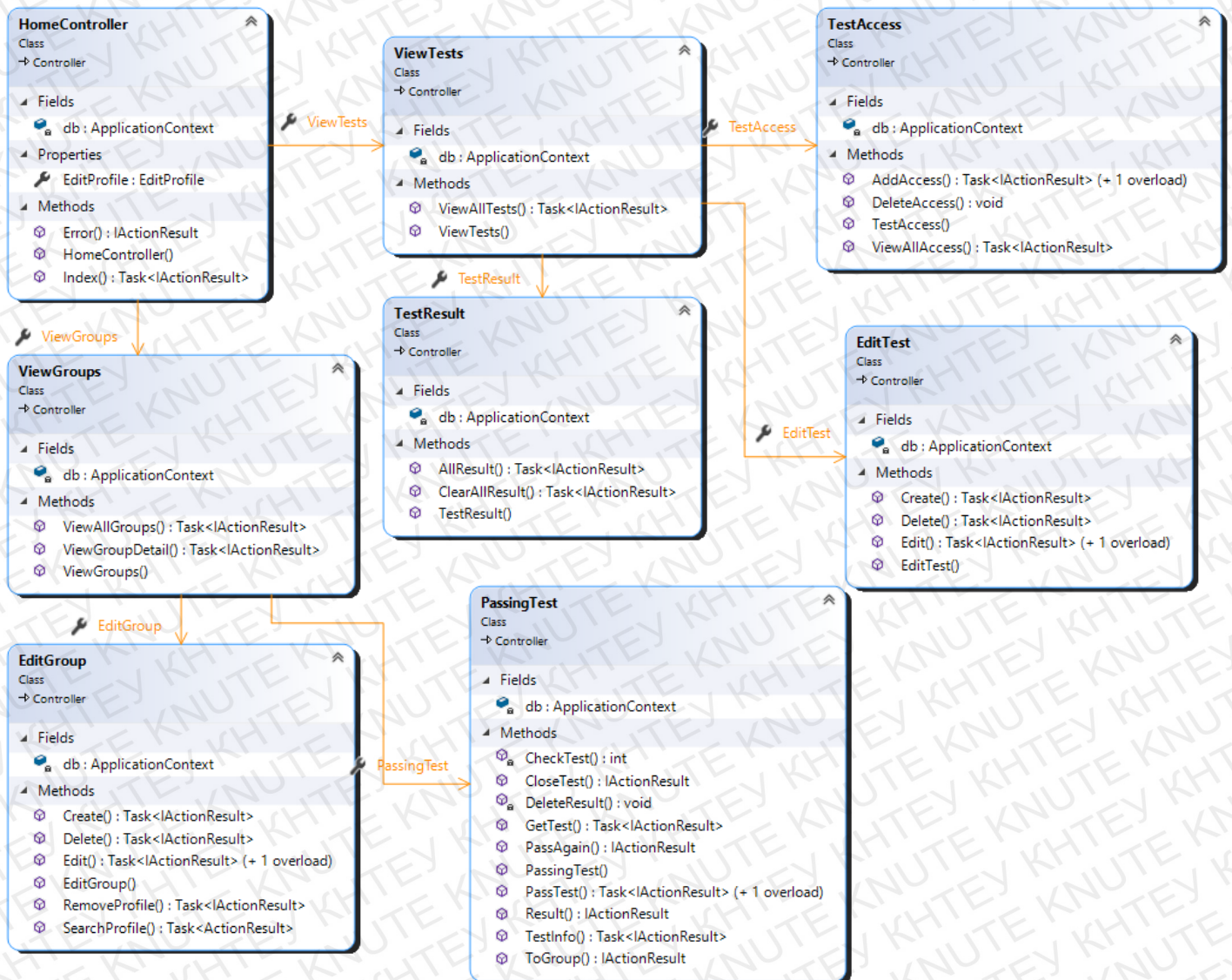


Рис. 3.2. Діаграма класів контролерів тестування

3.2 Опис інтерфейсу

Після авторизації користувач потрапляє до головного меню (Рис. 3.3.). Головним об'єктом є куб, що обертається, на гранях якого відображається інформація відповідно до пункту меню що обрав користувач.

Меню складається з 6 розділів, а саме: профіль, групи, тести, опитування, міні гра та детальна інформація.

