

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
Факультет інформатики та обчислювальної техніки  
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії  
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ  
(підпис) (ім'я прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

## Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення  
комп'ютеризованих систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Програмне забезпечення для аналізу відгуків та оцінок послуг

Виконав студент IV курсу, групи ПП-82  
(шифр групи)

Приймак Ростислав Ярославович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник ст. викладач Халус О. А. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з графічної документації доцент, к.т.н., доц., Ліщук К. І. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доцент кафедри ОТ, к.т.н. Волокіта А. М. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ –2022

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення  
комп'ютеризованих систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ  
(підпис) (ім'я прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Приймаку Ростиславу Ярославовичу

(прізвище, ім'я, по батькові)

1. Тема проєкту Програмне забезпечення для аналізу відгуків та оцінок послуг

керівник проєкту Халус Олена Андріївна, ст. викладач

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

затверджені наказом по університету від «10» червня 2022 р. №1033-с

2. Термін подання студентом проєкту « 19 » червня 2022 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз предметної області, огляд існуючих рішень, розробка функціональних та нефункціональних вимог.

2) Моделювання та конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, архітектура програмного забезпечення, конструювання програмного забезпечення, аналіз безпеки даних.

3) Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ, опис процесів тестування, опис контрольного прикладу.

## 5. Перелік графічного матеріалу

1) Схема структурна варіантів використань \_\_\_\_\_

2) Схема бази даних \_\_\_\_\_

3) Схема структурна класів програмного забезпечення \_\_\_\_\_

## 6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «10» березня 2022 року \_\_\_\_\_

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2022	
2	Аналіз існуючих методів розв'язання задачі	20.03.2022	
3	Постановка та формалізація задачі	01.04.2022	
4	Розробка інформаційного забезпечення	20.04.2022	
5	Алгоритмізація задачі	01.05.2022	
6	Обґрунтування вибору використаних технічних засобів	15.05.2022	
7	Розробка програмного забезпечення	29.05.2022	
8	Налагодження програми	01.06.2022	
9	Виконання графічних документів	02.06.2022	
10	Оформлення пояснювальної записки	04.06.2022	
11	Подання ДП на попередній захист	06.06.2022	
12	Подання ДП рецензенту	12.06.2022	
13	Подання ДП на основний захист	19.06.2022	

Студент

\_\_\_\_\_

(підпис)

Ростислав ПРИЙМАК

\_\_\_\_\_

(ініціали, прізвище)

Керівник

\_\_\_\_\_

(підпис)

Олена ХАЛУС

\_\_\_\_\_

(ініціали, прізвище)

## АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з трьох розділів, містить 70 таблиць, 3 рисунка та 8 джерел – загалом 56 сторінок.

Дипломний проєкт присвячений аналізу відгуків та оцінок послуг через інтегрування із Google Maps із запуском розширення в браузері Google Chrome.

**Мета дипломного проєкту:** розробити програмне забезпечення, що дозволяє персоналізувати відгуки, оцінки на певні послуги за різними критеріями, фільтрами і комплексним обрахуванням загальної оцінки.

**Об'єкт дослідження:** оцінка якості послуг

**Предмет дослідження:** програмне забезпечення оцінки якості послуг

В першому розділі проведено аналіз предметної області відгуків та оцінок послуг. Розроблені та описані функціональні та нефункціональні вимоги.

В другому розділі проведено аналіз моделювання, конструювання та архітектури програмного забезпечення. Було описано деталі реалізації, використання сторонніх бібліотек та паттернів проектування.

В третьому розділі було проаналізовано якість програмного забезпечення та описано використанні підходи та технології тестування.

**КЛЮЧОВІ СЛОВА:** РОЗШИРЕННЯ GOOGLE CHROME, GOOGLE MAPS, БАЗА ДАНИХ, АНАЛІЗ ВІДГУКІВ, КОМПЛЕКСНИЙ РЕЙТИНГ.

## ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 70 tables, 3 figures and 8 sources – in total 56 pages.

The thesis project is dedicated to the analysis of reviews and evaluations of services through integration with Google Maps with the launch of the extension in Google Chrome.

The purpose of the diploma project: to develop software that allows you to personalize reviews, evaluations of certain services according to various criteria, filters and complex calculations of the overall evaluation.

**Object of research:** assessment of service quality

**Subject of research:** software for assessing the quality of services

The first section contains an analysis of the subject area of feedback and evaluation of services. Functional and non-functional requirements are developed and described.

The second section contains an analysis of the modeling, design, and architecture of the software. Details of implementation, use of third-party libraries, and design patterns were described.

The third section contains an analysis of the quality of the software and described the use of testing approaches and technologies.

**KEYWORDS:** GOOGLE CHROME EXTENSION, GOOGLE MAPS, DATABASE, ANALYSIS OF REVIEWS, COMPLEX RATING.



Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ВІДГУКІВ ТА  
ОЦІНОК ПОСЛУГ  
Технічне завдання  
КПІ.ПІ-8225.045440.01.91**

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олена ХАЛУС

Нормоконтроль:

\_\_\_\_\_ Катерина ЛШЦУК

Виконавець:

\_\_\_\_\_ Ростислав ПРИЙМАК

Київ – 2022

## ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ .....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ .....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ .....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	6
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності.....	6
4.3	Умови експлуатації .....	6
4.4	Вимоги до складу і параметрів технічних засобів.....	7
4.5	Вимоги до інформаційної та програмної сумісності.....	7
4.6	Вимоги до маркування та пакування .....	7
4.7	Вимоги до транспортування та зберігання.....	7
4.8	Спеціальні вимоги.....	7
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....	9
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ .....	11
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	12

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

# 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Програмне забезпечення для аналізу відгуків та оцінок послуг.

Галузь застосування:

Наведене технічне завдання поширюється на розробку програмного забезпечення «Програмне забезпечення для аналізу відгуків та оцінок послуг» КП.ІП-8225.045440.02.81. Дане програмне забезпечення направлене на використання у повсякденному житті користувачами для зручного вибору певної послуги чи місця покладаючись на оцінку чи відгуки із даного програмного забезпечення.

					КП.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «Програмне забезпечення для аналізу відгуків та оцінок послуг» є завдання на дипломне проєктування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для звичайних користувачів що хочуть персоналізувати інформацію щодо певного місця чи послуги.

Метою розробки є покращення існуючого способу вибору певного місця чи послуги за наявними відгуками.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1.1 Для користувача:

- надавати можливість створення нового об'єкту місця у випадку відсутності;
- надавати можливість збереження налаштувань у вигляді створення фільтру.
- надавати можливість перегляду фінальної оцінки за посиланням
- надавати можливість для реєстрації та авторизації
- надавати можливість перегляду існуючих відгуків
- надавати можливість включати та виключати фільтри
- надавати можливість редагування та видалення фільтрів
- надавати можливість додавання відгуку за посиланням

4.1.2 Розробку виконати на платформі Java та на операційних системах Windows, Linux, MacOS

#### 4.1.3 Додаткові вимоги:

- Необхідно забезпечити захищене зберігання персональних користувачів.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

## 4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних.

## 4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

## 4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати персональних комп'ютерах із Windows, Linux, MacOS операційними системами

4.4.2 Мінімальна конфігурація технічних засобів:

4.4.2.1 Об'єм ОЗП ..... 2 ГБ.

## 4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем Windows, Linux та MacOS.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі: введення тексту у спеціальні поля вводу.

4.5.3 Результати повинні бути представлені в наступному форматі: текст у відповідних полях у таблицях.

## 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

## 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

#### 4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

## 5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему.

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 50 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Програма та методика тестування

5.4 Графічна частина повинна бути виконана не менше ніж на 3 аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема структура варіантів використань.

5.4.2 Схема бази даних.

5.4.3 Схема структурна класів програмного забезпечення.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

## 6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	10.03	
2.	Розробка технічного завдання	25.03	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	05.04	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	20.04	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму)
5.	Програмна реалізація програмного забезпечення	29.05	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	30.06	Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту	01.06	Пояснювальна записка
8.	Розробка матеріалів графічної частини проекту	02.06	Графічний матеріал проекту
9.	Оформлення технічної документації проекту	04.06	Технічна документація

Змін.	Арк.	№ докум.	Підп.	Дата.

КПІ.ІП-8225.045440.01.91

Арк.

10

## 7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-8225.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		11

**Пояснювальна записка  
до дипломного проєкту**

на тему: Програмне забезпечення для аналізу відгуків та оцінок послуг

КПІ.ІП-8225.045440.02.81

Київ – 2022

## ЗМІСТ

<b>ВСТУП</b> .....	<b>3</b>
<b>1. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>4</b>
<b>1.1 ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	<b>4</b>
<b>1.2 АНАЛІЗ УСПІШНИХ ІТ-ПРОЄКТІВ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	<b>4</b>
<i>1.2.1 Аналіз існуючого рішення в середині Google Maps</i> .....	<i>4</i>
<i>1.2.2 Аналіз типових існуючих рішень на прикладі веб-сервісу Tripadvisor</i> .....	<i>5</i>
<b>1.3 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>5</b>
<i>1.3.1 Розроблення функціональних вимог</i> .....	<i>6</i>
<i>1.3.2 Розроблення нефункціональних вимог</i> .....	<i>11</i>
<b>1.4 ПОСТАНОВКА ЗАДАЧІ</b> .....	<b>11</b>
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	<b>12</b>
<b>2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>14</b>
<b>2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>14</b>
<b>2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>15</b>
<b>2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>16</b>
<b>2.4 ОПИС СХОВИЩА ДАНИХ</b> .....	<b>28</b>
<b>2.5 АНАЛІЗ БЕЗПЕКИ ДАНИХ</b> .....	<b>31</b>
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	<b>31</b>
<b>3. АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>32</b>
<b>3.1 АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	<b>32</b>
<b>3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ</b> .....	<b>32</b>
<b>3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ</b> .....	<b>47</b>
<b>ВИСНОВКИ ДО РОЗДІЛУ</b> .....	<b>55</b>
<b>ВИСНОВКИ</b> .....	<b>57</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ</b> .....	<b>58</b>

## ВСТУП

Сьогодні залишилось мало людей що покладаються лише на відгуки знайомих при виборі певного місця чи послуг. Із розвитком телекомунікацій виростає проблема довіри до представленої інформації, у цьому випадку до правдивості поставленої оцінки та залишеного відгуку.

В більшості існуючих рішень є підрахунок оцінки використовуючи середнє арифметичне. В певних випадках це не зовсім коректно, оскільки конкуренти можуть знижувати оцінку один одному, навіть ніколи не користувавшись якимось послугами.

