

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ  
ВИЗНАЧЕННЯ УСПІШНОСТІ НАВЧАННЯ СТУДЕНТІВ З  
ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ C#»**

Виконав: студент 6 курсу, групи ПД-43  
спеціальності  
121 Інженерія програмного забезпечення  
(шифр і назва спеціальності/спеціалізації)

Кравчук В.С.

(прізвище та ініціали)

Керівник Гаманюк І.М.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Київ –2022

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ ” 2022 року

### ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

**КРАВЧУКУ ВЛАДИСЛАВУ СЕРГІЙОВИЧУ**

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для визначення успішності студентів за допомогою мови програмування C#»

Керівник роботи: Гаманюк І.М. ст. викладач

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «16» лютого 2022 року №22.

2. Строк подання студентом роботи «6» червня 2022 року

3. Вхідні дані до роботи

Інструменти з обробки, розробки та зберігання: Visual Studio 2022, pgAdmin, Git, GitHub, SourceTree;

Існуючі аналоги: Moodle, Excel та електронна пошта;

Методи обробки та зберігання даних.

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

4.1 Задача побудови онлайн журналу для використання в вищих навчальних закладах.

4.2 Вимоги та оцінка якості системи.

4.3 Проектування та реалізація веб-застосунку.

4.4 Ручне тестування.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Аналіз аналогів
  2. Функціональність системи
  3. Функціональні та нефункціональні вимоги
  4. Засоби реалізації
  5. Архітектура
6. Дата видачі завдання «11» квітня 2022 року

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз аналогів	11.04-13.04	Виконано
2	Створення вимог до системи	14.04-17.04	Виконано
3	Створення концепції та архітектури	18.04-25.04	Виконано
4	Розробка застосунку	25.04-19.05	Виконано
5	Моделювання та розробка бази даних	20.05-24.05	Виконано
6	Вступ, висновки, реферат	25.05-29.05	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	30.05-1.06	Виконано
8	Попередній захист роботи	02.06	Виконано
9	Здача роботи	06.06	

Студент \_\_\_\_\_  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_  
( підпис ) (прізвище та ініціали)





## РЕФЕРАТ

Текстова частина магістерської роботи 71 с., 30 рис., 34 джерел.

Ключеві слова: Журнал, Оцінки, Веб-додаток, база даних, застосунок, дистанційне навчання.

*Об'єкт дослідження* – процес моніторингу успішності студентів.

*Предмет дослідження* – програмне забезпечення моніторингу успішності студентів.

*Мета роботи* – покращення стану моніторингу успішності студентів.

*Методи дослідження* – методи спостереження та порівняння.

Перш за все в роботі проведено аналіз та порівняння вже існуючих програмних продуктів. Тому спочатку я використав пошук в інтернеті і в результаті побачив що є навчальні заклади в яких існує електронний журнал. Для порівняння їх між собою мені потрібно було зареєструватися до їх електронного журналу, але в мене не вийшло так як для підключення необхідно було щоб електронна пошта на яку реєструється користувач була внесена до бази даних. Отже даний метод аналізу мені не підходить.

Далі я спробував знайти електронний журнал у власному навчальному закладі. Проаналізувавши головний сайт та сайт дистанційного навчання я зробив висновок, що на сайті дистанційного навчання “Moodle” є предмети з необхідними матеріалами для навчання та завдання для виконання які можливо здати та отримати за них оцінку.

Але я виявив проблему повязану з тим що ми не завжди користуємося сайтом дистанційного навчання так як можемо отримати окремі завдання на інших платформах або навіть письмово. Тому ми не завжди можемо подивитися на отримані оцінки так як вони знаходяться в різних місцях і їх зазвичай потрібно шукати.

Отже для вирішення даної проблеми потрібно створити власний продукт який буде зберігати в собі всі оцінки студентів які можливо в будь-який проміжок

часу переглянути. Це дозволить проаналізувати свої оцінки та покращити їх коли це необхідно.

*Галузь використання* – моніторинг успішності студентів в вищих навчальних закладах.

## Зміст

<b>ВСТУП.....</b>	<b>9</b>
<b>1 ЗАДАЧА ПОБУДОВИ ОНЛАЙН ЖУРНАЛУ ДЛЯ ВИКОРИСТАННЯ В ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ.....</b>	<b>10</b>
1.1 Аналіз предметної області.....	10
1.2 Аналіз існуючих аналогів.....	11
<b>2 ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ .....</b>	<b>16</b>
2.1 Функціональні вимоги .....	16
2.2 Нефункціональні вимоги .....	17
2.3 Системні вимоги.....	18
2.4 Якість системи .....	19
<b>3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ.....</b>	<b>20</b>
3.1 Засоби реалізації.....	20
3.2 Архітектура API .....	26
3.2.1 Рівень DataAccess .....	26
3.2.2 Рівень Model .....	27
3.2.3 Рівень Application.....	28
3.2.4 Рівень API .....	30
<b>4 Ручне тестування.....</b>	<b>35</b>
4.1 Тестування функціоналу акаунта .....	35
4.2 Функціонал груп.....	41
4.3 Функціонал предметів .....	50
4.4 Функціонал оцінок .....	55
<b>ВИСНОВКИ .....</b>	<b>10</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>11</b>
<b>Додаток А .....</b>	<b>13</b>



## ВСТУП

*Обґрунтування вибору теми та її актуальність:* під час навчання в умовах карантину чи інших випадків при яких все навчання переходить в дистанційний режим навчання я зустрівся з проблемою знаходження оцінок за виконану роботу. Проблема полягала в тому що ми отримували дистанційно завдання на різних платформах й так само відправляли свої готові роботи на перевірку вчителю, а вже потім отримували в відповіді оцінку. Звідси й було знайдено проблему так як оцінки є на 3 чи більше платформах і їх потрібно завжди шукати. Коли студенту потрібно переглянути чи наявна у нього оцінка по темі або якщо необхідно ще щось додати, потрібно знайти платформу де ми вивчаємо предмет і вже там шукати оцінку якщо вона є і вже потім приймати рішення.

Тому для вирішення даної проблеми я спроектував веб-додаток для внесення, зберігання та перегляду оцінок. І це в свою чергу дозволить кожному влюбий момент часу зайти і переглянути його оцінки по предмету і допоможе зорієнтуватись в подальших діях по данному предмету. Також даний додаток легкий в використанні і тому кожен зможе з ним працювати.

*Ступінь вивчення проблеми:* На даний момент існує небагато продуктів які виконують роль онлайн журналу тому проблема не може бути цілком вивченою через нехватку інформації для обробки. Але з існуючими на даний момент продуктами та їх варіантами заміни можливо зробити свої висновки.

У зв'язку з широким переходом навчання виникла велика потреба в онлайн сервісах для навчання а це в свою чергу призвело до появи проблеми з переглядом оцінок. Тому для вирішення проблеми з оцінками застосовуються різні продукти.

Для внесення та зберігання оцінок ми використовуємо в основному три програмних продукта: по перше оцінки зберігаються на сайті дистанційного навчання Moodle, по друге для деяких предметів вчитель створює теблцю з всіма оцінками в Excel, по третє деякі роботи відправляють на пошту вчителів та в відповіді отримуємо оцінки.

*Об'єктом дослідження* є процес моніторингу успішності студентів.

*Предметом роботи* є програмне забезпечення моніторингу успішності студентів.

*Метою роботи* є покращення стану моніторингу успішності студентів.

*Завданням роботи* є розробка веб-додатку з функціоналом: збереження та перегляду оцінок по предмету.

*Методика дослідження:* Перш за все в роботі проведено аналіз та порівняння вже існуючих програмних продуктів. Тому спочатку я використав пошук в інтернеті і в результаті побачив що є навчальні заклади в яких існує електронний журнал. Для порівня їх між собою мені потрібно було зареєструватися до їх електронного журналу, але в мене не вийшло так як для підключення необхідно було щоб електронна пошта на яку реєструється користувач була внесена до бази даних. Отже даний метод аналізу мені не підходить.

Далі я спробував знайти електронний журнал у власному навчальному закладі. Проаналізувавши головний сайт та сайт дистанційного навчання я зробив висновок, що на сайті дистанційного навчання “Moodle” є предмети з необхідними матеріалами для навчання та завдання для виконання які можливо здати та отримати за них оцінку.

Але я виявив проблему повязану з тим що ми не завжди користуємося сайтом дистанційного навчання так як можемо отримати окремі завдання на інших платформах або навіть письмово. Тому ми не завжди можемо подивитися на отримані оцінки так як вони знаходяться в різних місцях і їх зазвичай потрібно шукати.

Отже для вирішення даної проблеми потрібно створити власний продукт який буде зберігати в собі всі оцінки студентів які можливо в будь-який проміжок часу переглянути. Це дозволить проаналізувати свої оцінки та покращити їх коли це необхідно.

*Наукова новизна роботи:* полягає в вдосконаленні простого журналу в його електронну версію з покращеним функціоналом зручності та його доступності.

*Практична значущість результатів:* даний програмний продукт може бути використаний в будь якому навчальному закладі.

# 1 ЗАДАЧА ПОБУДОВИ ОНЛАЙН ЖУРНАЛУ ДЛЯ ВИКОРИСТАННЯ В ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

## 1.1 Аналіз предметної області

В наш час навчання являється невідомою частиною нашого життя і займає дуже велику роль в подальшому житті. Навчальні матеріали постійно доповнюються, обновлюються та розвиваються. Тому паралельно з цим потрібно покращувати та розвивати технології навчання.

Більшість тенденцій розвитку опирається на цифрове навчання де кожний може легко знайти матеріали та приклади по необхідній темі в інтернет ресурсах. Це дозволяє будь-кому мати доступ до навчання будь-де та будь-коли за наявності інтернету.

За даними Державної служби якості освіти України на період карантину 97% закладів освіти в Закарпатській області використовують дистанційне навчання, а так як карантин впроваджений по всій Україні то і в інших областях схожий результат. Також було проведено опитування щодо подальшого використання засобів дистанційного навчання і 41% вчителів зазначили що мають бажання в подальшому після карантину використовувати дистанційне навчання, 54% зазначили що частково використовуватимуть і лише 5% зазначили що надалі не використовуватимуть.

Тому разом з дистанційним навчання розвивається і ведення журналу оцінок. Звичайний написаний журнал нам не підходить так як його неможливо переглянути дистанційно. Зазвичай в застосунках дистанційного навчання є можливість виставлення оцінок за виконану роботу але ці сервіси розвинені не цілком і тому є випадки виконання робіт на різних платформах. Звідси і виникає проблема з пошуком оцінок на різних інформаційних ресурсах.