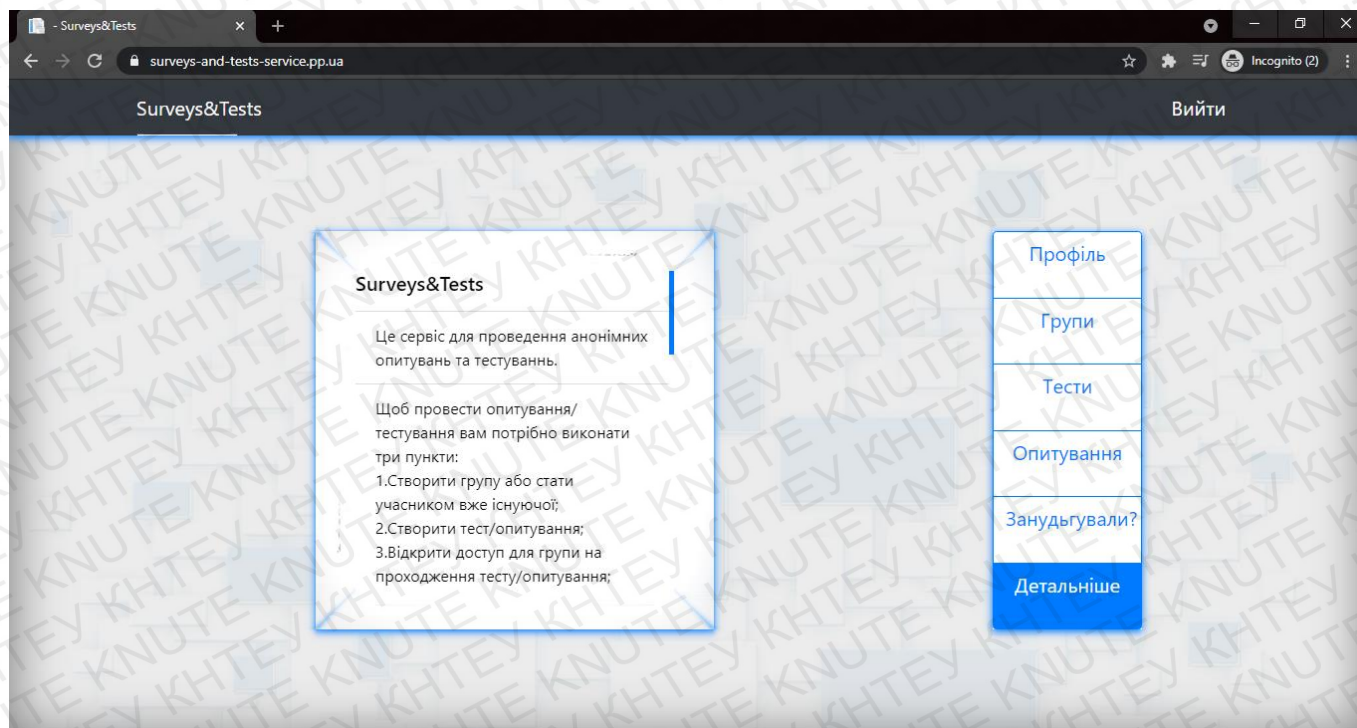


Рис. 3.3. Головне меню

В розділі «Профіль» знаходиться інформація про користувача та засоби її редагування. Інформація включає в себе лише основні дані користувача, такі як: ім'я, прізвище, адреса електронної пошти та ідентифікатор. Поле «Ідентифікатор» слугує додатковим засобом полегшення пошуку користувача в системі при створенні груп. Тут рекомендується вказувати такі дані, як наприклад – спеціальність, назву відділу, номер групи... або все разом.

В розділі «Групи» можна створити нову групу або ж перейти до вже існуючих, в складі яких знаходиться користувач. Потрапляючи в підрозділ «Всі групи» (Рис. 3.4.) користувач бачить список груп та кількість відкритих для нього тестів та опитувань в рамках кожної групи. Для полегшення пошуку список можна сортувати за кожним полем та фільтрувати за назвою і критерієм володіння групою. Публікувати тести та опитування в групі може кожен її учасник. Додавати учасників до групи може тільки її творець.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						33
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Пошук користувачів може відбуватись за ім'ям, прізвищем, ідентифікаторами або за будь-якою комбінацією цих даних (Рис. 3.5).