Хороші відгуки що були залишені декілька років тому мають для нас меншу вагу ніж відгук залишений вчора. Для вирішення даної проблеми можна використати пріоритеризацію або комплексну оцінку

Також інколи хочеться яось відфільтрувати відгуки, наприклад якщо відгук негативний, але був створений користувачем, що зареєструвався сьогодні та створив лише один відгук. Інколи якість послуг сильно змінюється, але у відповідних додатках цього не буде видно.

Можна інтуїтивно пролистувати відгуки і відкидати не актуальні для нас, але це надто важко і потребує багато часу. До прикладу для вибору ресторану можна орієнтуватися на відгуки користувачів, що до цього створили чималу кількість відгуків відкинувши нових користувачів. З іншого боку при виборі шиномонтажу нам буде цікаво переглянути відгуки тільки на певний сезон включаючи відгуків усіх користувачів. Вирішення даної проблеми потребує налаштувань для кожного користувача, але нажаль існуючі системи не мають такої функції.

					КПІ.ІП-8225.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## 1. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 1.1 Опис і аналіз предметної області

Під час повсякденного життя ми часто стикаємось із проблемами що пов'язані із недовірою до джерел інформації в інтернеті. Через збільшення асоціальності, також і збільшився відсоток людей що використовують інтернет ресурси для вибору певних послуг, закладів тощо. До асоціалізації можна було скористуватися рекомендаціями знайомих. Далі розглянемо наявні сервіси для аналізу послуг та закладів. Можна було б провести детальний опис інших можливих сервісів, але більшість із них ідеологічно не відрізняються від tripadvisor, тому достатньо буде зробити лише його аналіз.

### 1.2 Аналіз успішних ІТ-проектів предметної області

#### 1.2.1 Аналіз існуючого рішення в середині Google Maps

Найбільш зручним та доступним наявним рішенням є Google Maps, оскільки окрім відгуків можна також поглянути розташування на карті, одразу вибрати собі маршрут тощо. Через низький поріг оцінок, їх не велику кількість, не актуальні відгуки, відгуки від щойно зареєстрованих користувачів та упереджені відгуки без об'єктивності ми отримуємо неочікуваний результат після візиту певного закладу чи послуг. До прикладу у закладу змінився власник, але старі відгуки щодо певного закладу залишились, звідси оцінка послуг отримує неактуальні дані і робить хибну фінальну оцінку. Для того щоб уникнути усіх цих проблем і вибрати заклад що найбільше тобі підходить – потрібно переглянути сторінку, пошукати негативні та позитивні коментарі, із акцентом на користувачів що їх поставили.

Якщо дозволяє предметна область – пошукати інші сервіси що надають відгуки по даній предметній області. Але тут виникає інша проблема – часто такі ресурси мають спонсорів, послуги і заклади яких можуть отримувати несправедливу оцінку. По-суті, це вищезгадана проблема

Також до недоліків даного рішення можна віднести те що оцінка вираховується лише за середнім арифметичним. Не можна не побачити не

					КПІ.ІП-8225.045440.02.81	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

зручний інтерфейс застосунку, що дозволяє лише витягувати дані про 5 відгуків, що практично унеможлиблює надбудову інших додатків із використанням бази даних існуючих відгуків. Звісно останню проблему можна вирішити за допомогою інтеграції з іншими сервісами що пропонують інтерфейс для витягування відгуків із Google Maps за певну платню по тривалості використання або кількості зроблених запитів. До таких сервісів можна віднести Outscraper, Також можна написати аналогічний парсер але це буде надлишковою роботою. Політика Google Maps не ставить в пріоритеті можливості для завантаження відгуків.

### **1.2.2 Аналіз типових існуючих рішень на прикладі веб-сервісу Tripadvisor**

Спеціально для огляду типового джерела можна розглянути сервіс tripadvisor. Для цього в пошуковнику можна набрати «Київ найкращі суши» – першим результатом окрім закладів харчування та доставки буде даний ресурс із своїм рейтингом «Топ 10 ресторанів». На першому місці стоїть ресторан із 144 відгуками, усі із рейтингом 5 з 5, звідси фінальний рейтинг - 5. Очевидно що така ситуація не реально, можливо для об'єктивності замало відгуків, але жодні послуги чи заклади не можуть сподобатися всім – що ставить під питання незалежність даного сервісу.

### **1.3 Аналіз вимог до програмного забезпечення**

Розроблене програмне забезпечення повинне частково або повністю вирішувати наступні проблеми:

- фільтрування відгуків для показу користувачу, та відгуків що беруться до уваги при розрахунку фінальної оцінки
- пріоритеризація відгуків в залежності від їх актуальності
- збереження налаштувань щодо фільтрування та пріоритеризації відгуків, обрахування фінальної оцінки

Також важливим є розробити програмне забезпечення що легко інтегрувати до існуючих рішення у Google Maps

					<b>КПІ.ІП-8225.045440.02.81</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

### 1.3.1 Розроблення функціональних вимог

Аналіз функціональних вимог легко зробити за схеми структурної варіантів використань враховуючи вищенаведений аналіз предмет області та поставлених задач. Дану схему відображено на листі 1, додатку Е. В наступних таблицях опишемо всі варіанти використань.

Таблиця 1.1 – Варіант використання UC001

Назва	Створення нового об'єкту місця у випадку його відсутності
Опис	Користувач перебуваючи на наведеному місці у веб-сервісі Google Maps вмикає додаток і натискає кнопку "Load place"
Учасники	Користувач розширення
Передумови	Користувач перейшов на сторінку із місцем що не було зафіксовано у розробленому веб-сервісі
Постумови	Користувач отримав інформацію про створений об'єкт і може побачити кількість відгуків та фінальну оцінку
Основний сценарій	Користувач розширення натиснув на кнопку та отримав початкові значення для об'єкту місця

Таблиця 1.2 – Варіант використання UC002

Назва	Створення фільтру для персоналізації відгуків та фінальної оцінки
Опис	Користувач має можливість створити фільтр для подальшого використання у персоналізації відгуків і фінальної оцінки
Учасники	Користувач розширення
Передумови	Авторизований користувач перейшов на будь-яку сторінку із місцем запустив розширення

## Продовження таблиці 1.2

Постумови	Користувач отримав інформацію про створений об'єкт фільтру і може побачити його у списку фільтрів, що були ним створенні
Основний сценарій	Користувач перебуваючи на наведеному місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з фільтрами та створе новий фільтр із введення певних даних

Таблиця 1.3 – Варіант використання UC003

Назва	Перегляд фінальної оцінки за посиланням
Опис	Користувач має можливість переглянути фінальну оцінку для будь-якого об'єкту незалежно від того чи користувач авторизований та попереднього об'єкту в системі
Учасники	Користувач розширення
Передумови	Авторизований користувач перейшов на будь-яку сторінку із місцем запуску розширення
Постумови	Користувач бачить фінальну оцінку по наведеному місцю
Основний сценарій	Користувач перебуваючи на наведеному місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку із рейтингом. У випадку авторизованого користувача із ввімкненим фільтром – на екрані повинна бути виведена персоналізована фінальна оцінка. У іншому випадку – фінальна оцінка у вигляді середнього арифметичного рейтингів наявних відгуків

Таблиця 1.4 – Варіант використання UC004

Назва	Реєстрація
Опис	Користувач має можливість створити аккаунт для подальшої персоналізації
Учасники	Користувач розширення
Передумови	Не автоізований користувач ввімкнув розширення
Постумови	Користувач створив аккаунт, який пізніше зможе використати для персоналізації авторизації
Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з реєстрацією, вводить поля пошти, ім'я користувача, мобільного телефона та пароль. Після реєстрації користувач перенаправляється у розділ із авторизацією

Таблиця 1.5 – Варіант використання UC005

Назва	Перегляд існуючих відгуків за посиланням
Опис	Користувач має можливість переглянути відгуки для наведеного місця
Учасники	Користувач розширення
Передумови	Користувач перейшов на визначенне місце та ввімкнув розширення
Постумови	Користувач отримав список створених відгуків
Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з відгуками. У випадку відсутності відгуків для наведеного місця – отримує порожній список. В іншому випадку – бачить відгуки із коментарем, датою, рейтингом, іменем користувача та його поштою

Таблиця 1.6 – Варіант використання UC006

Назва	Виключення та включення фільтру
Опис	Користувач має можливість у списку створених фільтрів увімкнути та вимкнути розширення
Учасники	Користувач розширення
Передумови	Авторизований користувач, що попередньо створив щонайменше один фільтр, перейшов на будь-яку сторінку із місцем та запустив розширення
Постумови	Користувач отримав оновлену інформацію про включений фільтр, якщо такий існує
Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з фільтрами та вимикає/вмикає фільтр. Після чого бачить зміни у статусах фільтрів. Також у випадку включення фільтру, коли інший був включений – фільтр що був включеним до цього, стає виключеним

Таблиця 1.7 – Варіант використання UC007

Назва	Редагування фільтрів для персоналізації
Опис	Користувач має можливість у списку створених фільтрів увімкнути та вимкнути розширення
Учасники	Користувач розширення
Передумови	Авторизований користувач, що попередньо створив щонайменше один фільтр, перейшов на будь-яку сторінку із місцем та запустив розширення
Постумови	Користувач отримав оновлену інформацію про включений фільтр після редагування

## Продовження таблиці 1.7

Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з фільтрами та міняє будь-який з параметрів. Після чого бачить оновлені дані на тій же вкладці
-------------------	--

Таблиця 1.8 – Варіант використання UC008

Назва	Авторизація для персоналізації
Опис	Користувач має можливість авторизуватися у попередньо створений аккаунт для подальшого використання у цілях персоналізації
Учасники	Користувач розширення
Передумови	Користувач попередньо зареєстрував аккаунт та пам'ятає ім'я користувача та пароль
Постумови	Користувач отримав доступ до вкладок що потребують авторизації
Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку з авторизацією. Вводить правильні дані, після чого отримує доступ до вкладки з фільтрами

Таблиця 1.9 – Варіант використання UC009

Назва	Додавання відгуку за посиланням
Опис	Користувач має можливість додати лише один відгук до будь-якого місця
Учасники	Користувач розширення
Передумови	Користувач авторизований
Постумови	Користувач бачить свій відгук у вкладці із відгуками та зміну фінальної оцінки якщо вона змінилася

## Продовження таблиці 1.9

Основний сценарій	Користувач перебуваючи на будь-якому місці у веб-сервісі Google Maps вмикає додаток і переходить на вкладку із створенням відгуку, вводить коментар, свій рейтинг. Після чого перенаправляється до списку відгуків
-------------------	--

**1.3.2 Розроблення нефункціональних вимог**

Нефункціональні вимоги зручно розділити по їх типах

Таблиця 1.10 – Нефункціональні вимоги

Тип	Список нефункціональних вимог
Продуктивність	Google Chrome розширення повинне запускатися менше 1 секунди
	Фільтрування відгуків повинне тривати менше 3 секунд
	Вирахування фінальної оцінки з використанням фільтру повинне тривати менше 20 секунд
Переносимість	Розроблене Google Chrome розширення має бути доступним у будь-якій операційній системі із встановленим та оновленим браузером Google Chrome
Зручність використання	Усі елементи Google Chrome розширення повинні бути інтуїтивно зрозумілими користувачу

**1.4 Постановка задачі**

**Призначення** даного дипломного проєкту полягає у покращенні існуючих алгоритмів та підходів до аналізу відгуків та оцінок послуг через інтеграцію із існуючим рішенням.