З вирішенням даної проблеми допомагає централізована система в якій всі оцінки зберігаються в базі даних і всі зареєстровані користувачі можуть

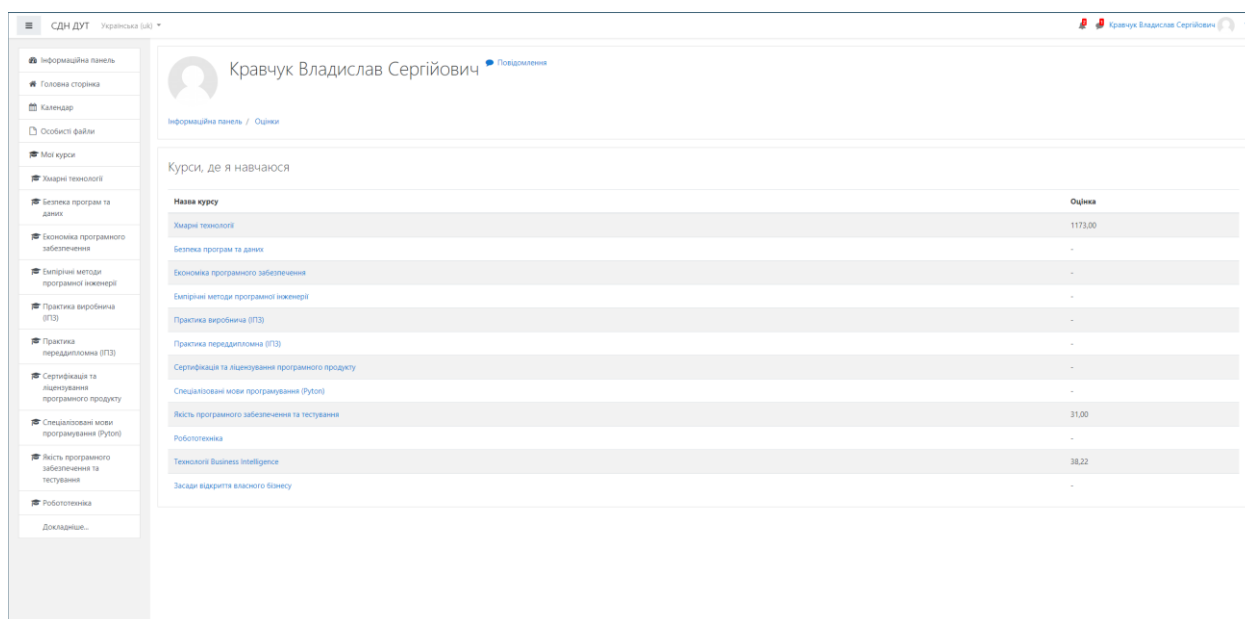
переглянути журнал по необхідному предмету. Використання даної системи дозволить значно спростити ведення журналу та мати доступ до нього онлайн в будь-який момент.

## 1.2 Аналіз існуючих аналогів

На даний момент проаналізувати існуючі аналоги журналу оцінок немає можливості так як дана програма чи веб-додаток існує в малій кількості університетів та доступ до них обмежений для незареєстрованих електронних пошт в базі даних.

Тому для проведення аналізу я використав програмні продукти якими ми користуємося самі. До таких програм входять: сайт дистанційного навчання Moodle, таблиці Excel та в деяких випадках власна пошта.

По-перше, сайт дистанційного навчання Moodle – це засіб для дистанційного навчання в якому зберігаються предмети з всім навчальним матеріалом, завданнями та інколи тестами. Для отримання оцінки спочатку необхідно створити завдання а також створити місце для завантаження готової роботи і тільки потім можливо отримати оцінку від вчителя. Оцінки можна переглядати в особистому кабінеті, також присутнє автоматичне сумування загальної оцінки але ця функція працює не зовсім коректно так як вона не враховує різних факторів оцінок. Також матеріали не завжди оновлюються або відправляються в іншому форматі що призводить до відсутності оцінок (наприклад: Рисунок1.1 – Приклад оцінок на сайті Moodle). Отже з цього можна зробити висновок що перевіряти та виставляти оцінки на сайті дистанційного навчання Moodle потребує багато часу та зусиль так як завжди потрібно створювати завдання та тести за які потім можливо отримати оцінку.



The screenshot shows a Moodle user profile for 'Кравчук Владислав Сергійович'. The main content area is titled 'Курси, де я навчаюся' (Courses I am studying) and contains a table with the following data:

Назва курсу	Оцінка
Хмарні технології	1173,00
Безпека програм та даних	-
Економіка програмного забезпечення	-
Емпіричні методи програмної інженерії	-
Практика виробника (ІТЗ)	-
Практика переддиплома (ІТЗ)	-
Сертифікація та ліцензування програмного продукту	-
Спеціалізовані мови програмування (Python)	-
Якість програмного забезпечення та тестування	31,00
Робототехніка	-
Технології Business Intelligence	38,22
Засади етикетки власного бізнесу	-

Рисунок 1.1 - Приклад оцінок на сайті Moodle

По-друге, я провів аналіз популярного додатку Excel який часто використовують як журнал з оцінками по предмету. Провівши аналіз я зрозумів що даний додаток користується більшою популярністю у вчителів так як програма має свої поля в які можна вносити дані (в поля можливо внести як числа так і текст) це дозволяє створити стовпці з оцінками і назвою теми (наприклад Рисунок 1.2 – Приклад оцінок в програмі Excel). В Excel присутні вмонтовані математичні функції за допомогою яких можливо будь-яким способом рахувати оцінки. Але для нових користувачів Excel є деякі проблеми з складністю освоєння так як на панелі інструментів присутня велика кількість кнопок з функціоналом. Також більшість математичних функцій розраховані на великі математичні обчислення які непотрібні для ведення журналу. З цього можливо зробити висновок що Excel добре підходить для ведення журналу оцінок але він більше орієнтований на складні математичні операції і складний в освоєнні новим користувачам.

	В	С	Д	Е	Ф	Г	Н
	ПД-43		Сертифікат	Доповідь	Презентація	Контрольна робота	Залік
1	Воронський Микита Андрійович	к					
2	Заїчко Ілля Олегівич	к	10		20		30
3	Зволінська Альона Аркадійвна	к		10	20		30
4	Киричов Олександр Петрович		10	10	20	50	90
5	Кобляцький Ілля Анатолійович	к	10	10	20	40	80
6	Король Роман Євгенійович	к	10	10	20	45	85
7	Косенко Андрій Павлович	к	10	10	20	50	90
8	Котул Олександр Юрійович	к	10	10	20	45	85
9	Кравчук Владислав Сергійович	к	10	10	20	50	90
10	Лозинський Станіслав Вадимович	к					
11	Луїгна Олександр Володимирович	к		10	20		30
12	Матківський Аскольд Олегівич	к	10	10	20		40
13	Мельник Максим Аркадійович			10	20	50	80
14	Музика Антон Павлович	к	10		20	50	80
15	Музика Владислав Павлович	к	10	10	20	50	90
16	Полтавець Наталя Віталіївна	к	10	10	20	55	95
17	Сарданов Володимир Ігорівич	к	10	10	20	55	95
18	Собко Іван Іванович	к	10	10	20	45	85
19	Соловейкін Олексій Миколайович	к				45	45
20	Стратіско Серафим Юрійович	к					
21	Цатурян Едуард Русланович	к	10	10	20	50	90
22	Швецов Владислав Ігорівич	к	10	10	20	55	95
23	Шиббаєва Тамара Дмитрівна	староста	к				

Рисунок 1.2 - Приклад оцінок в програмі Excel

По-третє, я провів аналіз пошти на яку інколи отримуємо завдання та відправляємо готову роботу на перевірку. Потім отримуємо у відповіді оцінку. Таким чином вчителів легко виставити оцінку але для учнів даний спосіб має багато недоліків так як отриману оцінку складно знайти серед отриманих листів. Також всі оцінки зберігаються в окремих місцях тому щоб переглянути їх всі по необхідному предмету необхідно продивитися всі листи (наприклад 1.3 – Приклад отриманої оцінки на пошту). Отже можна зробити висновок що використання пошти для журналу не підходить для учнів через важливі недоліки.

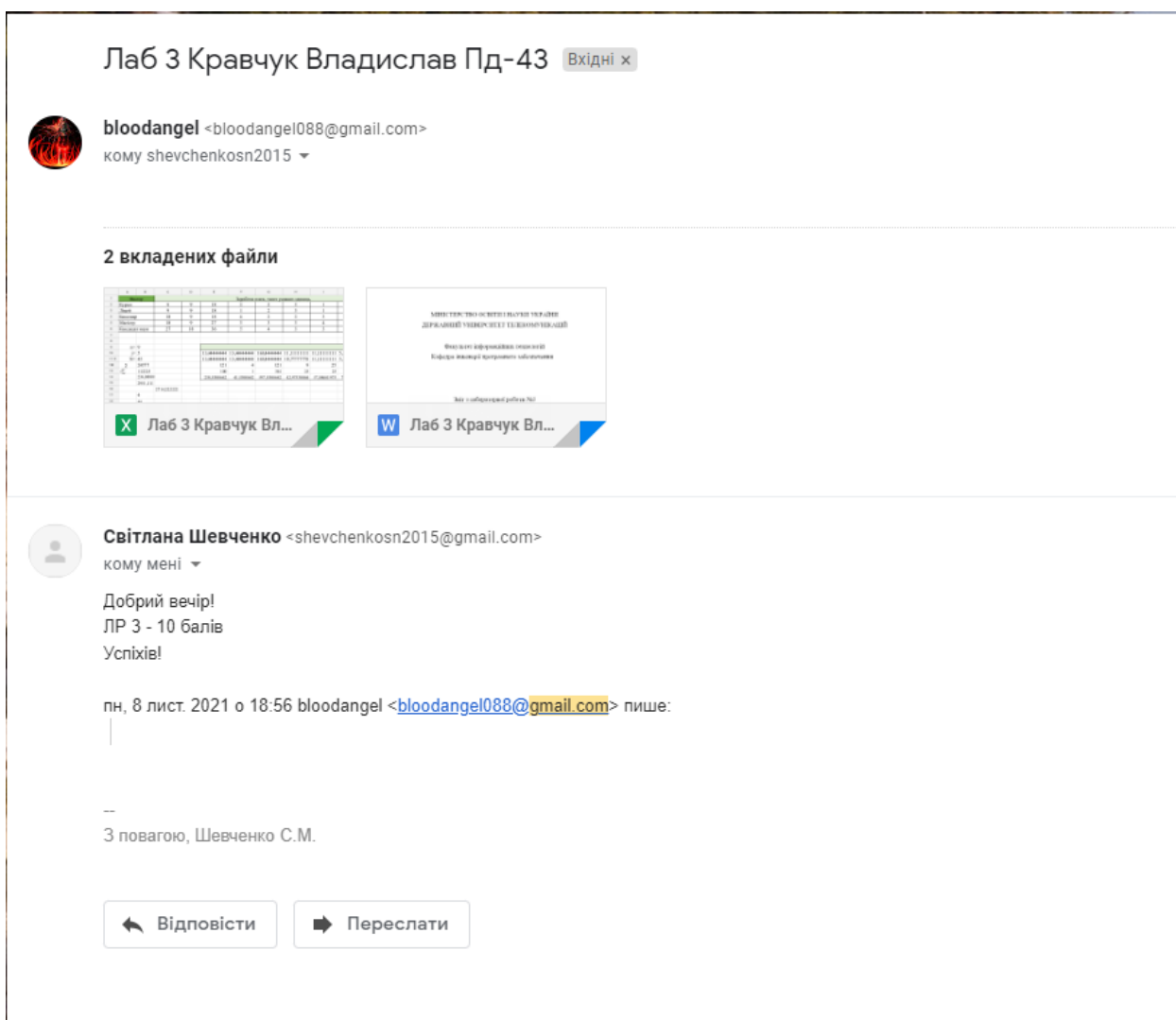


Рисунок 1.3 – Приклад отриманої оцінки на пошту

Після аналізу існуючих аналогів я провів порівняння їх між собою та власним запропонованим продуктом використовуючи різні критерії а результат записував трьома видами де: (+) - повністю задовольняє, (+-) - частково задовольняє та (-) - зовсім незадовольняє або відсутнє (Таблиця 1.1).