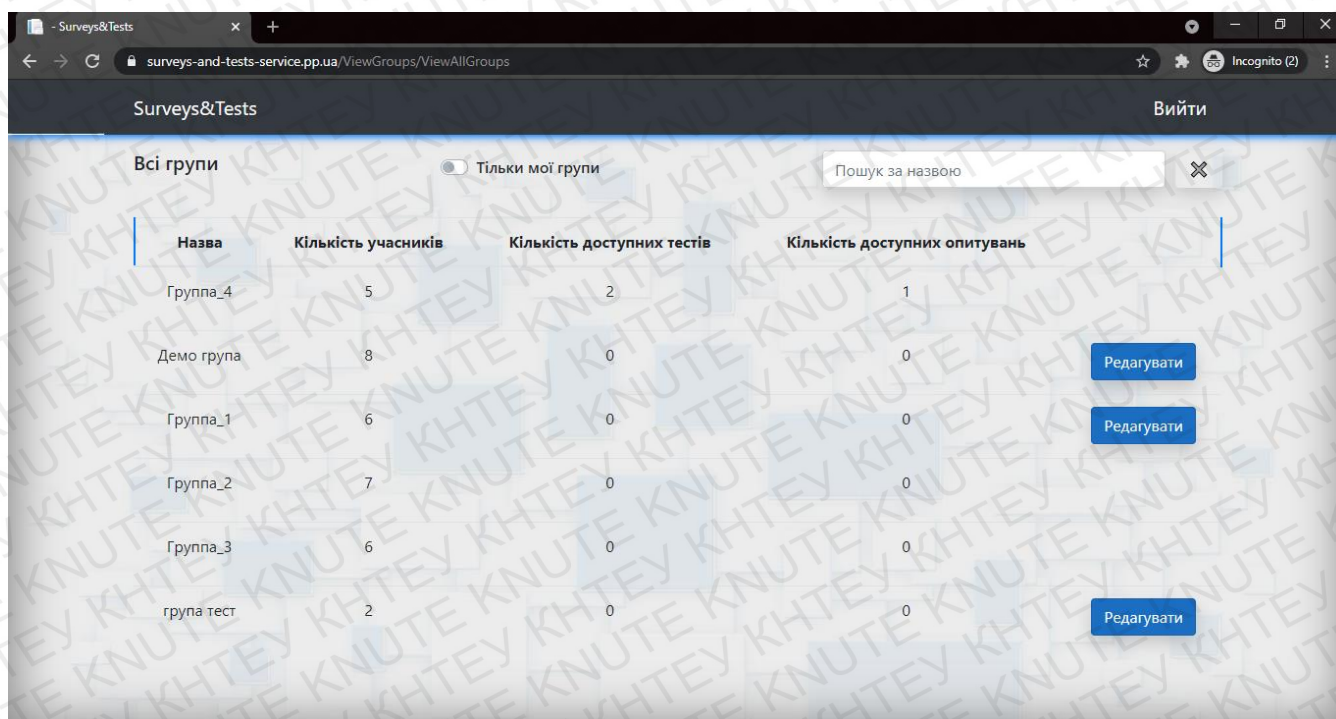


Рис. 3.4. Всі групи

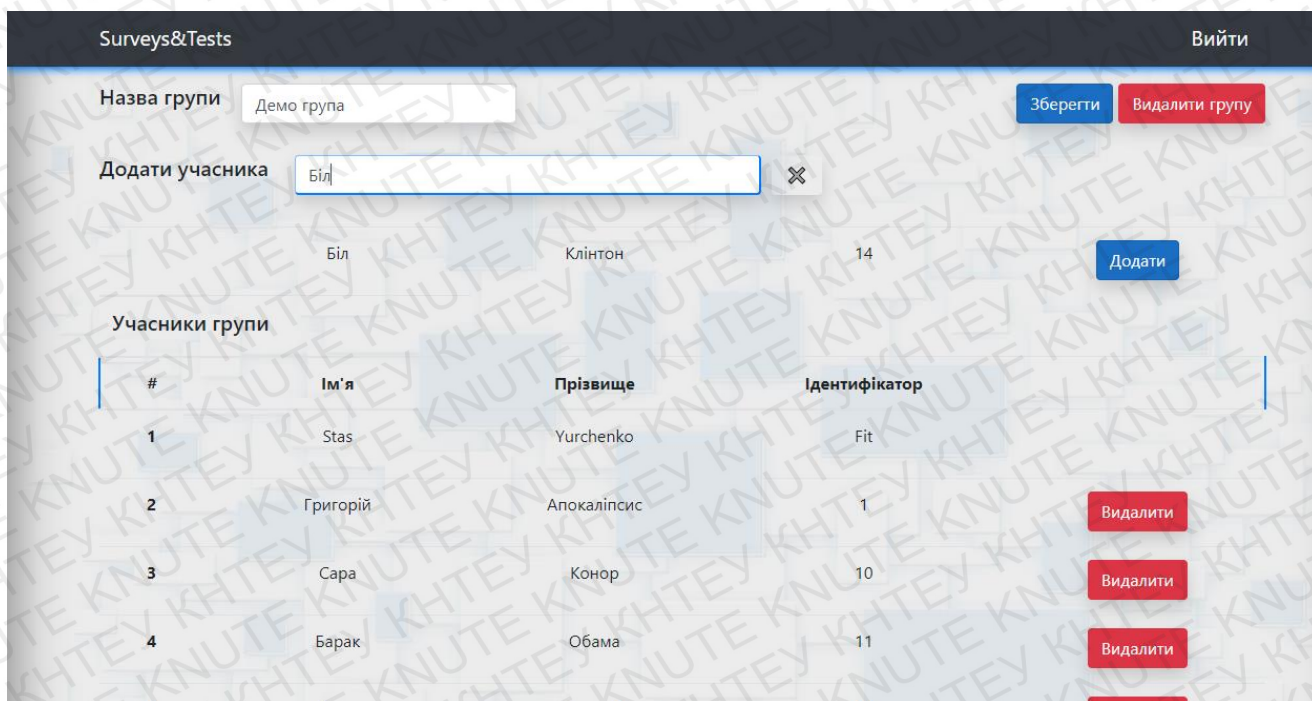


Рис. 3.5. Редагування групи

В розділі «Тести» користувач може перейти до створення нового тесту або ж до управління вже існуючими (Рис. 3.6.). При створенні тесту, користувач заповнює картки питань та вказує правильні відповіді. В підрозділі «Мої тести» (Рис. 3.7.) відображається список всіх створених їм тестувань та кількість відкритих до них доступів. Також можна перейти до редагування або перегляду результатів будь-якого тесту. Перейшовши до підрозділу доступів (Рис. 3.8.) користувач може спостерігати за всіма відкритими доступами до тестувань та їх налаштуваннями, відкрити новий доступ або закрити вже існуючий. Як і в попередньому підрозділі тут є засоби сортування та фільтрування записів для полегшеного пошуку потрібної інформації. Перегляд результатів тестування також передбачає вибірку за фільтрами доступу, груп, учасниками, часом тестування і т.д. (Рис. 3.9.).