**Метою** даного дипломного проєкту є розробити програмне забезпечення, що дозволяє персоналізувати відгуки, оцінки на певні послуги за різними критеріями, фільтрами і комплексним обрахуванням загальної оцінки

**Завданням** проєкту є створення сервісу для персоналізації, фільтрації, пріоритеризації оцінки та відгуків на послуги, розширення у браузері Google-Chrome для інтеграції сервісу із базою даних закладів та послух в Google Maps.

Для досягнення поставлених цілей потрібно виконати наступні **задачі**:

- реєстрація та авторизація для персоналізації
- додавання відгуків із коментарями за посиланням
- перегляд існуючих відгуків із збереженою оцінкою
- створення та редагування збережених налаштувань у вигляді фільтрів
- вмикання та вимикання фільтрів
- застосування фільтрів до фільтрування відгуків та пріоритеризації

Головними **цілями** даного проєкту є полегшення та автоматизація фільтрування відгуків та підрахунку фінальної оцінки товару чи послуг.

### **Висновки до розділу**

Вибір послуг чи закладів залишається великою проблемою при збільшенні асоціалізації людей. Враховуючи що вищезгаданий процес не зупиниться потрібно знайти зручне рішення що зможе інтегруватися із конкурентами, використовуючи їхні бази даних місць.

В даному розділі було проведено аналіз предметної області відгуків та оцінок послуг. Переглянуто існуючі рішення поставленої задачі. Проведено аналіз відгуків та обрахування фінальної оцінки у системі Google Maps та інших типових аналогів.

Розроблено список функціональних вимог до пропонованого програмного забезпечення у вигляді діаграми прецедентів та розбору кожної умови у таблицях із послідовним описом умов виконання та очікуваних результатів.

Проведено аналіз основних нефункціональних вимог таких як швидкість запуску основної частини Google Chrome розширення та найскладніших алгортимів фільтрації відгуків та персоналізації фінальної

					<b>КПІ.ІП-8225.045440.02.81</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

оцінки по створеному фільтру, переносимість у вигляді запуску на будь-якій системі.

					КПІ.ІП-8225.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

## 2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Моделювання та аналіз програмного забезпечення

Для опису моделювання бізнес процесів програмного забезпечення найкраще використовувати діаграми BPMN.

Для того щоб завантажити відгуки для певного вибраного місця у веб-сервісі Google Maps потрібно відкрити розширення у вкладці із цим місцем. У випадку якщо користувач авторизований і має певний включений фільтр – відгуки будуть відфільтровані, в іншому випадку – користувач побачить останні додані. Весь процес описано на Рисунку 2.1

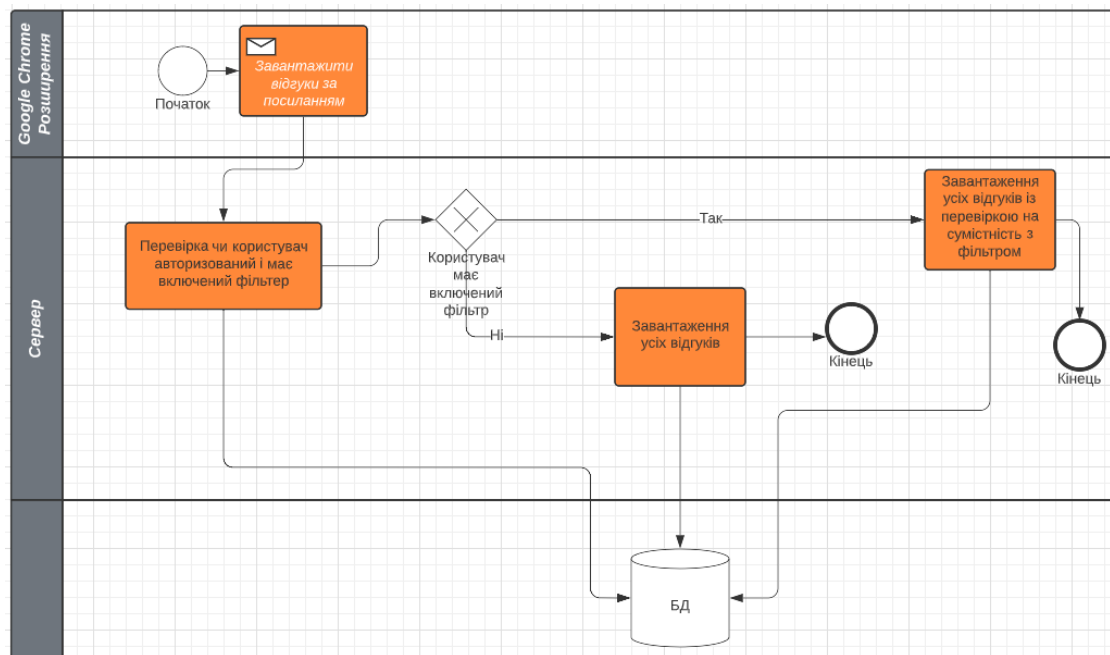


Рисунок 2.1 – Процес завантаження відгуків для певного місця

Для того щоб завантажити фінальну оцінку для певного вибраного місця у веб-сервісі Google Maps потрібно відкрити розширення у вкладці із цим місцем. У випадку якщо користувач авторизований і має певний включений фільтр – відгуки будуть відфільтровані і обраховані в залежності від налаштувань фільтру, в іншому випадку – користувач побачить фінальну оцінку що була збережена. Весь процес описано на Рисунку 2.2

Змн.	Арк.	№ докум.	Підпис	Дата

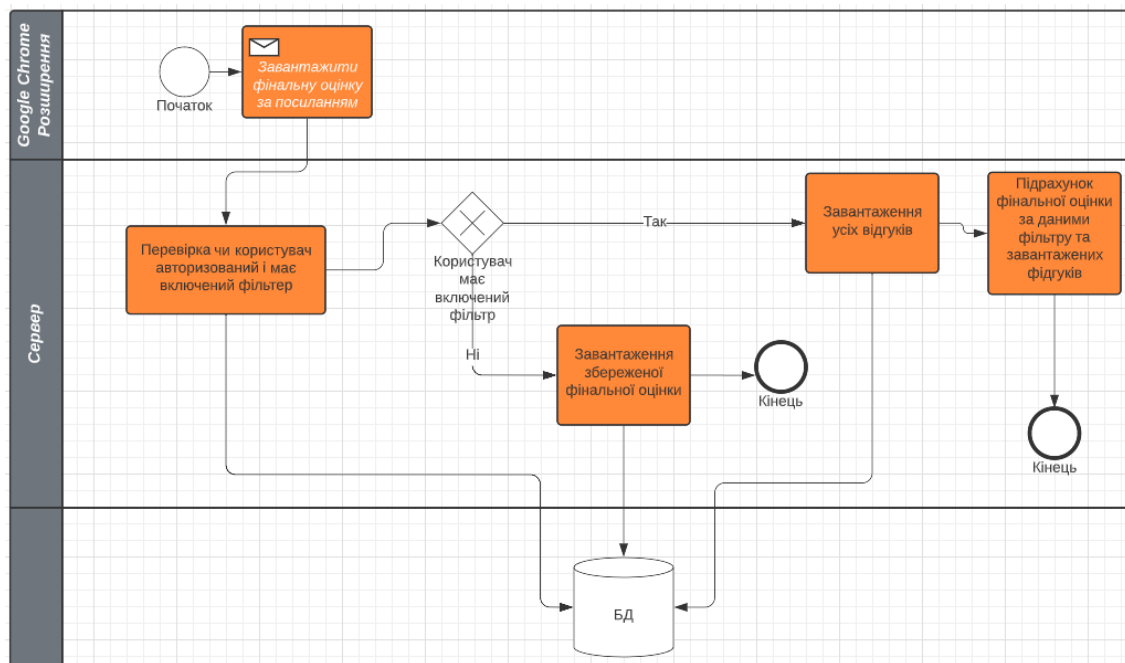


Рисунок 2.2 – Процес завантаження фінальної оцінки для певного місяця

## 2.2 Архітектура програмного забезпечення

Програмне забезпечення складається із клієнтської та серверної частини. Клієнтська частина зроблена у вигляді однієї сторінки як компонент бібліотеки vue. Вкладки переключаються за допомогою змінної що показує яку частину стоорінки показувати. Усі запити виконуються за допомогою бібліотеки axios.

Серверна частина реалізована за допомогою сімейства фреймворків Spring – Spring-boot, Spring Security, Spring-data-jpa. Також задіяно бібліотеки hibernate, mapstruct, lombok. Серверна частина поділена на наступні шари: controller, service, repository та базу даних. Перша виконує функцію фасаду та приймає всі запити. Service – виконує всю бізнес логіку. Repository представляє Dao паттерн. База даних реалізована на СКДБ PostgreSQL.

Для аналізу класів програмного забезпечення серверної частини було виділено основні класи та відображено на листі 1, додатку Є.

Усі запити приходять у класи контролерів що реалізують паттерн «Фасад». Далі щодо запитів виконується бізнес логіка в середині класів типу «Service». При необхідності вони взаємодіють між собою та використовуються класи типу «Repository», що з'єднують програмне забезпечення із базою даних. При цьому останні реалізують паттерн «DAO».