Таблиця 1.1 – Аналіз аналогів

Критерії аналізу	Moodle	Excel	Пошта	EMark
Легкий пошук оцінок	+ -	+	-	+



Критерії аналізу	Moodle	Excel	Пошта	EMark
Отримання всіх оцінок по предмету	+	+	-	+
Легкість додавання оцінок	+-	+	+	+
Легкість вивчення програми	+-	+-	+	+
Наявність власного акаунту	+	+	+	+
Перегляд оцінок учнем	+	+	+	+
Керування з різним доступом	+	+	-	+
Створення власного акаунту	-	-	+	+
Керування власним акаунтом	-	-	+	+

## 2 ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ

### 2.1 Функціональні вимоги

Основним функціоналом для веб-застосунку ведення журналу є створення журналу та його отримання користувачем.

По-перше, потрібно розробити функціонал реєстрації користувача і додати можливість вибору ролі (Teacher , Student). Також для зручності користування необхідно додати функціонал редагування даних що дозволить змінити: пошту, пароль та ПІБ. Видалення акаунту користувача може здійснити лише сам власник і лише за відсутності участі в групі. Кожен користувач має можливість переглядати інформацію про себе а вчителі мають можливість перевіряти інформацію певного користувача.

По-друге, для зберігання і розподілу великої кількості студентів необхідно мати функціонал груп до яких можливо додавати студентів та вчителів. Редагування групи та її учасників також необхідне при змінах. Для легшої перевірки групи учителем необхідно додати функціонал перегляду всіх вчителів та студентів в групі.

По-третє, для роботи з предметами використовується функціонал: додавання, редагування, отримання інформації про предмет та видалення. При створенні предмету вчитель який виконує даний запит зберігається як керівник по цьому предмету. Для оцінок по предмету необхідно додати заголовок стовпця (за що виставлена оцінка) та додати перевірку на наявність оцінки. Реалізувати оновлення оцінок та назви стовпця. Для кожного зареєстрованого користувача додати функціонал отримання журналу оцінок.

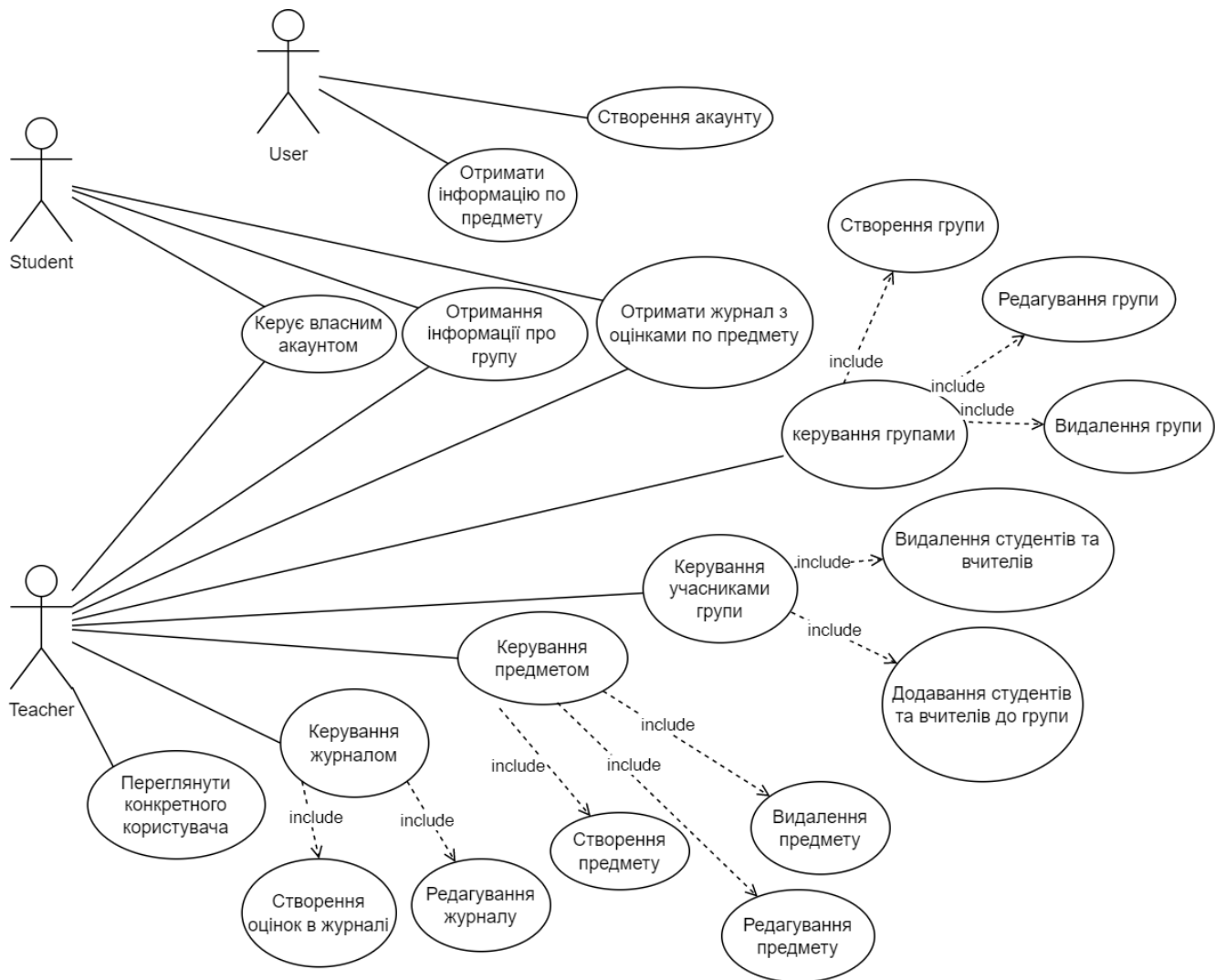


Рисунок 2.1 – Функціональність ситеми

## 2.2 Нефункціональні вимоги

Для забезпечення безпеки необхідно додати хешування та використання JWT токену (Рисунок 2.2).

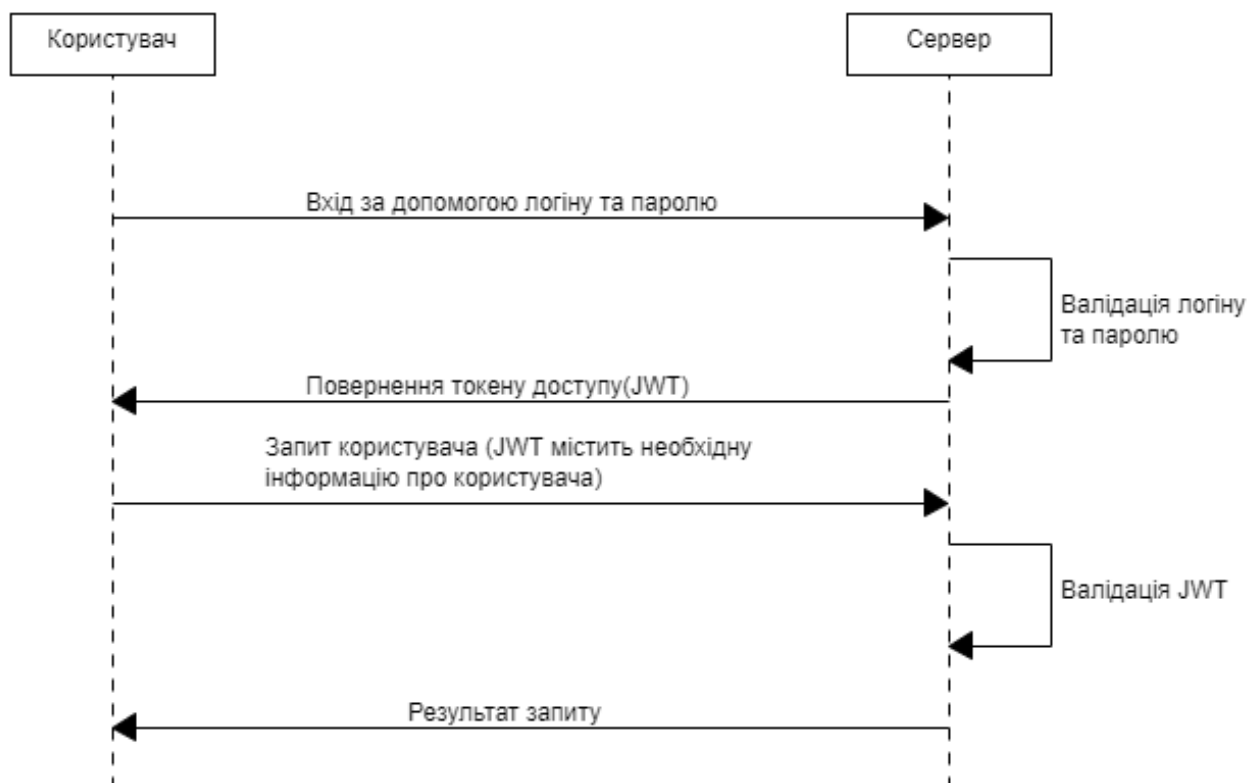


Рисунок 2.2 – Діаграма послідовності перевірки користувача

Також для безпеки основний функціонал видалення студентів дозволений тільки вчителям, а самі вчителі можуть видалити з групи тільки себе.