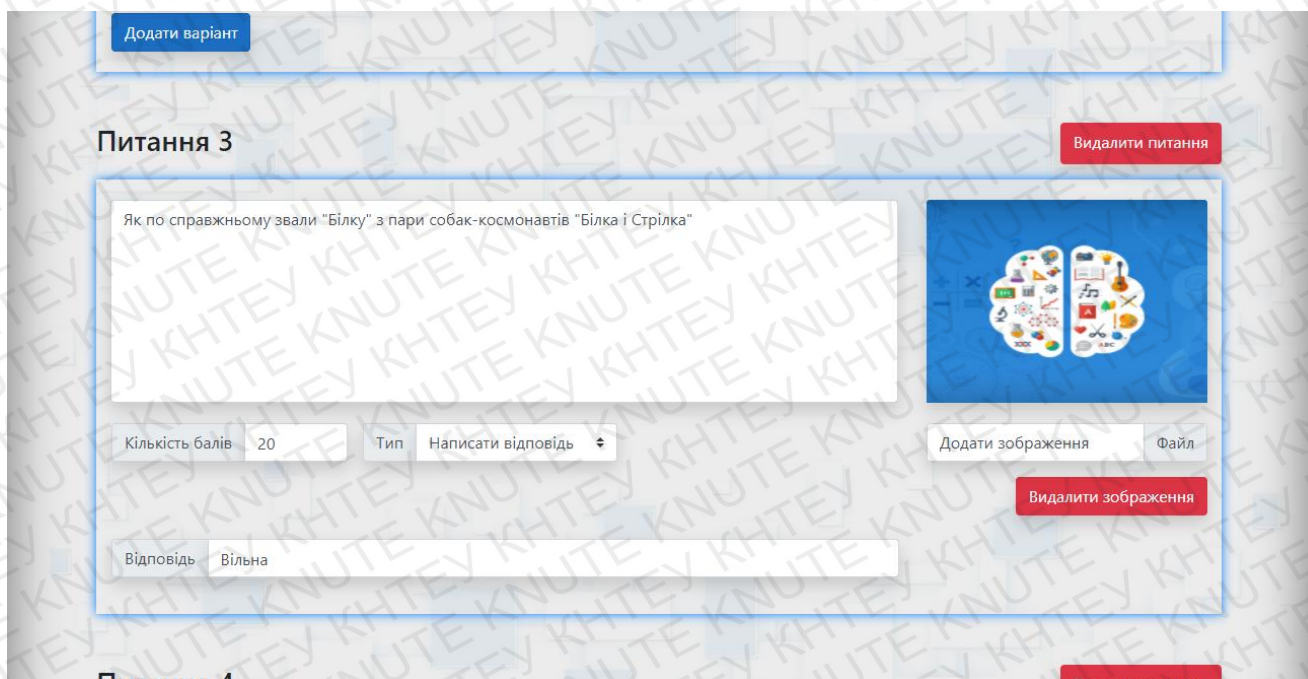


Рис. 3.6. Редагування тесту

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						35
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