Змн.	Арк.	№ докум.	Підпис	Дата

### 2.3 Конструювання програмного забезпечення

При розробці серверної частини програмного забезпечення у вигляді Google Chrome розширення було створено наступні класи що наведенні у Таблиці 2.1. Опис методів наведених у Таблицях 2.2 – 2.18

Таблиця 2.1 – Опис моделі класів серверної частини

Клас	Опис
WebSecurityConfig	Конфігураційний клас наслідуваний від WebSecutyConfigurerAdapter із модуля Spring Security, що відповідає за налаштування Spring Security в проєкті включно із шифруванням паролів, поведінкою додатку при різних варіантів авторизації тощо.
FilterController	Клас що реалізовує паттерн проєктування «Фасад» та управляє створенням фільтру, редагування фільтрів, витягом усіх фільтрів для поточного авторизованого користувача. Усі методи класу вимагають авторизації користувача із будь-якою ролюю
PlaceController	Клас що реалізовує паттерн проєктування «Фасад» та управляє усіма можливими діями пов'язаними із сутністю місце, включаючи створення нової сутності та витягнення даних із уже існуючої

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 2.1

Клас	Опис
ReviewController	Клас що реалізовує паттерн проектування «Фасад» та управляє усіма можливими діями пов'язаними із сутністю оглядів, включаючи створення нової сутності та витягнення усіх попередньо створених сутностей
UserController	Клас що реалізовує паттерн проектування «Фасад» та управляє усіма можливими діями пов'язаними із сутністю користувача, включаючи створення нової сутності у вигляді реєстрації та витягнення даних про поточну авторизацію
FilterMapper	Клас що відповідає за перетворення сутності Filter у об'єкт FilterDto і навпаки. Використовує бібліотеку mapstruct
PlaceMapper	Клас що відповідає за перетворення сутності Place у об'єкт PlaceDto і навпаки. Використовує бібліотеку mapstruct
ReviewMapper	Клас що відповідає за перетворення сутності Review у об'єкт ReviewDto і навпаки. Використовує бібліотеку mapstruct
UserMapper	Клас що відповідає за перетворення сутності User у об'єкт UserDto і навпаки. Використовує бібліотеку mapstruct

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 2.1

Клас	Опис
FilterRepository	Клас що відповідає за усі дії, що пов'язані із сутністю Filter та базою даних. Наслідується від класу JpaRepository<T,ID> із бібліотеки spring-data-jpa
PlaceRepository	Клас що відповідає за усі дії, що пов'язані із сутністю Place та базою даних. Наслідується від класу JpaRepository<T,ID> із бібліотеки spring-data-jpa
ReviewRepository	Клас що відповідає за усі дії, що пов'язані із сутністю Review та базою даних. Наслідується від класу JpaRepository<T,ID> із бібліотеки spring-data-jpa
UserRepository	Клас що відповідає за усі дії, що пов'язані із сутністю User та базою даних. Наслідується від класу JpaRepository<T,ID> із бібліотеки spring-data-jpa
FilterService	Клас що відповідає за усі бізнес процеси, що пов'язані із сутністю Filter
PlaceService	Клас що відповідає за усі бізнес процеси, що пов'язані із сутністю Place
ReviewService	Клас що відповідає за усі бізнес процеси, що пов'язані із сутністю Review

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 2.1

Клас	Опис
UserService	Клас що відповідає за усі бізнес процеси, що пов'язані із сутністю User
DiplomaApplication	Клас, з якого апускається серверна частина із усіма модулями з бібліотеки spring-boot
LinkUtils	Клас що валідує вхідне посилання та видаляє непотрібні елементи із нього

Таблиця 2.2 – Опис методів класу WebSecurityConfig

Назва методу	Призначення
setUserService	Встановлення об'єкту UserService в поточний об'єкт уникаючи можливої циклічної залежності через конструктор класу
configure	Встановлення об'єкту UserService та PasswordEncoder у налаштування аутентифікації через можливості бібліотеки spring-security
configure	Налаштування сценаріїв поведінки авторизації, аутентифікації у випадках негативного і позитвного результату
failureHandler	Повертає об'єкт що реалізовує клас AuthenticationFailureHandler із бібліотеки spring-security із певною поведінкою управління невдалої авторизації користувача
successHandler	Повертає об'єкт що реалізовує клас AuthenticationSuccessHandler із бібліотеки spring-security із певною поведінкою управління вдалої авторизації користувача

## Продовження таблиці 2.2

Назва методу	Призначення
getPasswordEncoder	Повертає об'єкт класу PasswordEncoder із бібліотеки spring-security, що включає в себе налаштування шифрування паролів користувачів

Таблиця 2.3 – Опис методів класу класу FilterController

Назва методу	Призначення
create	Викликає створення об'єкту Filter у класу FilterService із застосуванням об'єкту FilterDto та повертає результат що прийшов із FilterService. Обидва вказані об'єкти, Filter та FilterDto, приходять у сервіс та виходять з нього у вигляді json файлів
update	Викликає редагування об'єкту Filter у класу FilterService із застосуванням об'єкту FilterDto та повертає результат що прийшов із FilterService. Обидва вказані об'єкти, Filter та FilterDto, приходять у сервіс та виходять з нього у вигляді json файлів
findAll	Викликає витягування усіх об'єктів Filter у класу FilterService та повертає результат що прийшов із FilterService. Результат у вигляді List<FilterDto> конвертується у json файл як масив
delete	Викликає видалення певного фільтру за унікальним ідентифікатором

Таблиця 2.4 – Опис методів класу класу PlaceController

Назва методу	Призначення
getDetails	Викликає створення пошук об'єкту за посиланням у класу PlaceService. Після чого повертає значення у вигляді PlaceDto що буде зконвертовано у json файл

Таблиця 2.5 – Опис методів класу класу ReviewController

Назва методу	Призначення
addReview	Викликає створення об'єкту Review у класу ReviewService із застосуванням об'єкту ReviewDto та повертає результат що прийшов із ReviewService. Обидва вказані об'єкти, Review та ReviewDto, приходять у сервіс та виходять з нього у вигляді json файлів
findAll	Викликає витягування усіх об'єктів Review у класу ReviewService та повертає результат що прийшов із ReviewService. Результат у вигляді List< ReviewDto> конвертується у json файл як масив. Також створює об'єкт Pageable для пагінації в залежності від параметрів запиту про номер сторінки

Таблиця 2.6 – Опис методів класу класу UserController

Назва методу	Призначення
saveNewUser	Викликає створення об'єкту User у класу UserService із застосуванням об'єкту UserDto та повертає результат що прийшов із UserService. Обидва вказані об'єкти, User та UserDto, приходять у сервіс та виходять з нього у вигляді json файлів

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 2.6

Назва методу	Призначення
getLogin	Повертає значення ролі для поточного авторизованого користувача. У випадку відсутності авторизації повертає значення «UNKNOWN»

Таблиця 2.7 – Опис методів класу FilterMapper

Назва методу	Призначення
to	Конвертує значення об'єкту FilterDto в об'єкт Filter враховуючи поля початкової та кінцевої дат. Використовує бібліотеку mapstruct
from	Конвертує значення об'єкту Filter в об'єкт FilterDto враховуючи поля початкової та кінцевої дат. Використовує бібліотеку mapstruct

Таблиця 2.8 – Опис методів класу PlaceMapper

Назва методу	Призначення
to	Конвертує значення об'єкту PlaceDto в об'єкт Place. Використовує бібліотеку mapstruct
from	Конвертує значення об'єкту Place в об'єкт PlaceDto. Використовує бібліотеку mapstruct
rating	Виконує заокруглення рейтингу до 2 символів після крапки

Таблиця 2.9 – Опис методів класу ReviewMapper

Назва методу	Призначення
to	Конвертує значення об'єкту ReviewDto в об'єкт Review враховуючи поле user, через викликання методу user. Використовує бібліотеку mapstruct
from	Конвертує значення об'єкту Review в об'єкт ReviewDto. Використовує бібліотеку mapstruct

## Продовження таблиці 2.9

Назва методу	Призначення
User	Конвертує значення об'єкту користувача User в об'єкт UserDto

Таблиця 2.9 – Опис методів класу класу UserMapper

Назва методу	Призначення
to	Конвертує значення об'єкту UserDto в об'єкт User. Використовує бібліотеку mapstruct
from	Конвертує значення об'єкту User в об'єкт UserDto. Використовує бібліотеку mapstruct

Таблиця 2.10 – Опис методів класу класу FilterRepository

Назва методу	Призначення
findAllByUser_UserId	Витягує з бази даних список усіх створених об'єктів класу Filter за унікальним ідентифікатором користувача. Використовує бібліотеки spring-data-jpa та hibernate
findAllByUser_UserIdAndName	Витягує з бази даних список усіх створених об'єктів класу Filter за унікальним ідентифікатором користувача та назвою. Використовує бібліотеки spring-data-jpa та hibernate
findAllByUser_UserIdAndEnabled	Витягує з бази даних список усіх створених об'єктів класу Filter за унікальним ідентифікатором користувача та станом включення фільтру. Використовує бібліотеки spring-data-jpa та hibernate

Таблиця 2.11 – Опис методів класу класу PlaceRepository

Назва методу	Призначення
findByLink	Витягує з бази даних у вигляді обгортки із класом Optional<T> об'єкт класу Place за посиланням. Використовує бібліотеки spring-data-jpa та hibernate

Таблиця 2.12 – Опис методів класу класу ReviewRepository

Назва методу	Призначення
findAllByDateAfterEndDateBeforeAndPlace_Id	Витягує з бази даних список усіх створених об'єктів класу Review за унікальним ідентифікатором місця та обмеженням по даті до та після. Використовує бібліотеки spring-data-jpa та hibernate
findAllByDateAfterEndDateBeforeAndPlace_Id	Витягує з бази даних список усіх створених об'єктів класу Review за унікальним ідентифікатором місця та обмеженням по даті до та після. Також приймає об'єкт для пагінації. Використовує бібліотке spring-data-jpa та hibernate

## Продовження таблиці 2.12

Назва методу	Призначення
findAllByPlace_Id	Витягує з бази даних список усіх створених об'єктів класу Review за унікальним ідентифікатором місця. Використовує бібліотеки spring-data-jpa та hibernate
findAllByPlace_Id	Витягує з бази даних список усіх створених об'єктів класу Review за унікальним ідентифікатором місця та об'єктом пагінації. Використовує бібліотеки spring-data-jpa та hibernate
countAllByUser_UserId	Рахує кількість об'єктів класу Review за унікальним ідентифікатором користувача. Використовує бібліотеки spring-data-jpa та hibernate

Таблиця 2.13 – Опис методів класу класу UserRepository

Назва методу	Призначення
findByUsername	Витягує з бази даних у вигляді обгортки із класом Optional<T> об'єкт класу User за унікальним ім'ям користувача. Використовує бібліотеки spring-data-jpa та hibernate