Отримати інформацію по предмету може будь хто, а інший функціонал управління предметом дозволений лише вчителю. При створенні та видаленні предмету необхідна перевірка на наявність вчителя в групі.

### 2.3 Системні вимоги

Електронний журнал для моніторингу успішності являє собою веб-застосунок тому для його роботи необхідний сервер. Для стабільної роботи застосунки досить забезпечити роботу для 500 користувачів разом. Це зумовлено швидким використанням системи лише для отримання журналу й після успішного

отримання журналу користувач більше не потребує послуг застосунку до наступної перевірки оцінок. Основними вимогами до конфігурації серверу є:

- 2-х ядерний процесор 4 потоки, 4.7 GHz
- 4 Gb оперативної пам'яті DDR4
- SSD диск 250 Gb

Для користувацької сторони використання системними вимогами є будь-який компютер чи телефон з доступом до інтернету, що дозволить здійснити вхід і працювати з журналом.

## **2.4 Якість системи**

З наведених вище функціональних вимог можливо зробити висновок що даний веб-застосунок включає в себе основний функціонал онлайн журналу з оцінками. Головною перевагою програмного продукту є швидкість взаємодії з централізованою системою для зберігання даних. Так як непотрібно шукати оцінки в трьох програмах як це робилося в аналогах, для отримання необхідного журналу необхідно спочатку увійти до власного акаунту й після успішного входу зробити запит на отримання журналу оцінок по необхідному предмету. Це дозволяє майже миттєво отримати всі оцінки. Тому використання веб-застосунку для визначення успішності студентів дозволяє замість трьох програм користуватись однією.

## 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

### 3.1 Засоби реалізації

Для реалізації веб-застосунку було використано наступні інструменти: Visual Studio 2022, Git, GitHub, pgAdmin, PowerShell. Також було використано такі програмні засоби: C#, .NET, AutoMapper, Kirpichyov FriendlyJwt, Entity Framework Core та PostgreSQL.

Visual Studio – середовище розробки програмного забезпечення й інших інструментальних засобів. Ця програма розроблена компанією Microsoft. Середовище розробки використовується для створення програмних продуктів з різним функціоналом. Програми можуть бути: консольними, програми з графічним інтерфейсом та веб-застосунками. Visual Studio підтримує декілька мов програмування та має декілька різних версій. Для проектування застосунку було обрано Visual Studio 2022(Рисунок 3.1) як найновіша версія з необхідним функціоналом.



Рисунок 3.1 – Логотип Visual Studio

Git – система для управління версіями (Рисунок 3.2). Ця система призначена для обробки різних по складності проектів що дозволяє постійно контролювати версії програми та повертатися до минулих версій при виникненні такої необхідності. Git легкий до використання та освоєння. Головною перевагою

системи є можливість поділення застосунку на гілки і паралельна робота з ними якщо сама реалізація не конфліктує між собою. Завжди можливо створити нову гілку для впровадження нової функції і якщо виникне помилка є можливість повернутися на стару гілку з поверненням змін.



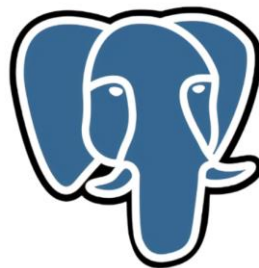
Рисунок 3.2 – Логотип Git

GitHub (Рисунок 3.3) – система управління проектами та версіями коду, яка взаємодіє разом з Git для швидкої та комфортної роботи. Також це платформа соціальної мережі для роботи з іншими людьми, яка дозволяє відслідковувати роботи інших користувачів та спілкуватися з ними. GitHub має в собі зручну функцію відслідковування та перегляду версій, а також являється одним з найбільших онлайн-сховищ програмних продуктів. Широкі можливості Git дозволяють об'єднувати свої репозиторії, а GitHub надає зручний інтерфейс й надалі продемонструвати вклад кожного розробника в вигляді дерева з виконаних ним гілок. Для кожного проекту є своя окрема веб-сторінка яка включає в себе можливість перегляду коду. Кожен користувач має можливість створити приватний репозиторій до якого матиме доступ лише користувач та при необхідності обрані люди.



Рисунок 3.3 – Логотип GitHub

pgAdmin (Рисунок 3.4) – провідний інструмент призначений для управління відкритим вихідним кодом для бази даних Postgres. Інструмент допомагає з моніторингом та управлінням декількома базами даних як локальними так і віддаленими використовуючи єдиний графічний інтерфейс. Цей інтерфейс дозволяє легко керувати та додавати об'єкти БАЗАМИ ДАНИХ, а також іншими інструментами управління.



**pgAdmin**

Рисунок 3.4 – Логотип pgAdmin

PowerShell (Рисунок 3.5) – кросплатформне рішення для автоматизації поставлених задач. Складається з командного рядка, фреймворку керування та мови сценаріїв. Включає в себе функціонал інших аналогів й має свою особливість в можливості приймати й повертати об'єкти .NET так як побудований



на його основі. Мова сценаріїв використовується для автоматизації управління системами так як має функціонал створення, тестування та розгортання рішень.



Рисунок 3.5 – Логотип PowerShell

C# (Рисунок 3.6) - об'єктно-орієнтована мова програмування для платформи .NET. Розроблений компанією Microsoft. За схожістю подібний до C++ та Java. Має в собі необхідний функціонал запозичений від аналогів та певний функціонал неможливий при використанні аналогів. Підходить для різних видів програм, а також для веб-застосунків.



Рисунок 3.6 – Логотип C#

.NET (Рисунок 3.7) – модульна платформа для розробки програмного забезпечення. Підтримує мову програмування C#. В зв'язку з модульністю платформи дозволяє програму розділити на різні модулі з окремими зв'язками та

підключеними різними пакетами. Кожен модуль може працювати як окремий і не залежить від оновлення всієї системи. Добре підходить для розробки веб-застосунку.



Рисунок 3.7 – Логотип .NET

AutoMapper (Рисунок 3.8) – бібліотека яка використовує API вільного налаштування для визначення стратегії відображення об'єкта. Бібліотека спрямована на побудову зв'язків між об'єктом та об'єктом для транспортування даних й інших простих об'єктів. Це дозволяє скоротити час та об'єм написання коду за рахунок готового функціоналу для використання.



Рисунок 3.8 – Логотип AutoMapper

Kirpichyov FriendlyJwt (Рисунок 3.9) – бібліотека з відкритим кодом що дозволяє легко користуватися авторизацією JWT за допомогою готового функціоналу. Розроблена для роботи з модульною платформою .NET Core 5.0.



Рисунок 3.9 – Логотип Kirpichyov FriendlyJwt

Entity Framework Core (Рисунок 3.10) – ядро для взаємодії з сутностями. Це легка, відкрита та кросплатформена O/RM. За допомогою ядра .NET має змогу працювати з базою даних за допомогою своїх об'єктів, що усуває необхідність написання значної частини коду.



Рисунок 3.10 – Логотип Entity Framework Core

PostgreSQL (Рисунок 3.11) – потужна об'єктно-реляційна база даних з відкритим вихідним кодом. Розроблена на основі мови SQL й підтримує більшість можливостей стандарту SQL:2011.



Рисунок 3.11 – Логотип PostgreSQL

## 3.2 Архітектура API

Так як в проекті використовується .NET 5.0 архітектура системи буде розділена на різні рівні. Кожен рівень буде відповідати за свій функціонал та матиме для цього свої підключені пакети які спрощують написання самого коду. Приклад розробленої багаторівневої архітектури зображено на (Рисунок 3.12).

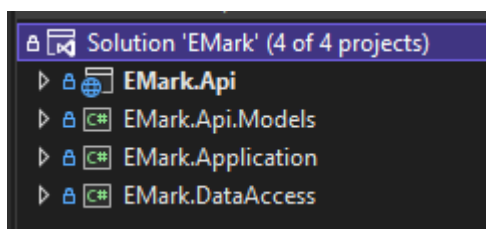


Рисунок 3.12 – Багаторівнева архітектура застосунку

### 3.2.1 Рівень DataAccess

DataAccess відповідає за підключення до бази даних та має в собі всі шаблони сутностей які зберігаються та відображаються в базі даних. Також на цьому рівні будується міграції в базі даних що дозволяє побудувати взаємозв'язки між сутностями приклад зображено на (Рисунок 3.13).

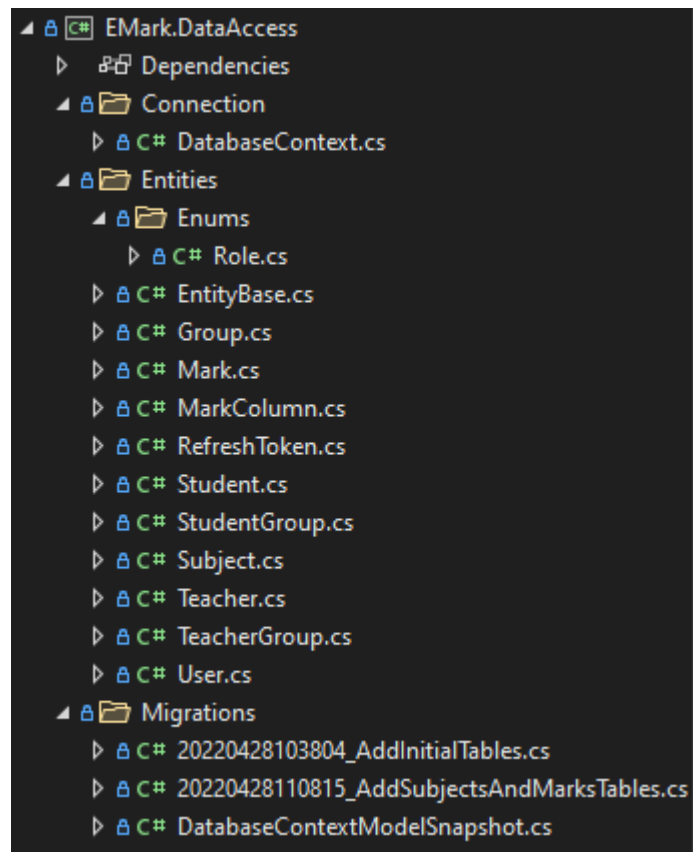


Рисунок 3.13 – Внутрішня архітектура шару DataAccess

### 3.2.2 Рівень Model

Model відповідає за моделі запитів від користувача до методів функціоналу та навпаки від функціоналу до моделі отриманої в відповіді на запит (Рисунок 3.14). Майже кожна сутність має для себе додаткову модель для отримання запиту або відповіді так як не завжди потребує отримувати дані які можуть заповнюватись автоматично або в іншому функціоналі та для відповіді на запит демонструвати лише необхідне.