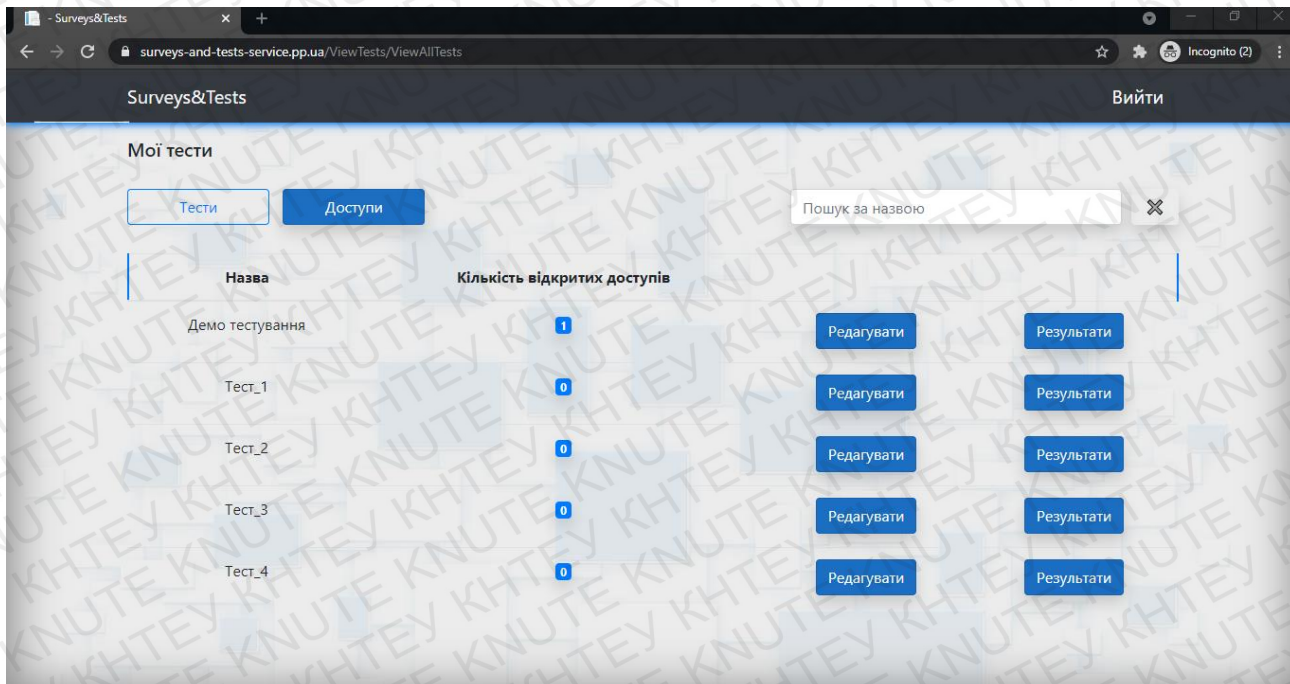


Рис. 3.7. Мои тесты

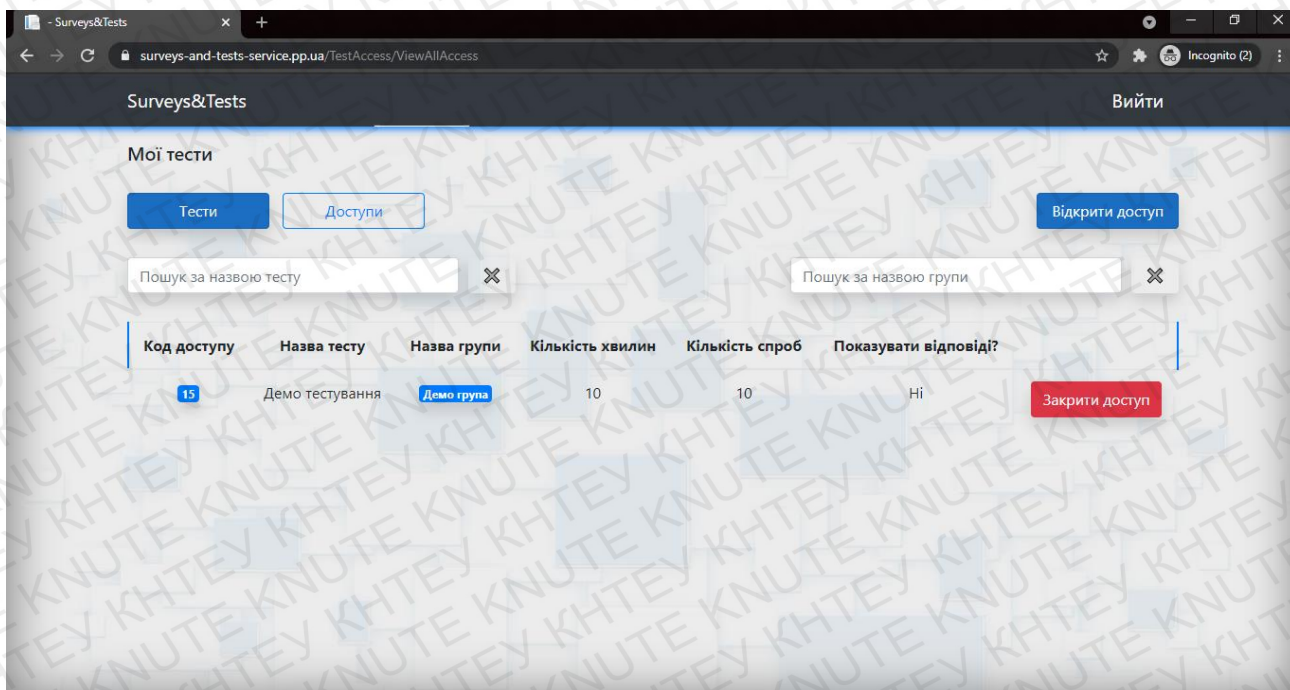


Рис. 3.8. Доступы до тестуван

Surveys&Tests Вийти

Результати

Демо тестування Видалити всі результати

Пошук за профілем

Код доступу	Назва групи	Профіль	Спроба	Час здачі	Кількість балів
6	Демо група	Григорій Апокаліпсис	3	Не зданий	Не зданий
6	Демо група	Григорій Апокаліпсис	2	4/1/2021 1:20:46 AM	65
6	Демо група	Григорій Апокаліпсис	1	4/1/2021 1:19:33 AM	55
6	Демо група	Кейт Комлев	1	4/1/2021 1:43:15 AM	45
6	Демо група	Сара Конор	1	4/1/2021 1:33:01 AM	25
6	Демо група	Біл Квінто	1	4/1/2021 1:41:40 AM	15

Рис. 3.9. Результати тестування

По завершенню тестування учасник отримує інформацію стосовно його результатів, та має змогу переглянути правильні відповіді якщо це передбачено налаштуваннями доступу. Помилки в тесті помічаються червоним кольором (Рис. 3.10.). Бали за питання нараховуються тільки за відсутності помилок.