Таблиця 2.14 – Опис методів класу FilterService

Назва методу	Призначення
create	Створення нового об'єкту класу Filter та збереження його у базу даних із об'єкту класу FilterDto
validateName	Валідація назви фільтру на унікальність при створенні та редагуванні об'єкту класу Filter
findAll	Пошук усіх об'єктів класу Filter для авторизованого користувача
update	Редагування попередньо створеного об'єкту Filter
disableAnotherFilter	Виключення поточно включеного об'єкту класу Filter у випадку включення іншого об'єкту класу Filter
getEnabledFilter	Пошук включеного фільтру за унікальним ідентифікатором користувача
getEnabledFilter	Пошук включеного фільтру
delete	Видалення фільтру за ідентифікатором

Таблиця 2.15 – Опис методів класу PlaceService

Назва методу	Призначення
getPlace	Створення нового об'єкту класу Place у випадку відсутності такого у базі даних за унікальним посиланням
getByLink	Витягнення об'єкту класу Place за унікальним посиланням на місце в Google Maps

## Продовження таблиці 2.15

Назва методу	Призначення
reEvaluateRatingAfterReview	Оновлення фінальної оцінки що зберігається разом із сутністю після додавання нового огляду

Таблиця 2.16 – Опис методів класу ReviewService

Назва методу	Призначення
setPlaceService	Встановлення об'єкту PlaceService для уникнення циклічних залежностей через конструктор
Create	Створення нового об'єкту Review для авторизованого користувача
findAll	Пошук усіх створених об'єктів класу Review із пагінацією та можливим включеним фільтром
getCountByUser	Обрахування кількості відгуків створених певним користувачем за унікальним ідентифікатором користувача
isFitsFilter	Перевірка чи огляд підходить під умови певного фільтру
isMinReviewsFilterCheckFailed	Перевірка чи задовільняється умова мінімальної кількості відгуків для користувача
isEndDateCheckFailed	Перевірка чи відгук був створений до певної дати
isStartDateCheckFailed	Перевірка чи відгук був створений після певної дати

Змн.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.17 – Опис методів класу класу UserService

Назва методу	Призначення
saveNewUser	Створення нового об'єкту користувача під час реєстрації
loadUserByUsername	Завантаження об'єкту користувача за унікальним ім'ям користувача
getCurrentUser	Витягнення поточного авторизованого користувача із контексту

Таблиця 2.18 – Опис методів класу класу DiplomaApplication

Назва методу	Призначення
main	Запуск додатку із модулями бібліотеки spring-boot

## 2.4 Опис сховища даних

Було спроектовано схема бази даних та відображено на листі 1, додатку Ж. Моделі сховища бази даних описано у таблицях 2.1 - 2.5, база даних спроектована на СКДБ PostgreSQL

Таблиця 2.19 – Опис моделі users\_role

Назва	Тип	Опис
user_id	bigint	Унікальний ідентифікатор користувача
roles	varchar(255)	Список ролей для цього користувача

Таблиця 2.20 – Опис моделі users

Назва	Тип	Опис
user_id	bigint	Унікальний ідентифікатор користувача
active	boolean	Значення активності, актуальності аккаунту
email	varchar(255)	Пошта користувача

## Продовження таблиці 2.20

Назва	Тип	Опис
password	varchar(255)	Пароль користувача у зашифрованому вигляді
phone	varchar(255)	Мобільний телефон користувача
registration_date	date	Дата реєстрація користувача
username	varchar(255)	Ім'я користувача

Таблиця 2.21 – Опис моделі place

Назва	Тип	Опис
id	bigint	Унікальний ідентифікатор сутності місця
link	varchar(255)	Посилання на об'єкт у вигляді у вигляді посилання на вкладці з відкритим об'єктом
rates	bigint	Кількість оцінок
rating	Double precision	Фінальна оцінка у вигляді середнього арифметичного

Таблиця 2.22 – Опис моделі review

Назва	Тип	Опис
id	bigint	Унікальний ідентифікатор сутності огляду
comment	varchar(255)	Коментар що користувач додав до огляду
date	date	Дата створення огляду на місце користувачем
place_id	bigint	Унікальний ідентифікатор сутності місця
user_id	bigint	Унікальний ідентифікатор сутності користувача

Таблиця 2.23 – Опис моделі filter

Назва	Тип	Опис
id	bigint	Унікальний ідентифікатор сутності фільтру
end_date	date	Дата після якої усі відгуки не беруться до уваги при фільтруванні та обрахуванні фінальної оцінки
min_review_for_user	bigint	Мінімальна кількість відгуків для користувача у випадку не задовільності усі відгуки не беруться до уваги при фільтруванні та обрахуванні фінальної оцінки
start_date	date	Дата перед якою усі відгуки не беруться до уваги при фільтруванні та обрахуванні фінальної оцінки
user_id	bigint	Унікальний ідентифікатор сутності користувача
discount	Double precision	Коефіцієнт що використовується для обрахування фінальної оцінки шляхом пріоритеризації старіших або новіших відгуків
enabled	boolean	Значення що показує чи включений даний фільтр для користувача
name	varchar(255)	Унікальна назва фільтру для користувача
use_discount	boolean	Поле, що показує чи включення в цьому фільтрі функція пріоритеризації

Змн.	Арк.	№ докум.	Підпис	Дата

## 2.5 Аналіз безпеки даних

Враховуючи застосування додатку можна допустити невелике незтування безпекою даних – до прикладу не зашровані дані про мобільний телефон на електронну пошту. Також прототип програмного забезпечення не використовує захищений прокол з'єднання між клієнтською та серверними частинами.

Пароль користувача шифрується за допомогою алгоритму bcrypt, що є хорошим, але не найкращим рішенням для зберігання паролів користувача, але це можна опустити по вищезгаданій причині.

### Висновки до розділу

У даному розділі було проведено аналіз моделювання, конструювання та архітектури програмного забезпечення. Для моделювання було використано BPMN діаграми основних бізнес процесів, а саме завантаження відгуків і обрахування фінальної оцінки місця із врахуванням можливого включеного фільтру.

Було коротко описано архітектуру всього програмного забезпечення включаючи клієнтську та серверну частини. Також використані технології та бібліотеки з фреймворками. Було додано діаграму основних класів серверної частини.

Проаналізовано усі класи що були створені для даного програмного забезпечення із коротким описом кожного методу.

Було проведено аналіз діаграми бази даних з повним описом кожної таблиці та поля, включаючи призначення та тип.

Звернуто увагу на безпеку даних у розробленому програмному забезпеченні

					КПІ.ІП-8225.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

### 3. АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Аналіз якості програмного забезпечення

Аналіз якості та тестування програмного забезпечення є невідкладною складовою розробки програмного забезпечення. Найкращим способом проаналізувати якість програмного забезпечення є написання тестів, що виконується комп'ютером та не потребують ніяких дій з боку людини. Сюди входять інтеграційні тести, що взаємодіють із інтерфейсом користувача, що запускають окремі частини, до прикладу серверну частину в ізольованому середовищі та модульні тести, що перевіряють справність кожного компонента незалежно від інших. При розробці будь-якого програмного забезпечення не можливо уникнути «мануальних» тестів, що виконуються вручну певною людиною.

В рамках розробки даного проекту було використано 2 типи тестування: модульне для серверно частини та «мануальне».

#### 3.2 Опис процесів тестування

Для проведення модульного тестування було використано найбільш популярні та найзручніші інструменти для мови програмування Java, а саме JUnit 5 та Mockito. Де перший є так званим тест-провайдером, а другий механізм для використання та менеджменту «моками» - обманками.

В Таблицях 3.1 – 3.25 наведенні тест плани що були покриті модульними тестами в середині серверної частини.

Таблиця 3.1 – TC001

Мета тесту	Перевірка чи при відсутності унікальності ім'я користувача ми отримаємо помилку
Початковий стан	Попередньо створений користувач із таким ім'ям

## Продовження таблиці 3.1

Вхідні дані	Сутність користувача із ім'ям, що попередньо збережено для іншого користувача
Схема проведення тесту	Ставимо обманку кидання помилки при збереженні нового користувача. Створюємо сутність користувача із не унікальним ім'ям. Викликаємо збереження
Очікуваний результат	Помилка із текстом «User already exists»
Стан програмного продукту після проведення випробувань	Користувач не був створений

## Таблиця 3.2 – TC002

Мета тесту	Перевірка чи створиться користувач із унікальним ім'ям
Початковий стан	Користувача із таким ім'ям не створено
Вхідні дані	Сутність користувача із ім'ям, що попередньо не збережено для іншого користувача
Схема проведення тесту	Ставимо обманку при збереженні нового користувача. Створюємо сутність користувача із унікальним ім'ям. Викликаємо збереження
Очікуваний результат	Отримуємо створену сутність користувача
Стан програмного продукту після проведення випробувань	Користувач був створений

Таблиця 3.3 – ТС003

Мета тесту	Перевірка чи повернеться сутність користувача якщо користувач існує
Початковий стан	Попередньо створений користувач із таким ім'ям
Вхідні дані	Ім'я користувача
Схема проведення тесту	Ставимо обманку при витягненні нового користувача, що буде повертати певну сутність користувача. Викликаємо витягнення
Очікуваний результат	Отримуємо сутність користувача
Стан програмного продукту після проведення випробувань	Стан користувача не змінився

Таблиця 3.4 – ТС004

Мета тесту	Перевірка чи повернеться сутність користувача якщо користувач не існує
Початковий стан	Користувач із таким ім'ям не створений
Вхідні дані	Ім'я користувача
Схема проведення тесту	Ставимо обманку при витягненні нового користувача, що буде повертати порожнє значення. Викликаємо витягнення
Очікуваний результат	Отримуємо порожнє значення

## Продовження таблиці 3.4

Стан програмного продукту після проведення випробувань	Стан користувачів не змінився
--	-------------------------------

Таблиця 3.5 – TC005

Мета тесту	Перевірка на правильність посилання на певне місце із Google Maps та перетворення посилання
Початковий стан	
Вхідні дані	Правильне посилання із сторінки Google Maps
Схема проведення тесту	Надсилаємо посилання
Очікуваний результат	Отримуємо посилання без позначки про мову в кінці
Стан програмного продукту після проведення випробувань	

Таблиця 3.6 – TC006

Мета тесту	Перевірка спрацювання перерахунку рейтингу місця після створення нового відгуку для цього місця
Початковий стан	Попередньо створена сутність місця
Вхідні дані	Відгук із позначкою місця
Схема проведення тесту	Створюємо обманки для користувача та місця. Викликаємо створення відгуку. Перевіряємо виклик метода про переразунк рейтинг