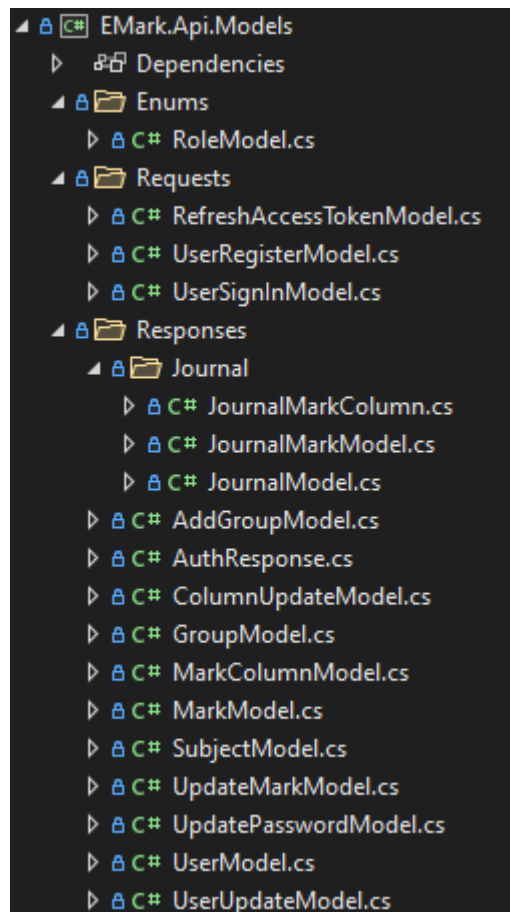


Рисунок 3.14 – Моделі на рівні Models

### 3.2.3 Рівень Application

Application відповідає за основний функціонал застосунку. Кожен функціонал має свій окремий сервіс який має в собі всі операції визначені в технічному завданні. Також цей рівень має в собі інтерфейси для кожного сервісу які використовуються для інєкції залежностей (Рисунок 3.15). Інєкція залежностей це шаблон проектування в якому об'єкт отримує інші об'єкти від яких він залежить. Шаблон гарантує, що об'єкт який хоче використати запозичений функціонал інших об'єктів не повинен знати як побудувати цей функціонал а просто працювати з всим функціоналом відправляючи лише необхідний запит. Інєкція залежностей складається з передачі параметрів в метод та дозволяє змінити його поведінку не редагуючи сам код. Сам користувач не бачить сам код

функціоналу а взаємодіє з ним за допомогою інтерфейсу й робить запити передаючи необхідні дані.

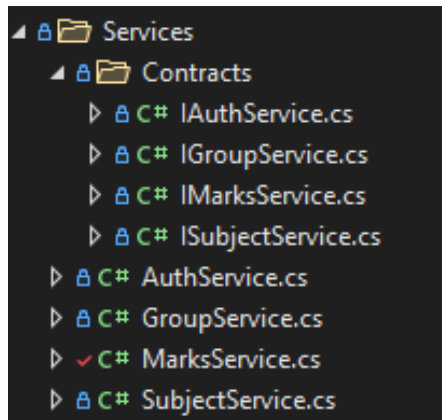


Рисунок 3.15 – Сервіси та Інтерфейси

Для трансферу даних між моделями створюється мапінг та правила для нього (Рисунок 3.16). Так як в застосунку використано бібліотеку AutoMapper правила та сам трансфер виконується вже готовим функціоналом. Трансфер даних виконується між моделями на рівні Models (Рисунок 3.14) та моделями сутностей на рівні DataAccess (Рисунок 3.13). За допомогою функціоналу AutoMapper є можливість додати різні правила мапінгу й серед них головним є можливість додати правило оберненого мапігу. Це дозволяє мати 2 звязки між моделями в одному напрямку та оберненому. Також за допомогою мапінгу реалізовується функціонал хешування паролю при реєстрації за допомогою BCrypt правила. BCrypt це бібліотека яка дозволяє швидко та безпечно хешувати дані та виконувати перевірку при авторизації за допомогою перевірки хешу введеного паролю та збереженого в базі даних.

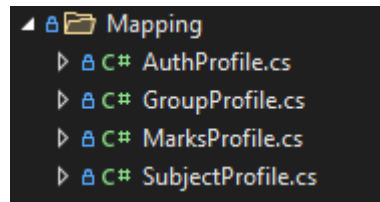


Рисунок 3.16 – Профілі маперу

Для обробки виключень створена обробка сповіщень (Рисунок 3.17), яка при отриманні виключення повертає результат в вигляді тексту помилки який сповіщує про неправильно виконані дії.

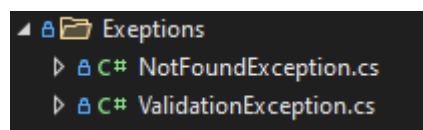


Рисунок 3.17 – Профілі виключень

Для використання JWT для акаунту користувача створюються опції (Рисунок 3.18), які мають в собі характеристики такі як: видавець токenu, кому призначений цей токен, час життя токenu, час життя токenu оновлення та ключ шифрування.

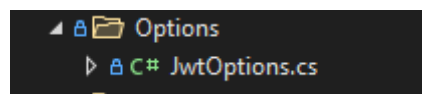


Рисунок 3.18 – Опції JWT

### 3.2.4 Рівень API

API головний засіб управління застосунком. Запуск програми відбувається в класі Program.cs а всі налаштування зберігаються в класі Startup.cs. Startup відповідає за конфігурацію бази даних та за підключення сервісів з шару Application (Рисунок 3.15). Також в Startup підключається сервіс управління



Swagger який спрощує розробку API додатків за рахунок вмонтованих інструментів. Swagger може автоматично зчитувати структуру API та створити автодокументацію API (Рисунок 3.19).

```

services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo {Title = "EMark.Api", Version = "v1"});

    c.MapType<RoleModel>(() => new OpenApiSchema()
    {
        Enum = new List<IOpenApiAny>()
        {
            new OpenApiString(nameof(RoleModel.Teacher)),
            new OpenApiString(nameof(RoleModel.Student))
        }
    });

    c.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme
    {
        In = ParameterLocation.Header,
        Description = "Please insert JWT with bearer into field",
        Name = "Authorization",
        Type = SecuritySchemeType.ApiKey
    });
    c.AddSecurityRequirement(new OpenApiSecurityRequirement
    {
        {
            new OpenApiSecurityScheme
            {
                Reference = new OpenApiReference
                {
                    Type = ReferenceType.SecurityScheme,
                    Id = "Bearer"
                }
            },
            new string[] { }
        }
    });
});

```

Рисунок 3.19 – Використання Swagger

Також на рівні API реалізовано валідацію для всіх даних які має вводити користувач (Рисунок 3.20). Головною валідацією є перевірка введеної пошти на

правильність написання та перевірка на кількість символів паролю. Для введених даних додано перевірку на порожність для недопущення не введених даних. Також крім паролю додано перевірку на кількість мінімальних та максимальних символів для даних в яких потрібно обмежити введення символів.

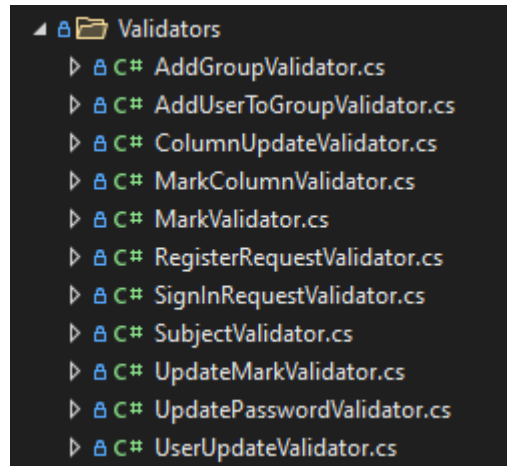


Рисунок 3.20 – Валідатори даних

На рівні API присутні контролери які відповідають за контроль за тим, як користувач взаємодіє з додатком (Рисунок 3.21). Контролер містить логіку керування потоком для програми ASP.NET MVC. Також контролер відповідає за відповідь яку отримує користувач після запиту. Кожен метод записаний в контролерах має свій ендпоінт та має налаштування доступу за ролями так як користувачі з різними ролями мають різний доступний їм функціонал для забезпечення безпеки.

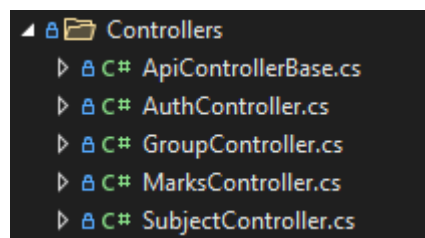


Рисунок 3.21 - Контролери

Деякі архітектурні рівні мають зв'язки між собою які зображені на діаграмі (Рисунок 3.22). Загальна архітектура сутностей зображена на діаграмі класів (Рисунок 3.23).