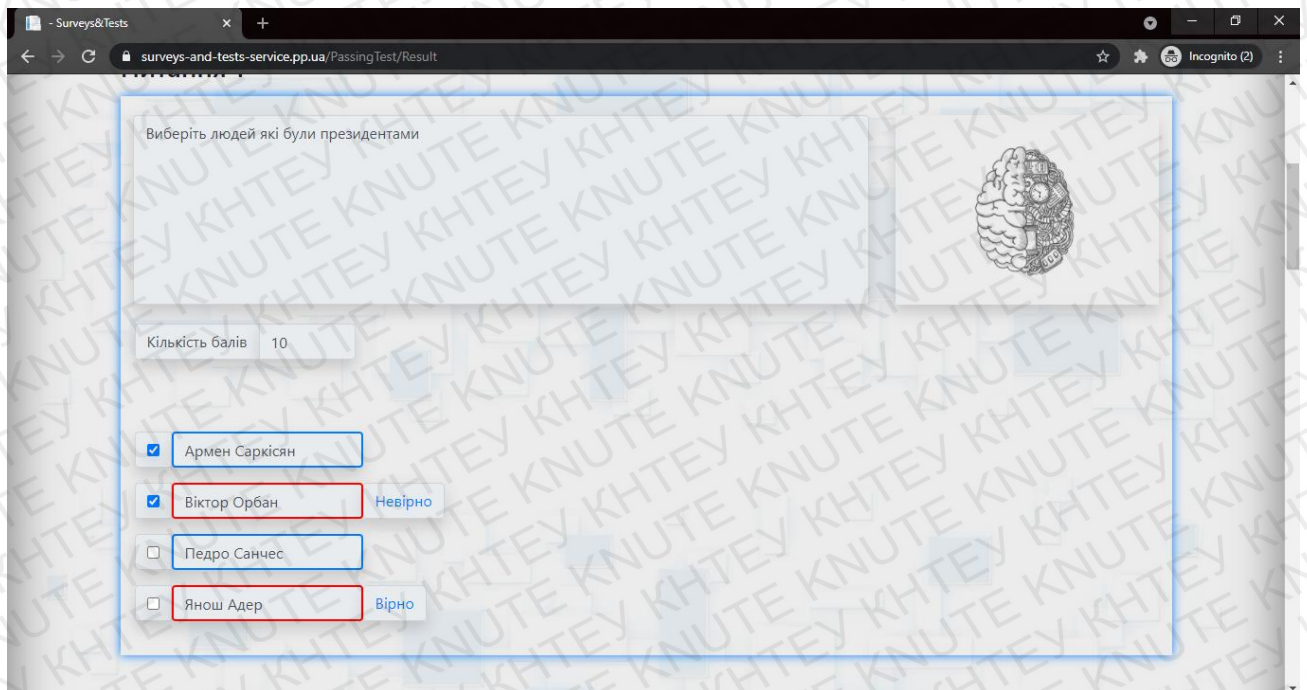


Рис. 3.10. Помилки в тестуванні

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						37
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

В розділі «Опитування» знаходяться підрозділи меню для створення нового опитування (Рис. 3.11.) або ж управлінням вже існуючими. Як і в підрозділах меню тестувань, тут є список всіх опитувань, меню доступів до них, засоби по редагуванню опитувань та переглядом результатів за ними. Відповідно до технічного завдання результати за опитуванням зберігаються і подаються у вигляді статистики (Рис. 3.12.). Також під кожним питанням учасники можуть залишати коментарі, які будуть анонімними та подаватись у вигляді списку.

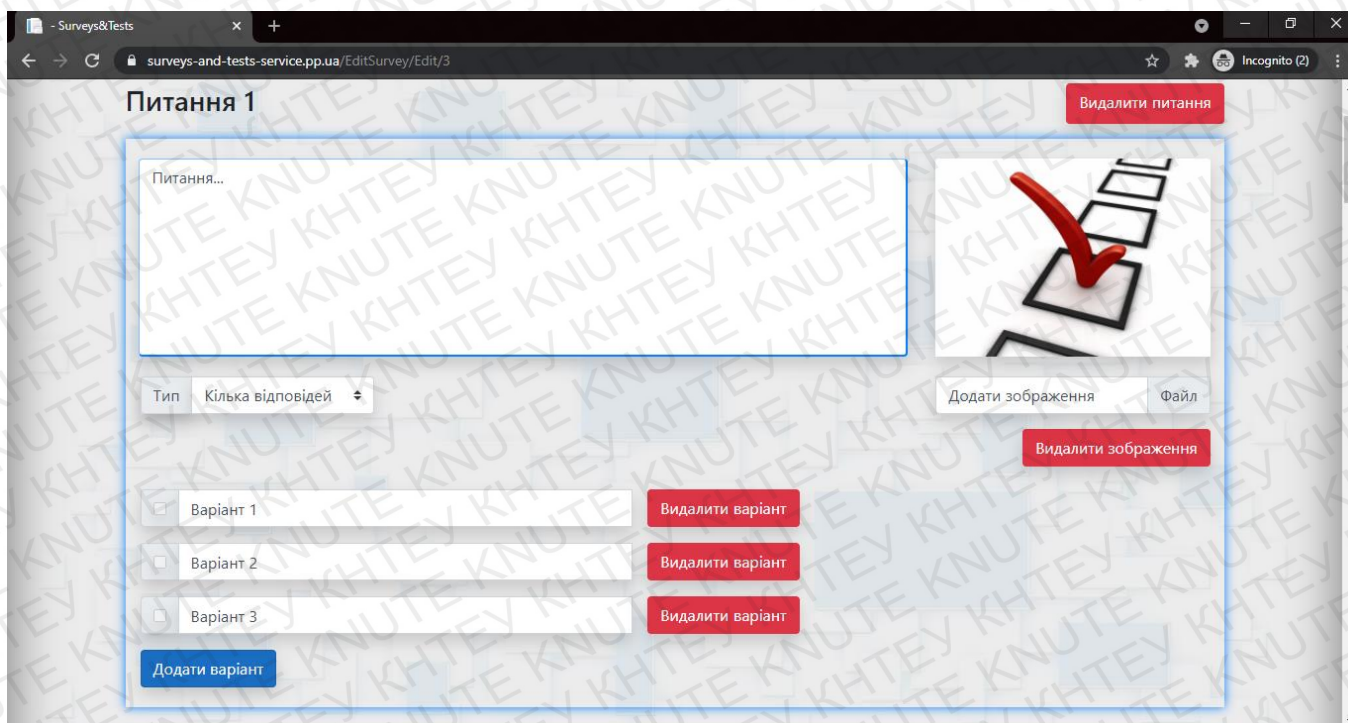


Рис. 3.11. Створення опитування

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						38
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Питання 1