## Продовження таблиці 3.6

Очікуваний результат	Отримуємо сутність створеного відгуку
Стан програмного продукту після проведення випробувань	Рейтинг місця змінився враховуючи рейтинг із доданого відгуку за середнім арифметичним

## Таблиця 3.7 – TC007

Мета тесту	Перевірка пошуку відгуків із виключеним фільтром
Початковий стан	Фільтр для користувача відсутній або користувач не авторизований
Вхідні дані	Унікальний ідентифікатор місця та об'єкт для пагінації
Схема проведення тесту	Створюємо обманки для оглядів. Викликаємо витягнення оглядів. Перевіряємо кількість оглядів
Очікуваний результат	Не отримуємо оглядів
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

## Таблиця 3.8 – TC008

Мета тесту	Перевірка чи буде огляд відфільтрований якщо користувач авторизований та має включений фільтр, що відкине значення огляду
Початковий стан	Користувач авторизований та фільтр включений для нього

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 3.8

Вхідні дані	Унікальний ідентифікатор місця та об'єкт для пагінації
Схема проведення тесту	Створюємо обманки для оглядів та включеного фільтру. Викликаємо витягнення оглядів. Перевіряємо кількість оглядів
Очікуваний результат	Не отримуємо оглядів
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

Таблиця 3.9 – TC009

Мета тесту	Перевірка пошуку відгуків
Початковий стан	Відгуки для цього місця відсутні
Вхідні дані	Унікальний ідентифікатор місця
Схема проведення тесту	Створюємо обманки для оглядів. Викликаємо витягнення оглядів. Перевіряємо кількість оглядів
Очікуваний результат	Не отримуємо оглядів
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

Таблиця 3.10 – TC010

Мета тесту	Перевірка підрахунку кількості відгуків у користавача
------------	---

Продовження таблиці 3.10

Початковий стан	Створений один відгук для користувача
Вхідні дані	Унікальний ідентифікатор користувача
Схема проведення тесту	Створюємо обманки для кількості оглядів. Викликаємо підрахунок оглядів. Перевіряємо кількість оглядів
Очікуваний результат	Отримуємо один огляд
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

Таблиця 3.11 – TC011

Мета тесту	Перевірка фільтра коли дата створення відгуку раніше нижньої межі фільтра
Початковий стан	Створений один відгук для користувача та фільтр
Вхідні дані	Створений фільтр та відгук із датою раніше нижньої межі
Схема проведення тесту	Викликаємо метод перевірки сумісності із фільтром. Перевіряємо результат
Очікуваний результат	Отримуємо негативний результат
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

Таблиця 3.12 – TC012

Мета тесту	Перевірка фільтру коли дата створення відгуку пізніше верхньої межі фільтра
Початковий стан	Створений один відгук для користувача та фільтр
Вхідні дані	Створений фільтр та відгук із датою пізніше верхньої межі
Схема проведення тесту	Викладаємо метод перевірки сумістності із фільтром. Перевіряємо результат
Очікуваний результат	Отримуємо негативний результат
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

Таблиця 3.13 – TC013

Мета тесту	Перевірка фільтру коли кількість оглядів у користувача менша за відповідний параметр фільтру
Початковий стан	Створений один відгук для користувача та фільтр
Вхідні дані	Створений фільтр із мінімальною кількістю відгуків для користувача та відгук з користувачем що має менше відгуків ніж потрібно для сумістності із фільтром
Схема проведення тесту	Ставимо обманку на підрахунок кількості створених відгуків для користувача. Викладаємо метод перевірки сумістності із фільтром. Перевіряємо результат

Змн.	Арк.	№ докум.	Підпис	Дата

## Продовження таблиці 3.13

Очікуваний результат	Отримуємо негативний результат
Стан програмного продукту після проведення випробувань	Стан оглядів не змінився

## Таблиця 3.14 – TC014

Мета тесту	Перевірка чи при виключеному фільтрі користувача отримуємо дані про місце із рейтингом що збережений у базі даних
Початковий стан	Створена сутність місця із одним відгуком
Вхідні дані	Посилання на місце у Google Maps
Схема проведення тесту	Ставимо обманку на включений фільтр щоб поверталось пuste значення. Виклакаємо метод витягнення місця. Перевіряємо результат
Очікуваний результат	Отримуємо об'єкт місця із фінальною оцінку що дорівнює значенню із бази даних
Стан програмного продукту після проведення випробувань	Стан місць не змінився

## Таблиця 3.15 – TC015

Мета тесту	Перевірка чи при включеному фільтрі отримуємо дані про місце із рейтингом що були параметризовані за фільтром
------------	---

Продовження таблиці 3.15

Початковий стан	Створена сутність місця із одним відгуком та користувач авторизований і має включений фільтр
Вхідні дані	Посилання на місце у Google Maps
Схема проведення тесту	Ставимо обманку на включений фільтр. Виклакаємо метод витягнення місця. Перевіряємо результат
Очікуваний результат	Отримуємо об'єкт місця із фінальною оцінку що не дорівнює значенню із бази даних
Стан програмного продукту після проведення випробувань	Стан місць не змінився

Таблиця 3.16 – TC016

Мета тесту	Перевірка чи при витягненні місця за посилання повертається попередньо створене місце
Початковий стан	Створена сутність місця
Вхідні дані	Посилання на місце у Google Maps
Схема проведення тесту	Ставимо обманку на попередньо створене місце. Виклакаємо метод витягнення місця. Перевіряємо результат
Очікуваний результат	Отримуємо об'єкт місця із даними що були збережені у базі даних
Стан програмного продукту після проведення випробувань	Стан місць не змінився

Таблиця 3.17 – TC017

Мета тесту	Перевірка чи при витягнені місця за посилання повертається щойно створене місце, якщо місце попередньо не було створене
Початковий стан	Сутність місця за посиланням відсутня
Вхідні дані	Посилання на місце у Google Maps
Схема проведення тесту	Ставимо обманку на попередньо створене місце щоб поверталось пусте значення. Виклакаємо метод витягнення місця. Перевіряємо результат
Очікуваний результат	Отримуємо об'єкт місця із початковими значеннями
Стан програмного продукту після проведення випробувань	Створено об'єкт місця

Таблиця 3.18 – TC018

Мета тесту	Перевірка чи при додаванні нового огляду рейтинг що збережений у базі даних перераховується
Початковий стан	Створена сутність місця
Вхідні дані	Створений відгук із посиланням на створене місце
Схема проведення тесту	Ставимо обманку на збереження сутності місця. Виклакаємо метод перерахунку рейтингу. Перевіряємо результат
Очікуваний результат	Не отримуємо результат

## Продовження таблиці 3.18

Стан програмного продукту після проведення випробувань	Об'єкт місця змінив стан рейтингу у базі даних
--	--

## Таблиця 3.19 – ТС019

Мета тесту	Перевірка чи отримаємо помилку при створенні фільтра із не унікальним ім'ям
Початковий стан	Попередньо створений фільтр із даним ім'ям
Вхідні дані	Створена сутність фільтра користувачем
Схема проведення тесту	Ставимо обманку на попередньо створений фільтр. Виклакаємо метод збереження фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо помилку із текстом «filter name not unique for user»
Стан програмного продукту після проведення випробувань	Об'єкт фільтру не був створений у базі даних

## Таблиця 3.20 – ТС020

Мета тесту	Перевірка чи створиться фільтр з унікальним ім'ям
Початковий стан	Немає попередньо створеного фільтра із даним ім'ям
Вхідні дані	Створена сутність фільтра користувачем

Продовження таблиці 3.20

Схема проведення тесту	Ставимо обманку на попередньо створений фільтр щоб поверталось порожнє значення. Виклакаємо метод збереження фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо створену сутність нового фільтру
Стан програмного продукту після проведення випробувань	Об'єкт фільтру був створений у базі даних

Таблиця 3.21 – TC021

Мета тесту	Перевірка на витягнення усіх фільтрів для користувача
Початковий стан	Попередньо створений один фільтр
Вхідні дані	
Схема проведення тесту	Ставимо обманку на попередньо створений фільтр. Виклакаємо метод витягнення фільтрів. Перевіряємо результат
Очікуваний результат	Отримуємо список фільтрів що містить лише сутність одного фільтру
Стан програмного продукту після проведення випробувань	Стан не змінився

Таблиця 3.22 – TC022

Мета тесту	Перевірка чи при редагуванні неіснуючого фільтру отримаємо помилку
------------	--

Продовження таблиці 3.22

Початковий стан	Попередньо фільтру із таким ідентифікатором не створено
Вхідні дані	Сутність фільтру із редагованими полями та ідентифікатором
Схема проведення тесту	Ставимо обманку на попередньо створений фільтр. Виклакаємо метод редагування фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо помилку із текстом «filter not found»
Стан програмного продукту після проведення випробувань	Стан не змінився

Таблиця 3.23 – TC023

Мета тесту	Перевірити чи при включені одного фільтру інший виключиться
Початковий стан	Попередньо створений включений фільтр для певного користувача
Вхідні дані	Сутність фільтру для збереження об'єкту у базі даних
Схема проведення тесту	Ставимо обманку на попередньо створений фільтр. Виклакаємо метод створення фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо створений фільтр із паратетром що показує що він включений

## Продовження таблиці 3.23

Стан програмного продукту після проведення випробувань	Попередньо створений фільтр став виключеним
--	---

Таблиця 3.24 – TC024

Мета тесту	Перевірити чи для не авторизованого користувача є включений фільтр
Початковий стан	Користувач не авторизований
Вхідні дані	
Схема проведення тесту	Ставимо обманку на витягнення авторизації користувача. Виклакаємо метод пошуку включеного фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо порожній результат
Стан програмного продукту після проведення випробувань	Стан не змінився

Таблиця 3.25 – TC025

Мета тесту	Перевірити чи видаляється фільтр
Початковий стан	Користувач авторизований
Вхідні дані	Унікальний ідентифікатор фільтру
Схема проведення тесту	Виклакаємо метод видалення фільтру. Перевіряємо результат
Очікуваний результат	Отримуємо порожній результат

## Продовження таблиці 3.25

Стан програмного продукту після проведення випробувань	Фільтр був видалений із бази даних
--	------------------------------------

Також було проведено аналіз покриття коду тестами із результатом на Рисунок 3.1

The screenshot shows a code coverage tool interface. At the top, it says 'Coverage: kpi.ua.diploma in diploma'. Below that, there's a table with the following data:

Element	Class, %	Method, %	Line, %
kpi.ua.diploma	67% (19/28)	68% (101/148)	64% (223/346)