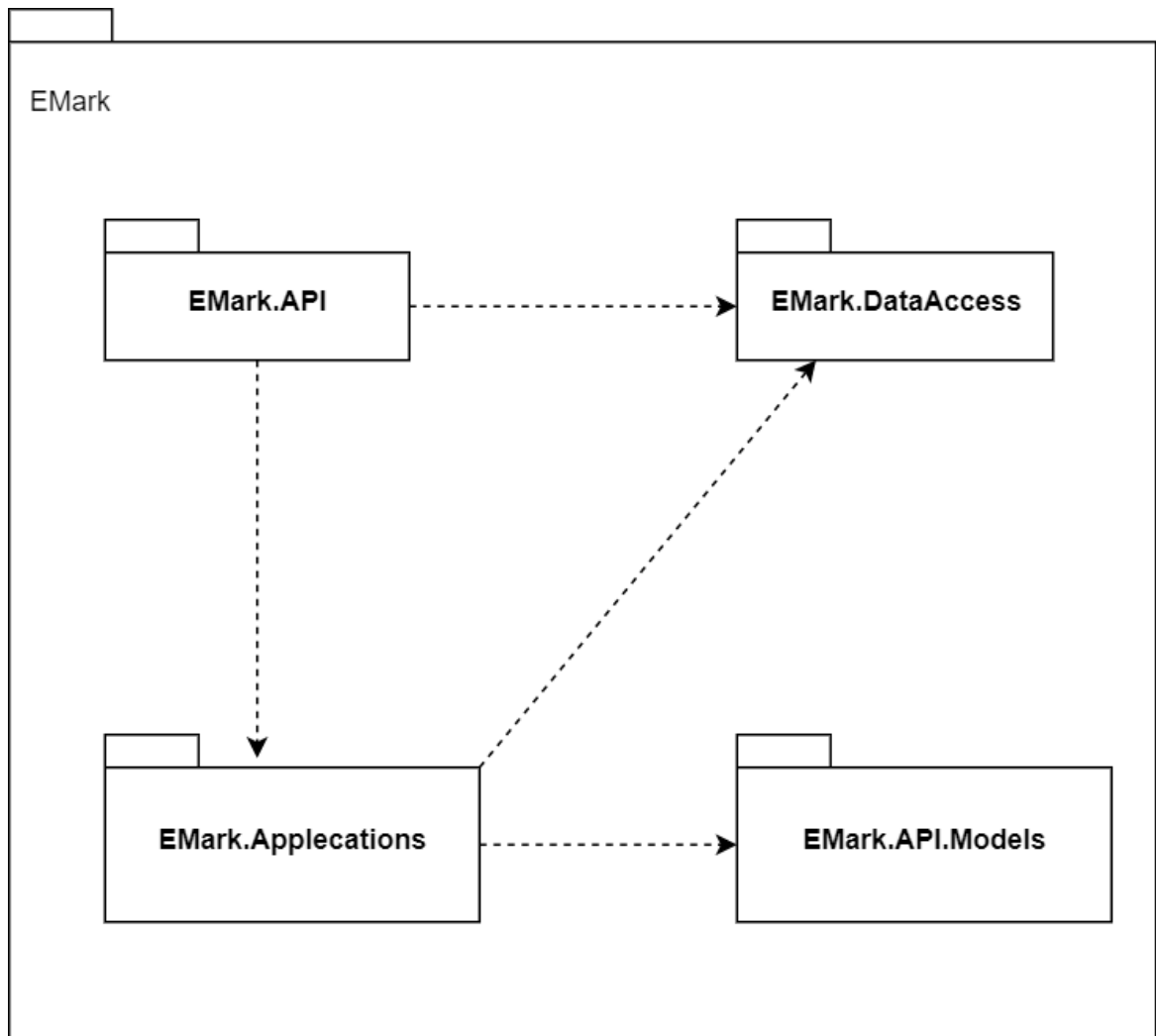


Рисунок 3.22 – Діаграма пакетів

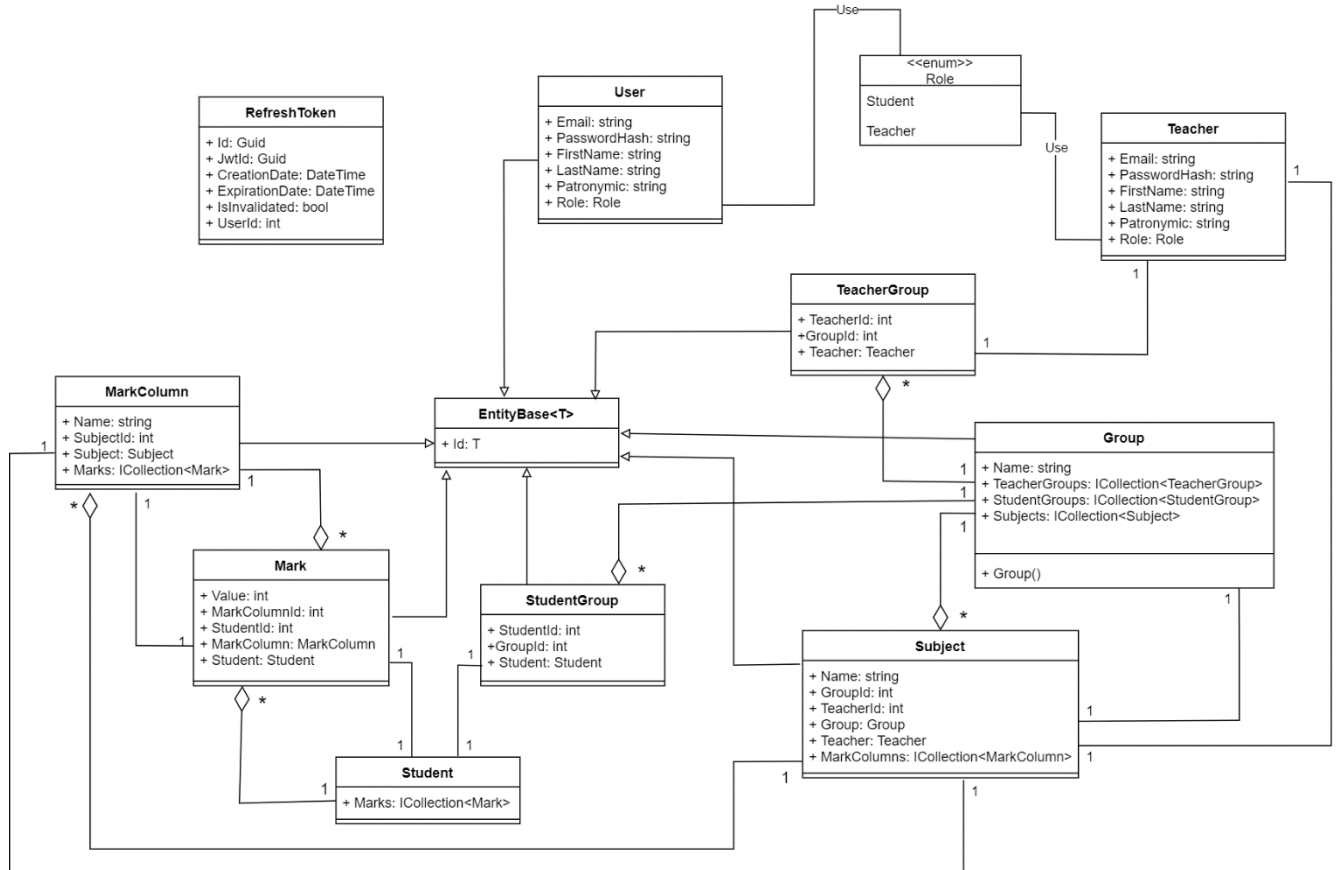


Рисунок 3.23 – Діаграма класів

## 4 Ручне тестування

Тестування дуже важлива частина створення застосунку. Існують різні види тестування але для розробленого застосунку було використано ручне тестування. Ручне тестування це процес перевірки програмного забезпечення на дефекти тестувальником за допомогою ручного введення. Тестувальник відіграє роль користувача й перевіряє кожен доступний метод застосунку на всі помилки та виключення. Ключовою метою тестування є тестування програми на правильну поведінку перед випуском кінцевого продукту.

### 4.1 Тестування функціоналу акаунта

Спочатку необхідно протестувати функціонал реєстрації користувача. Для реєстрації користувач обирає необхідну роль за якою він хоче зареєструватись (Teacher або Student) та вводить логін який за допомогою валідації перевіряється на правильність формату, далі вводить пароль який складається мінімум з 8 символів. Також при реєстрації вводиться ПІБ. Приклад тестування зображено на рисунку 4.1-4.2













Рисунок 4.9 – Зміна паролю користувачем(запит з однаковими паролями)

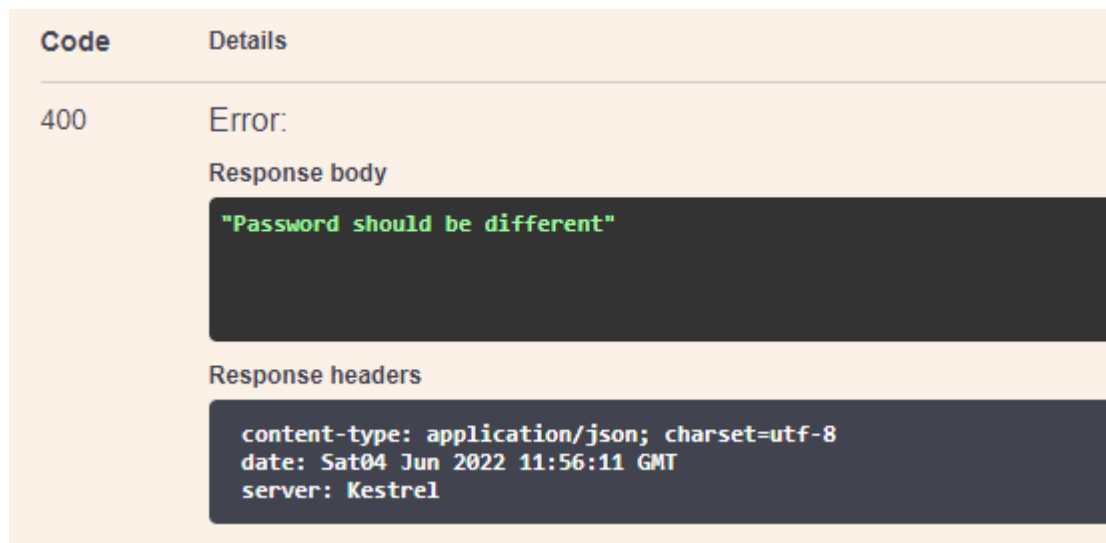


Рисунок 4.10 – Зміна паролю користувача(результат запиту)

Видалення акаунту користувача може виконати сам користувач. Приклад зображено на рисунку 4.11

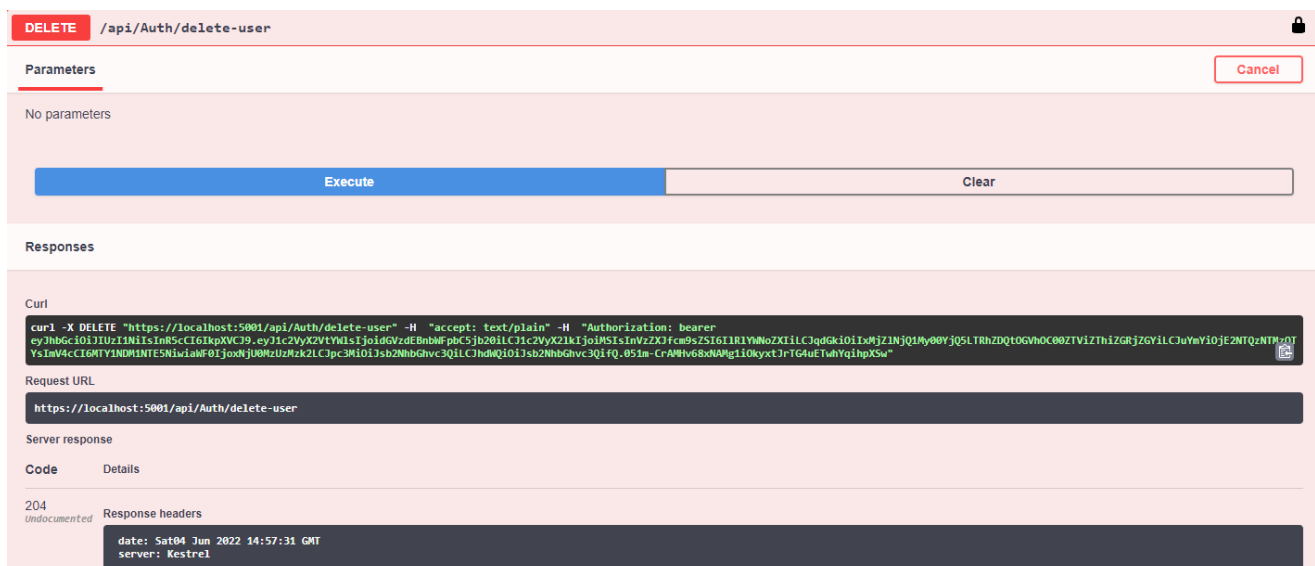


Рисунок 4.11 – Видалення акаунту користувача

## 4.2 Функціонал груп

Створити групу має змогу вчитель. Для створення групи необхідно дати їй назву й виконати запит. Приклад зображено на рисунку 4.12-4.13