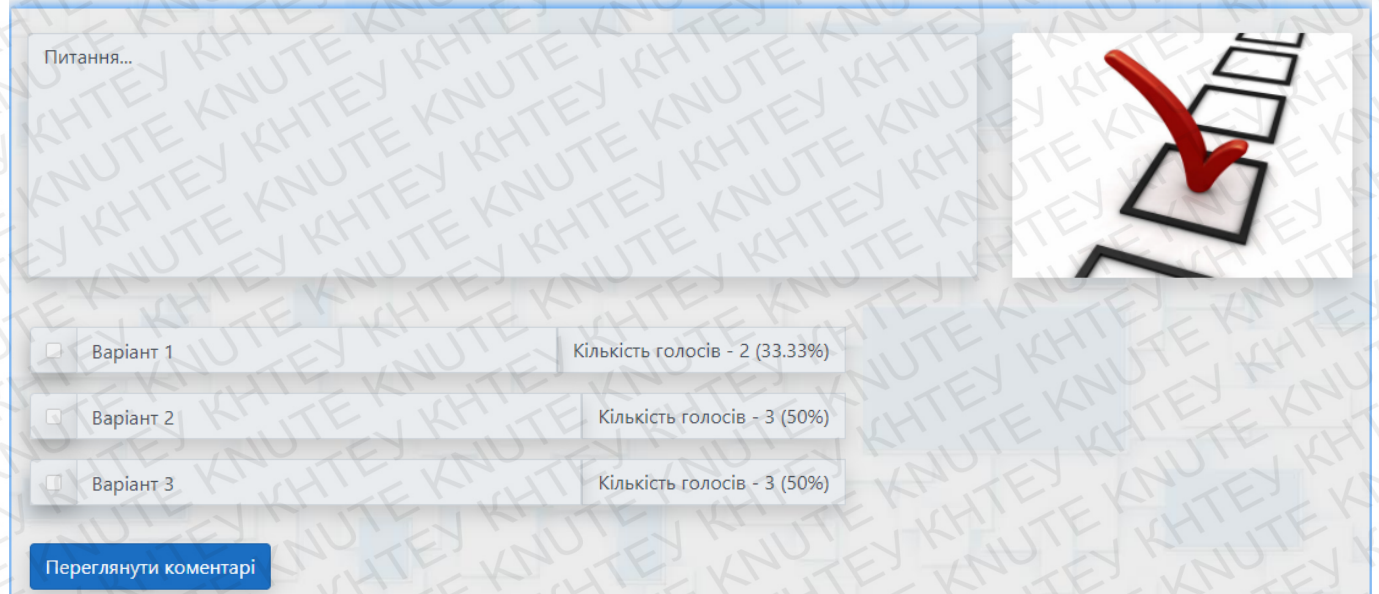


Рис. 3.12. Результати опитування

3.3 Висновки до розділу 3

У цьому розділі розглянуто створення функціоналу системи відповідно до розробленої архітектури та опис інтерфейсу. Підсистеми тестування та опитування працюють разом із авторизацією і відповідають вимогам технічного завдання. Засоби захисту від спроб втручання в роботу алгоритму успішно пройшли тестування. Інтерфейс виконаний у мінімалістичному дизайні, що позитивно вплинуло на фактор зручності у користуванні.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						39
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У цій роботі було спроектовано та розроблено веб-додаток для проведення тестувань та опитувань. Після розглядання проблематики завдання, було розроблено технічне завдання, вимоги якого були повністю виконані.

Проектування та розробка бази даних стала одним із перших етапів створення проекту. Після аналізу переваг і недоліків обраної платформи та технологій було розпочато моделювання архітектури додатку.

Відповідно до вимог технічного завдання та основної проблематики обраної теми – перш за все було спроектовано та розроблено системи безпеки та реалізації доступів. Успішне проходження тестувань на кожному етапі сприяло більш швидкій розробці основного функціоналу додатку.

Веб-додаток має досить простий принцип роботи, який вирішує проблеми контролю та доступу. Для проведення тестування або опитування необхідно лише три речі:

1. Створити групу цільових осіб або стати учасником вже існуючої;
2. Створити сам тест або опитування;
3. Відкрити доступ для групи на певних умовах;

Підсистема тестування має достатні засоби захисту від спроб взлому алгоритму або несанкціонованого доступу до правильних відповідей. Підсистема опитування вирішує проблематику анонімності в рамках збереження контролю за доступом. Клієнтська частина має простий та інтерактивний інтерфейс що позитивно вплинуло на фактор зручності користування.

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Висновки та пропозиції</i>	ВП	40	43
Зав. кафедри		Криворучко О.В.		24.04.2021				
Керівник		Котенко Н.О.		24.04.2021				
Гарант		Цензура М.О.		24.04.2021				
Розроб.		Юрченко С.С.		24.04.2021				
					<i>Факультет інформаційних технологій, 4 курс, 7 група</i>			

В якості розвитку веб-додатку в майбутньому, можна додати модуль оповіщення через електронну пошту для користувачів про появу нових доступів до тестів або опитувань. Основним напрямком покращення проекту має бути розширення функціоналу систем тестування та опитування відповідно до побажань активної частини користувачів.