Рисунок 3.1 – Результат аналізу покриття коду модульними тестами

Якщо взяти найменшим показник у 64%, що є достатньо малим показником в загальному, можна сказати що код покритий достатньо. Невеликий показник можна пояснити тим що проект не має багато коду на серверній частині, містить багато коду що потрібно покривати комплексними інтеграційними тестами що в даному випадку не є доцільним. Усі бізнес логіка серверної частина покрита модульними тестами

### 3.3 Опис контрольного прикладу

В якості контрольного прикладу було перевірено весь основний функціонал програмного забезпечення. Варіанти використання у виглядів тест планів:

- завантаження нового місця у системі(Таблиця 3.26);
- реєстрація користувача(Таблиця 3.27);
- авторизація користувача(Таблиця 3.28);
- додавання відгуку(Таблиця 3.29);
- перегляд відгуків(Таблиця 3.30);

					КПІ.ІП-8225.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

- створення фільтру(Таблиця 3.31);
- перегляд створених фільтрів(Таблиця 3.32);
- редагування фільтрів(Таблиця 3.33);
- видалення фільтрів(Таблиця 3.34);
- включення та виключення фільтрів(Таблиця 3.35);
- фільтрування відгуків за включеним фільтром(Таблиця 3.36);
- параметризація рейтингів відгуків за включеним фільтром(Таблиця 3.37);

Таблиця 3.26 – TC026

Мета тесту	Завантаження нового місця у системі
Початковий стан	Користувач перейшов на місце що не було попередньо збережене у системі
Вхідні дані	Посилання на місце у системі Google Maps
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення
Очікуваний результат	Отримуємо місце без відгуків із написом «Not reviewed before»
Стан програмного продукту після проведення випробувань	У базі даних створився об'єкт місця що відповідає відкритій сторінці

Таблиця 3.27 – TC027

Мета тесту	Реєстрація користувача
Початковий стан	Користувач перейшов на будь-яке місце та не авторизований

Продовження таблиці 3.27

Вхідні дані	Дані користувача для реєстрації
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Sign up». Вводимо дані користувача і натискаємо кнопку «Submit»
Очікуваний результат	Перенаправляємось на сторінку із авторизацією
Стан програмного продукту після проведення випробувань	У базі даних створився об'єкт користувача із введеними даними та зашифрованим паролем

Таблиця 3.28 – TC028

Мета тесту	Авторизація користувача
Початковий стан	Користувач перейшов на будь-яке місце та не авторизований
Вхідні дані	Дані користувача для авторизації
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Sign in». Вводимо дані користувача і натискаємо кнопку «Submit»
Очікуваний результат	Перенаправляємось на сторінку із переглядом відгуків
Стан програмного продукту після проведення випробувань	Стан бази даних не змінився. У браузері та на серверній частині створилась сесія для користувача

Таблиця 3.29 – TC029

Мета тесту	Додавання відгуку
Початковий стан	Користувач перейшов на будь-яке місце та авторизований
Вхідні дані	Дані що користувач бажає додати у відгук
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Add review». Вводимо рейтинг та коментар і натискаємо кнопку «Add»
Очікуваний результат	Перенаправляємось на сторінку із переглядом відгуків
Стан програмного продукту після проведення випробувань	У базі створився відгук та рейтинг місця змінився відповідно до оцінки із відгуку

Таблиця 3.30 – TC030

Мета тесту	Перегляд відгуків
Початковий стан	Користувач перейшов на будь-яке місце
Вхідні дані	
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Reviews»
Очікуваний результат	Бачимо максимум 5 попередньо доданих відгуків та фінальну оцінку місця

## Продовження таблиці 3.30

Стан програмного продукту після проведення випробувань	Відсутні зміни
--	----------------

Таблиця 3.31 – TC031

Мета тесту	Створення фільтру
Початковий стан	Авторизований користувач перейшов на будь-яке місце
Вхідні дані	Назва фільтру, початкова та кінцеві дати, мінімальна кількість відгуків для користувача, значення знижки, включення фільтру та знижки
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Add filter». Вводимо дані для фільтру
Очікуваний результат	Перенаправляємося у вкладку «Filters» і бачимо створений фільтр
Стан програмного продукту після проведення випробувань	Фільтр був створений у базі даних. У випадку якщо даний фільтр потрібно включити, інший фільтр, що був до цього включеним, стане виключеним

Таблиця 3.32 – TC032

Мета тесту	Перегляд створених фільтрів
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має створені фільтри
Вхідні дані	

## Продовження таблиці 3.32

Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Filters»
Очікуваний результат	Бачимо усі створені фільтри, якщо фільтр включений – обведений зеленим кольором
Стан програмного продукту після проведення випробувань	Стан не змінився

Таблиця 3.33 – TC033

Мета тесту	Редагування фільтрів
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має створені фільтри
Вхідні дані	
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Filters». Натискаємо на будь-яку кнопку із назвою фільтру. Змінюємо назву фільтру. Натискаємо кнопку «Save»
Очікуваний результат	Виключення редагування. На екрані вкладка із фільтрами, де оновлена назва фільтру
Стан програмного продукту після проведення випробувань	Назва фільтру змінилась у базі даних

Таблиця 3.34 – TC034

Мета тесту	Видалення фільтрів
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має створені фільтри
Вхідні дані	
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Filters». Натискаємо на будь-яку кнопку із назвою фільтру. Натискаємо кнопку «Delete»
Очікуваний результат	Виключення редагування. На екрані вкладка із фільтрами, де немає фільтру що ми видалили
Стан програмного продукту після проведення випробувань	Сутність фільтру видалена із бази даних

Таблиця 3.35 – TC035

Мета тесту	Включення та виключення фільтрів
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має щонайменше 2 створені фільтри, один з яких включений. Назвемо його фільтр 1.
Вхідні дані	

## Продовження таблиці 3.35

Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Filters». Натискаємо на будь-яку кнопку із назвою фільтру що має сірий колір. Включаємо фільтр. Натискаємо кнопку «Save»
Очікуваний результат	Виключення редагування. На екрані вкладка із фільтрами, де оновлена фільтр 1 сірий, а фільтр 2 - зелений
Стан програмного продукту після проведення випробувань	Значення щодо включення та виключення фільтрів 1 та 2 змінились

## Таблиця 3.36 – TC036

Мета тесту	Фільтрування відгуків за включеним фільтром
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має включений фільтр із початковою датою сьогодні. Також для цього місця є два відгуки, що були створені вчора та сьогодні
Вхідні дані	
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Reviews».
Очікуваний результат	Фінальна оцінка дорівнює відгуку створеному сьогодні

## Продовження таблиці 3.36

Стан програмного продукту після проведення випробувань	Змін стану не відбулось
--	-------------------------

Таблиця 3.37 – TC037

Мета тесту	Параметризація рейтингів відгуків за включеним фільтром
Початковий стан	Авторизований користувач перейшов на будь-яке місце та має включений фільтр із знижкою що дорівнює одиниці. Для цього місця є щонайменше 2 відгуки
Вхідні дані	
Схема проведення тесту	Відкриваємо певне місце в Google Maps. Відкриємо розроблене розширення. Переходимо у вкладку із написом «Reviews».
Очікуваний результат	Фінальна оцінка дорівнює першому створеному відгуку
Стан програмного продукту після проведення випробувань	Змін стану не відбулось

**Висновки до розділу**

Аналіз якості та тестування програмного забезпечення стали невід'ємною частиною розробки програмного забезпечення, тому надзвичайно важливо робити все вчасно та автоматизовано де це можливо.

В даному розділі було проаналізовано якість програмного забезпечення та описано використанні підходи та технології тестування. Описано 25 тест планів

					<b>КПІ.ІП-8225.045440.02.81</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

для модульного тестування із вхідними даними та станами та схемою проведення.

Також було описано тест плани для контрольного прикладу у вигляді «мануального» тестування із використанням повного запуску додатку.

					КПІ.ІП-8225.045440.02.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

## ВИСНОВКИ

В рамках даного дипломного проекту було розроблено програмне забезпечення для аналізу відгуків та оцінок послуг.

Спочатку було проведено аналіз предметної області відгуків та оцінок послуг. Переглянуто існуючі рішення поставленої задачі. Проведено аналіз відгуків та обрахування фінальної оцінки у системі Google Maps та інших типових аналогів. Розроблено список функціональних вимог до пропонованого програмного забезпечення у вигляді діаграми. Проведено аналіз основних нефункціональних вимог.

Наступним було проведено аналіз моделювання, конструювання та архітектури програмного забезпечення. Для моделювання було використано BPMN діаграми основних бізнес процесів. Було коротко описано архітектуру всього програмного забезпечення, використані технології та бібліотеки. Було додано діаграму основних класів серверної частини, після чого проаналізовано усі класи що були створені для даного програмного забезпечення. Також було проведено аналіз діаграми бази даних з повним описом кожної таблиці та поля, включаючи призначення та тип. Звернуто увагу на безпеку даних у розробленому програмному забезпеченні.

Після розробки було проаналізовано якість програмного забезпечення та описано використанні підходи та технології тестування. Описано 25 тест планів для модульного тестування із вхідними даними та станами та схемою проведення. Також було описано тест плани для контрольного прикладу у вигляді «мануального» тестування із використанням повного запуску додатку.

					<b>КПІ.ІП-8225.045440.02.81</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

## ПЕРЕЛІК ПОСИЛАНЬ

- 1) Google Maps Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://developers.google.com/maps/documentation>
- 2) Tripadvisor site [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.tripadvisor.com/>
- 3) BPMN [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/BPMN>
- 4) Google Chrome extension Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://developer.chrome.com/docs/extensions/>
- 5) Vue Google Chrome Extension template [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://github.com/Kocal/vue-web-extension>
- 6) Wextractor service [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://wextractor.com/>
- 7) ISO/IEC 25010, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>.
- 8) Spring Security Reference [Електронний ресурс] / [В. Alex, L. Taylor, R. Winch та ін.]. – 2021. – Режим доступу до ресурсу: <https://docs.spring.io/spring-security/site/docs/current/reference/html5>.