Рисунок 4.12 – Створення групи(запит на створення)



Рисунок 4.13 – Створення групи(результат запиту)

При неправильно введеній назві групи вчитель завжди може її змінити виконавши необхідний запит. Приклад зображено на рисунку 4.14-4.15

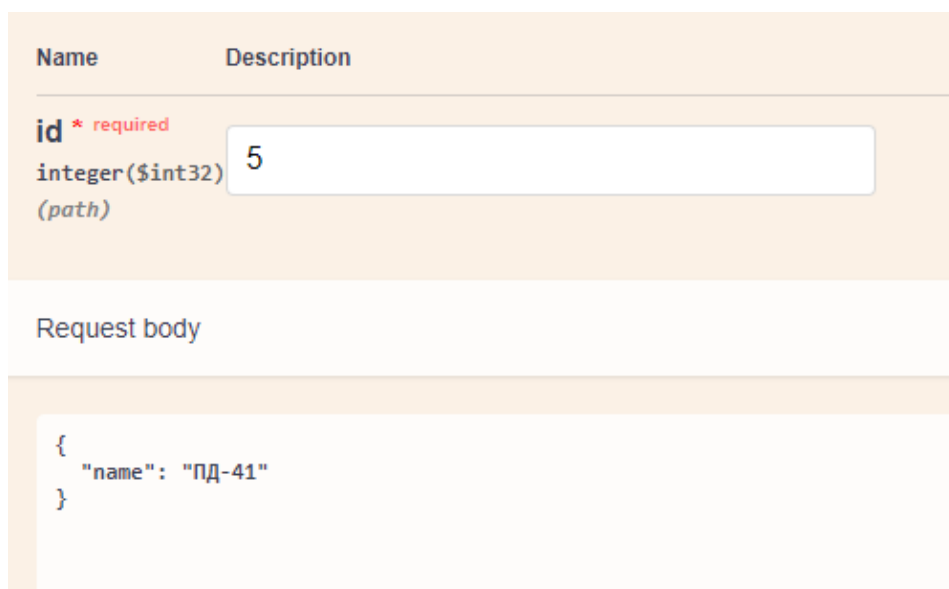


Рисунок 4.14 – Оновлення назви групи(запит на оновлення)







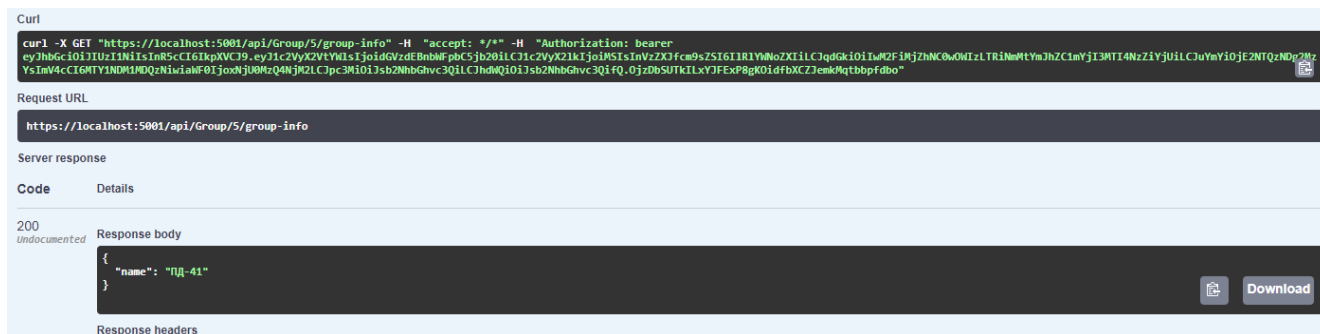


Рисунок 4.21 – Отримання інформації про групу(результат запиту)

Також вчитель може переглянути весь список вчителів та студентів в групі. Приклад зображено на рисунку 4.22-4.25.

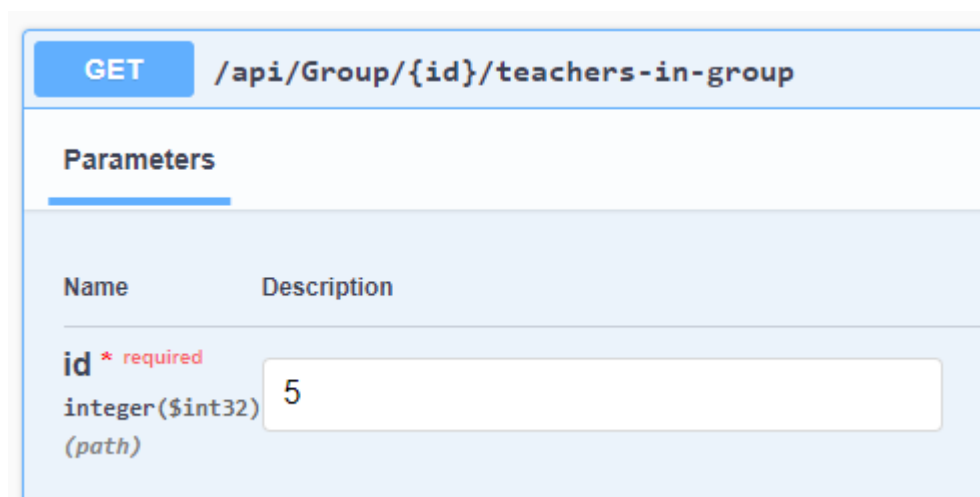


Рисунок 4.22 – Отримання списку вчителів в групі(запит)





Рисунок 4.23 – Отримання списку вчителів в групі(результат запити)

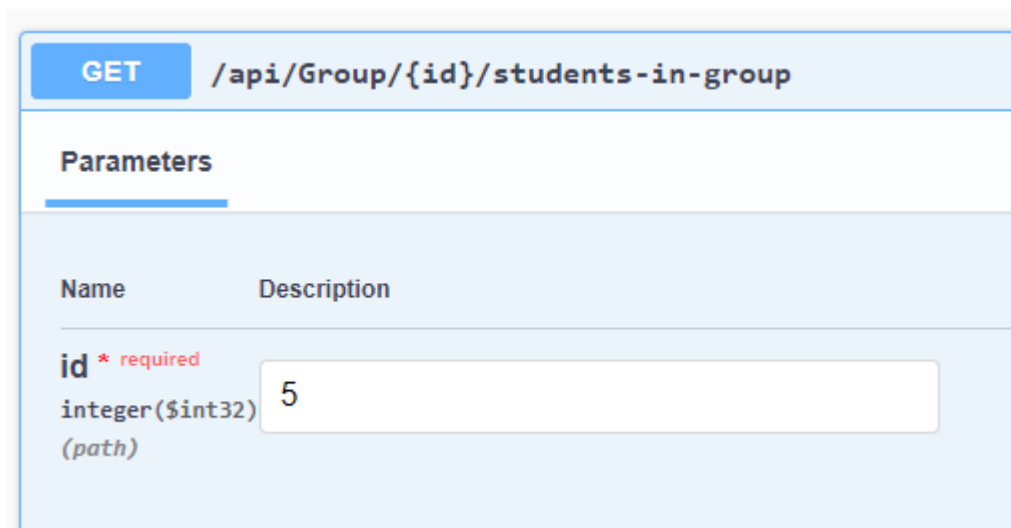


Рисунок 4.24 – Отримання списку студентів в групі(запит)



```
200
Undocumented Response body

[
  {
    "email": "VladKravchuk@gmail.com",
    "firstname": "Владислав",
    "lastname": "Кравчук",
    "patronymic": "Сергійович"
  },
  {
    "email": "test2@gmail.com",
    "firstname": "Ілля",
    "lastname": "Кобиляцький",
    "patronymic": "Анатолійович"
  },
  {
    "email": "test1@gmail.com",
    "firstname": "Максим",
    "lastname": "Мельник",
    "patronymic": "Аркадійович"
  }
]
```

Рисунок 4.25 – Отримання списку студентів в групі(результат запити)

Разом з додаванням вчителів та студентів, вчитель має функціонал виходу з групи щоб залишити її та видвлити студента з групи. Приклад зображено на рисунку 4.26-4.29.



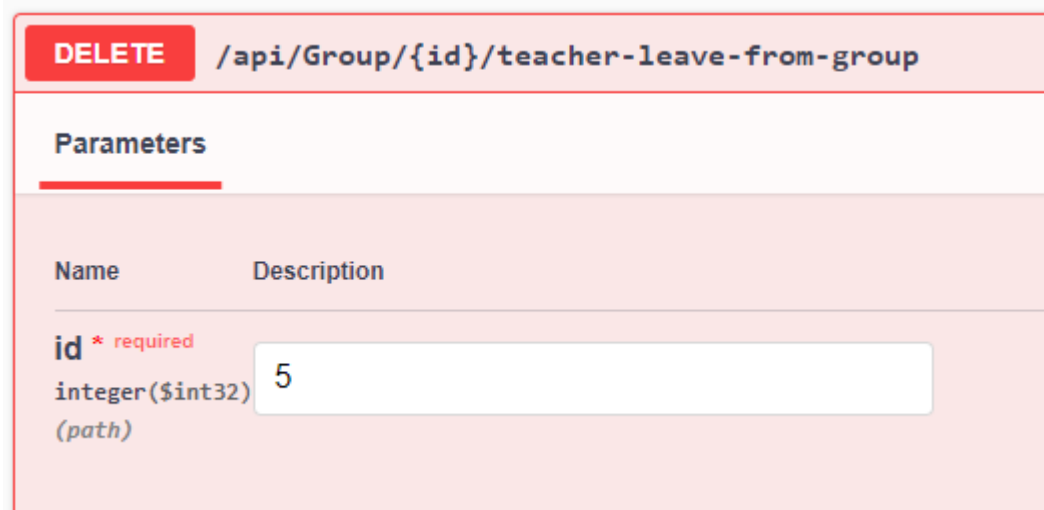


Рисунок 4.28 – Вихід вчителя з групи(запит)



Рисунок 4.29 – Вихід вчителя з групи(результат запиту)

### 4.3 Функціонал предметів

Кожен вчитель в групі має можливість створити предмет за який він відповідає. Приклад зображено на рисунку 4.30-4.31.