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		41

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Підручники

1. Герберт Ш. С# 4.0: Повне керівництво.: Пер. з англ. — Х.: ТОВ "И.Д. Вільямс", 2011. — 1056 с.
2. Стиллмен Е. Вивчаємо С# / Стиллмен Е., Грин Дж. – 2-ге вид, – К.: Знання-Прес, 2012. – 696 с.
3. Бхаргава А. Грокаем алгоритми. Ілюстрований посібник для програмістів. - К.: Марвін, 2017. - 288 с.
4. Флэнаган Д. JavaScript. Детальне керівництво. – Пер. с англ. – К: Символ-Плюс, 2008. – 992 с.

Електронні ресурси

5. MDN JavaScript Навчальні матеріали [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en/docs/Web/JavaScript> (дата звернення: 22.03.2021)
6. Microsoft Docs ASP.NET Core 5 Посібник [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-en/aspnet/core/tutorials/razor-pages/?view=aspnetcore-5.0> (дата звернення: 03.03.2021)
7. Microsoft Docs Entity Framework Core Керівництво [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-en/ef/core/modeling/> (дата звернення: 28.02.2021)
8. W3School HTML Tutorial [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/html/default.asp> (дата звернення: 22.02.2021)

					<i>КНТЕУ 121 07-23.БР</i>			
					<i>Розробка програмного забезпечення он-лайн опитувань та тестувань</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		СД	42	43
Зав. кафедри		Криворучко О.В.		14.12.2020	<i>Список використаних джерел</i>	<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		
Керівник		Котенко Н.О.		14.12.2020				
Гарант		Цензура М.О.		14.12.2020				
Розроб.		Юрченко С.С.		14.12.2020				

9. Refactoring.Guru Design Patterns [Електронний ресурс]. – Режим доступу: <https://refactoring.guru/design-patterns> (дата звернення: 21.02.2021)
10. Metanit Повне керівництво по мові програмування C # 9.0 і платформі .NET 5 [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/tutorial/> (дата звернення: 21.03.2021)
11. Bootstrap Документація [Електронний ресурс]. – Режим доступу: <https://getbootstrap.com/docs/4.3/getting-started/introduction/> (дата звернення: 25.03.2021)

					<i>КНТЕУ 121 07-23.БР</i>	<i>Аркуш</i>
						43
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ДОДАТКИ

Додаток А

Клас контролера доступів до тестування

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Surveys_and_Tests.Models;
using Surveys_and_Tests.Models.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Surveys_and_Tests.Controllers.TestControllers
{
    [Authorize]
    public class TestAccess : Controller
    {
        ApplicationDbContext db;
        public TestAccess(ApplicationContext context)
        {
            db = context;
        }

        [HttpGet]
        public async Task<IActionResult> ViewAllAccess(int? TestId = null)
        {
            List<Test_access> AllAccess = await
            db.Tests_access.Include(u=>u.Test).Include(u=>u.Group).Where(u => u.Test.Creator.Email ==
            User.Identity.Name).ToListAsync<Test_access>();
            ViewBag.TestId = TestId;
            return View(AllAccess);
        }

        [HttpGet]
        public async Task<IActionResult> AddAccess()
        {
            Profile profile = await db.Profiles.FirstOrDefaultAsync(u => u.Email ==
            User.Identity.Name);
            TestAccessModel model = new TestAccessModel();
            model.AllTests = await db.Tests.Where(u => u.Creator ==
            profile).ToListAsync<Test>();
            model.AllGroups = await db.Groups.Where(u =>
            u.Profiles.Contains(profile)).ToListAsync<Group>();
            return View(model);
        }

        [HttpPost]
        public async Task<IActionResult> AddAccess(Test_access NewAccess)
        {

```



```

        if (NewAccess != null)
        {
            Test CheckTest = await db.Tests.Include(u=>u.Creator).FirstOrDefaultAsync(u =>
u.Id == NewAccess.Test.Id);
            Group CheckGroup = await
db.Groups.Include(u=>u.Profiles).FirstOrDefaultAsync(u => u.Id == NewAccess.Group.Id);
            if (CheckTest != null && CheckGroup != null)
            {
                Profile profile = await db.Profiles.FirstOrDefaultAsync(u => u.Email ==
User.Identity.Name);
                if (CheckTest.Creator.Email == User.Identity.Name &&
CheckGroup.Profiles.Contains(profile))
                {
                    Test_access Access = new Test_access();
                    Access.Group = CheckGroup;
                    Access.Test = CheckTest;
                    Access.Minutes = NewAccess.Minutes;
                    if (NewAccess.Attempts==0)
                    {
                        Access.Attempts = 1;
                    }
                    else
                    {
                        Access.Attempts = NewAccess.Attempts;
                    }
                    Access.Show_answers = NewAccess.Show_answers;
                    db.Tests_access.Add(Access);
                    await db.SaveChangesAsync();
                }
            }
        }
        return RedirectToAction("ViewAllAccess", "TestAccess");
    }

    [HttpGet]
    public void DeleteAccess(int AccessId)
    {
        Test_access DeleteAccess = db.Tests_access.Include(u =>
u.Test.Creator).Include(u=>u.test_Results).FirstOrDefault(u => u.Id == AccessId);
        if (DeleteAccess.Test.Creator.Email==User.Identity.Name)
        {
            db.Tests_access.Remove(DeleteAccess);
            db.SaveChanges();
        }
    }
}
}
}

```