GET /api/Subject/subject-info

**Parameters**

Name	Description
<b>subjectId</b> * required integer(\$int32) (query)	<input type="text" value="5"/>
<b>groupId</b> * required integer(\$int32) (query)	<input type="text" value="5"/>

Рисунок 4.34 – Отримання інформації по предмету(запит)

**Code**      **Details**

200  
*Undocumented*

**Response body**

```
{  
  "name": "Захист диплому"  
}
```

**Response headers**

```
content-length: 38  
content-type: application/json; charset=utf-8  
date: Sat04 Jun 2022 14:03:49 GMT  
server: Kestrel
```

Response

Рисунок 4.35 – Отримання інформації по предмету(результат запиту)

Зареєстровані користувачі мають змогу переглянути журнал предмету. Приклад зображено на рисунку 4.36-4.37.

**GET** /api/Subject/{subjectId}/journal

**Parameters**

Name	Description
<b>subjectId</b> * required integer(\$int32) (path)	<input type="text" value="5"/>

Рисунок 4.36 – Перегляд журналу оцінок(запит)

200

Response body

```
{
  "name": "Захист диплому",
  "groupName": "ПД-41",
  "teacherFullname": "Олександр Кирпичов Петрович",
  "columns": [
    {
      "name": "Передзахист диплому 03.06",
      "marks": [
        {
          "value": 74,
          "studentFullname": "Максим Мельник Аркадійович"
        },
        {
          "value": 85,
          "studentFullname": "Ілля Кобиляцький Анатолійович"
        },
        {
          "value": 90,
          "studentFullname": "Владислав Кравчук Сергійович"
        }
      ]
    }
  ],
  {
    "name": "Захист диплому",
    "marks": [
      {
        "value": 75,
        "studentFullname": "Владислав Кравчук Сергійович"
      }
    ]
  }
}
```

Response headers

Рисунок 4.37 - Перегляд журналу оцінок(результат запиту)













## ВИСНОВКИ

Метою роботи було покращення стану моніторингу успішності студентів. Робота була спрямована на реалізацію веб-застосунку для вищих навчальних закладів.

1. Проаналізовано та зрівняно існуючі в даний час аналоги для визначення успішності які використовують в вищих навчальних закладах. Досліджено переваги та недоліки які виникають при їх використанні. Було виявлено недоліки з пошуком оцінок серед декількох аналогів.
2. Сформовано функціональні та нефункціональні вимоги для застосунку на основі переваг та недоліків аналогів. Проаналізовано якість системи в зменшені витрати часу для пошуку оцінок в порівнянні з аналогами.
3. Описано архітектуру, ключові особливості та алгоритм роботи застосунку за допомогою діаграм UML.
4. Досліджено особливості створення, зберігання, обробки, отримання та видалення даних й використано базу даних PostgreSQL для роботи з цими даними.
5. Спроектовано архітектуру системи та розроблено веб-застосунок з використанням різних інструментів та програмних засобів.
6. Протестовано веб-застосунок на наявність недоліків за допомогою ручного тестування.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Результати опитування щодо дистанційного навчання в закладах загальної середньої освіти , - Державна служба якості освіти України [Електронний ресурс] – Режим доступу до ресурсу: <https://deponms.carpathia.gov.ua/news/rezultati-opituvannia-shchodo-distantiinogo-navchannia-v-zakladah-zagalnoyi-serednoyi-osviti>.
2. PostgreSQL documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/>.
3. Information and general information about the Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/uk-ua/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
4. Середовище створення діаграм [Електронний ресурс] – Режим доступу до ресурсу: <https://www.draw.io/index.html>
5. C# The modern, innovative, open-source programming language for building your apps [Електронний ресурс] – Режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/languages/csharp>.
6. What is .NET [Електронний ресурс] – Режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>.
7. Entity Framework Core [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/ef/core/>.
8. About Git [Електронний ресурс] – Режим доступу до ресурсу: <https://git-scm.com/about>.
9. AutoMapper documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.automapper.org/en/latest/>.
10. PowerShell documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/powershell/>
11. API development for everyone [Електронний ресурс] – Режим доступу до ресурсу: <https://swagger.io/>.
12. C# documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/>.

13. FriendlyJwt overview [Электронный ресурс] – Режим доступа до ресурсу:  
<https://github.com/kirpichyov/FriendlyJwt>.
14. SourceTree basics [Электронный ресурс] – Режим доступа до ресурсу:  
<https://confluence.atlassian.com/sourcetreekb/sourcetree-basics-780870007.html>.



## Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ УСПІШНОСТІ  
СТУДЕНТІВ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ С#

Виконав студент 4 курсу  
групи ПД-43  
Кравчук Владислав Сергійович

Керівник роботи  
ст. викладача манюкІгор Михайлович

Київ - 2022

## Аналоги



## Аналіз аналогів

	Moodle	Excel	Пошта	EMark
Легкий пошук оцінок	+-	+	-	+
Отримання всіх оцінок по предмету	+	+	-	+
Легкість додавання оцінок	+-	+	+	+
Легкість вивчення програми	+-	+-	+	+
Наявність власного акаунту	+	+	+	+
Перегляд оцінок учнем	+	+	+	+
Керування з різним доступом	+	+	-	+
Створення власного акаунту	-	-	+	+
Керування власним акаунтом	-	-	+	+

3

## Мета, об'єкт та предмет дослідження

Мета роботи – покращення стану моніторингу успішності студентів.

Об'єкт дослідження – процес моніторингу успішності студентів.

Предмет дослідження – програмне забезпечення для моніторингу успішності студентів.

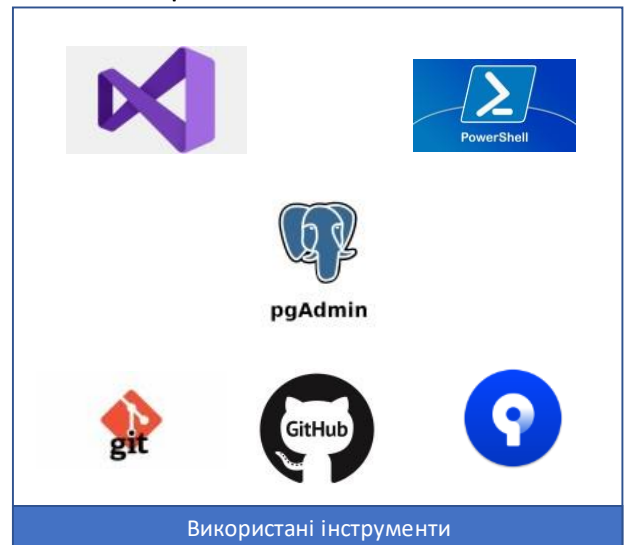
4

## Технічне завдання

1. Визначити архітектуру системи та її елементи.
2. Змодельювати структуру проекту.
3. Реалізувати наступний функціонал :
  - Створення та керування власним акаунтом;
  - Операції з групами;
  - Операції з предметами та оцінками.
4. Протестувати систему.

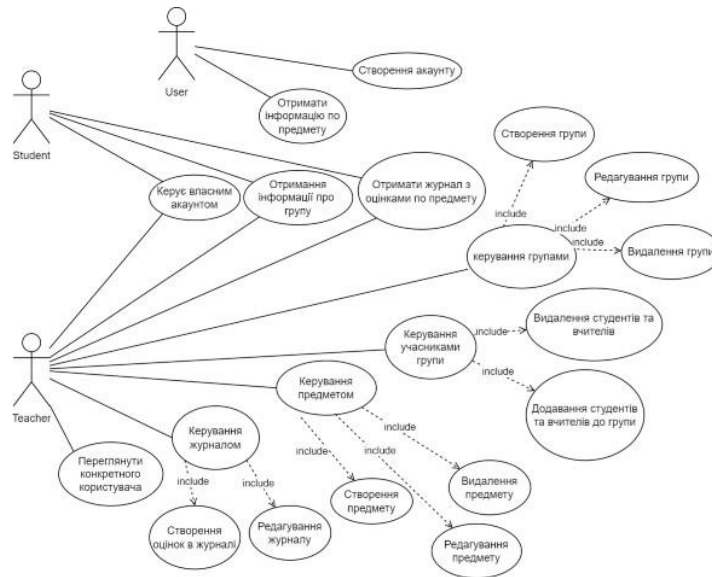
5

## Засоби реалізації



6

# Функціональність системи



7

## Моделювання діяльності перевірки користувача



Діаграма послідовності

8



## Приклад відповіді від API (отримання журналу оцінок)

GET /api/Subject/{subjectId}/journal

Name	Description
subjectId	Integer (51st32) (path)

4

Execute

Responses

```
200
Response body
{
  "name": "Math",
  "groupName": "PO-43",
  "teacherFullName": "Олександр Киричов Петрович",
  "columns": [
    {
      "name": "Тест",
      "marks": [
        {
          "value": 85,
          "studentFullName": "Максим Мельник Аркадійович"
        },
        {
          "value": 67,
          "studentFullName": "Ілля Кобиляцький Анатолійович"
        },
        {
          "value": 82,
          "studentFullName": "Влад Кравчук Сергійович"
        }
      ]
    }
  ],
  "name": "Контрольна",
  "marks": [
    {
      "value": 95,
      "studentFullName": "Максим Мельник Аркадійович"
    }
  ]
}

Response headers
content-length: 696
content-type: application/json; charset=utf-8
date: Tue31 May 2022 13:44:04 GMT
server: Kestrel
```

11

## Апробація результатів дослідження

- Кравчук В.С. Використання AutoMapper: Матеріали науково-технічної конференції “застосування програмного забезпечення в інфокомунікаційних технологіях”. Збірник тез. – ст. 8-9, 20.04.2022 м.Київ.
- Кравчук В.С. Використання jwt bearer авторизації для забезпечення безпеки: Матеріали науково-технічної конференції “застосування програмного забезпечення в інфокомунікаційних технологіях”. Збірник тез. – ст. 52-53, 20.04.2022 м.Київ.

12

## ВИСНОВОК

1. Проаналізовано існуючі аналоги.
2. Змодельовано структуру проекту.
3. Реалізовано необхідний функціонал зазначений в ТЗ.
4. Протестовано за допомогою Manual testing.

**Дякую за увагу**