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ВІДГУКІВ ТА  
ОЦІНОК ПОСЛУГ  
Звіт подібності**

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олена ХАЛУС

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Ростислав ПРИЙМАК

Київ – 2022

Имя пользователя:  
Лісовиченко Олег Іванович

ID проверки:  
1011516560

Дата проверки:  
09.06.2022 11:13:21 EEST

Тип проверки:  
Doc vs Internet + Library

Дата отчета:  
09.06.2022 11:18:38 EEST

ID пользователя:  
76913

Название файла: ИП-82\_Приймак\_ПЗ

Количество страниц: 54    Количество слов: 7817    Количество символов: 61942    Размер файла: 1,006.02 KB    ID файла: 1011390660

## 6.32% Совпадения

Наибольшее совпадение: 3.94% с источником из Библиотеки (ID файла: 1000020642)

3.33% Источники из Интернета

12

Страница 56

6.32% Источники из Библиотеки

238

Страница 56

## 0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

## 0% Исключений

Нет исключенных источников

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ВІДГУКІВ ТА  
ОЦІНОК ПОСЛУГ**

**Опис програми**

КПІ.ПІ-8225.045440.03.13

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олена Халус

Нормоконтроль:

\_\_\_\_\_ Катерина ЛШЦУК

Виконавець:

\_\_\_\_\_ Ростислав Приймак

Київ – 2022

# ТЕКСТ ПРОГРАМНОГО КОДУ

## *Тексти програмного коду*

Програмне забезпечення для аналізу відгуків та оцінок послуг

---

(Найменування програми (документа))

DVD-R

---

(Вид носія даних)

7 арк, 225 Мб

---

(Обсяг програми, арк., Кб)

Київ – 2022

					КПІ.ІП-8225.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

```

package kpi.ua.diploma.service;

import kpi.ua.diploma.dto.FilterDto;
import kpi.ua.diploma.entity.Filter;
import kpi.ua.diploma.entity.User;
import kpi.ua.diploma.mapper.FilterMapper;
import kpi.ua.diploma.repository.FilterRepository;
import lombok.RequiredArgsConstructor;
import org.mapstruct.factory.Mappers;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Objects;
import java.util.Optional;

@Service
@RequiredArgsConstructor
public class FilterService {
    private final FilterRepository repository;
    private final UserService userService;
    private static final FilterMapper mapper = Mappers.getMapper(FilterMapper.class);

    public FilterDto create(FilterDto filterDto) {
        Filter filter = mapper.to(filterDto);
        filter.setUser(userService.getCurrentUser());
        filter.setEnabled(false);

        validateName(filter);

        return mapper.from(repository.save(filter));
    }

    private void validateName(Filter filter) {
        Optional<Filter> byName =
        repository.findByUser_UserIdAndName(filter.getUser().getUserId(),
        filter.getName());
        byName.ifPresent(value -> {
            if (!Objects.equals(filter.getId(), value.getId())) {
                throw new RuntimeException("filter name not unique for user");
            }
        });
    }

    public List<FilterDto> findAll() {
        User currentUser = userService.getCurrentUser();
        return repository.findAllByUser_UserId(currentUser.getUserId())
            .stream()
            .map(mapper::from)
            .toList();
    }

    public FilterDto update(FilterDto filterDto) {
        Filter request = mapper.to(filterDto);
        Filter filter = repository.findById(filterDto.getId()).orElseThrow(() -> new
        RuntimeException("filter not found"));

        filter.setEndDate(request.getEndDate());
        filter.setMinReviewForUser(request.getMinReviewForUser());
        filter.setName(request.getName());
        filter.setStartDate(request.getStartDate());
        filter.setEnabled(request.getEnabled());
        filter.setDiscount(request.getDiscount());
        filter.setUseDiscount(request.getUseDiscount());

        validateName(filter);
        disableAnotherFilter(filter);

        return mapper.from(repository.save(filter));
    }

    private void disableAnotherFilter(Filter filter) {
        if (filter.getEnabled()) {
            getEnabledFilter(filter.getUser().getUserId()).filter(f ->
            !Objects.equals(f.getId(), filter.getId())).ifPresent(value -> {
                value.setEnabled(false);
                repository.save(value);
            });
        }
    }
}

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІП-8225.045440.03.13

Арк.

3

```

private Optional<Filter> getEnabledFilter(Long userId) {
    return repository.findByUser_UserIdAndEnabled(userId, true);
}

public Optional<Filter> getEnabledFilter() {
    try {
        return getEnabledFilter(userService.getCurrentUser().getUserId());
    } catch (Exception e) {
        return Optional.empty();
    }
}

public void delete(Long id) {
    repository.deleteById(id);
}
}
package kpi.ua.diploma.service;

import kpi.ua.diploma.dto.PlaceDto;
import kpi.ua.diploma.entity.Place;
import kpi.ua.diploma.entity.Review;
import kpi.ua.diploma.mapper.PlaceMapper;
import kpi.ua.diploma.repository.PlaceRepository;
import lombok.RequiredArgsConstructor;
import org.mapstruct.factory.Mappers;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;

@Service
@RequiredArgsConstructor
public class PlaceService {
    private final PlaceRepository repository;
    private static final PlaceMapper mapper = Mappers.getMapper(PlaceMapper.class);
    private final ReviewService reviewService;
    private final FilterService filterservice;

    public PlaceDto getPlace(String link) {
        Place place = getByLink(link);

        filterservice.getEnabledFilter().ifPresent(filter -> {
            List<Review> reviews = reviewService.findAll(place.getId()).stream()
                .filter(review -> reviewService.isFitsFilter(filter, review))
                .toList();

            if (reviews.isEmpty()) {
                return;
            }

            if (!filter.getUseDiscount()) {
                reviews.stream()
                    .mapToDouble(Review::getRate)
                    .average()
                    .ifPresent(place::setRating);

                return;
            }

            reviews = new ArrayList<>(reviews);
            reviews.sort(Comparator.comparing(Review::getDate));

            double finalRate = 0.0;
            double discount = filter.getDiscount();
            for (Review review : reviews) {
                Double rate = review.getRate();
                finalRate = rate * (1 - discount) + finalRate * discount;
            }

            place.setRating(finalRate);
        });

        return mapper.from(place);
    }

    public Place getByLink(String link) {
        return repository.findByLink(link).orElseGet(() -> {

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІП-8225.045440.03.13

Арк.

4

```

        Place place = new Place();
        place.setLink(link);
        repository.save(place);
        return place;
    });
}

public void reEvaluateRatingAfterReview(Review review) {
    Place place = review.getPlace();
    if (place.getRating() != -1.0) {
        Long rates = place.getRates();
        Double rating = place.getRating();
        place.setRating((rates * rating + review.getRate()) / (rates + 1));
    } else {
        place.setRating(review.getRate());
    }
    place.setRates(place.getRates() + 1);
    repository.save(place);
}
}

package kpi.ua.diploma.service;

import kpi.ua.diploma.dto.ReviewDto;
import kpi.ua.diploma.entity.Filter;
import kpi.ua.diploma.entity.Review;
import kpi.ua.diploma.mapper.ReviewMapper;
import kpi.ua.diploma.repository.ReviewRepository;
import lombok.RequiredArgsConstructor;
import org.mapstruct.factory.Mappers;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Lazy;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
@RequiredArgsConstructor
public class ReviewService {
    private final ReviewRepository repository;
    private PlaceService placeService;
    private final UserService userService;
    private final FilterService filterService;
    private static final ReviewMapper mapper = Mappers.getMapper(ReviewMapper.class);

    @Autowired
    @Lazy
    public void setPlaceService(PlaceService placeService) {
        this.placeService = placeService;
    }

    public ReviewDto create(ReviewDto dto) {
        Review review = mapper.from(dto);

        review.setUser(userService.loadUserByUsername(userService.getCurrentUser().getUsername()));
        review.setPlace(placeService.getByLink(dto.getPlaceLink()));
        Review saved = repository.save(review);
        placeService.reEvaluateRatingAfterReview(review);
        return mapper.to(saved);
    }

    public List<ReviewDto> findAll(Long placeId, Pageable pageable) {
        Optional<Filter> enabledFilter = filterService.getEnabledFilter();
        if (enabledFilter.isPresent()) {
            Filter filter = enabledFilter.get();

            return repository.findAllByPlace_Id(placeId, pageable)
                .stream()
                .filter(review -> isFitsFilter(filter, review))
                .map(mapper::to)
                .toList();
        }
        return repository.findAllByPlace_Id(placeId, pageable)
            .stream()
            .map(mapper::to)
            .toList();
    }
}

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІП-8225.045440.03.13

Арк.

5

```

public List<Review> findAll(Long placeId) {
    return repository.findAllByPlace_Id(placeId);
}

public long getCountByUser(Long userId) {
    return repository.countAllByUser_UserId(userId);
}

public boolean isFitsFilter(Filter filter, Review review) {
    return !isStartDateCheckFailed(filter, review) &&
!isEndDateCheckFailed(filter, review) && !isMinReviewsFilterCheckFailed(filter,
review);
}

private boolean isMinReviewsFilterCheckFailed(Filter filter, Review review) {
    return filter.getMinReviewForUser() != null &&
        filter.getMinReviewForUser() > 0 &&
        getCountByUser(review.getUser().getUserId()) <
filter.getMinReviewForUser();
}

private boolean isEndDateCheckFailed(Filter filter, Review review) {
    return filter.getEndDate() != null &&
filter.getEndDate().isBefore(review.getDate());
}

private boolean isStartDateCheckFailed(Filter filter, Review review) {
    return filter.getStartDate() != null &&
filter.getStartDate().isAfter(review.getDate());
}
}
package kpi.ua.diploma.service;

import kpi.ua.diploma.dto.UserDto;
import kpi.ua.diploma.entity.User;
import kpi.ua.diploma.mapper.UserMapper;
import kpi.ua.diploma.repository.UserRepository;
import lombok.RequiredArgsConstructor;
import lombok.extern.log4j.Log4j2;
import org.mapstruct.factory.Mappers;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;

import java.time.LocalDate;

@Log4j2
@Service
@RequiredArgsConstructor
public class UserService implements UserDetailsService {
    private final UserRepository repository;
    private final PasswordEncoder passwordEncoder = new BCryptPasswordEncoder(8);
    private static final UserMapper mapper = Mappers.getMapper(UserMapper.class);

    public void saveNewUser(UserDto userDTO) {
        log.info("save new user {}", userDTO.toString());

        User user = mapper.from(userDTO);
        user.setPassword(passwordEncoder.encode(user.getPassword()));
        user.setRegistrationDate(LocalDate.now());
        try {
            repository.save(user);
        } catch (Exception e) {
            throw new RuntimeException("User already exists");
        }
    }

    @Override
    public User loadUserByUsername(String username) {
        return repository.findByUsername(username)
            .orElseThrow(() -> new RuntimeException("User not found"));
    }

    public User getCurrentUser() {
        Object principal =
SecurityContextHolder.getContext().getAuthentication().getPrincipal();

```

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

КПІ.ІП-8225.045440.03.13

Арк.

6

```
if (principal instanceof User) {
    return repository.getById(((User) principal).getUserId());
} else {
    throw new RuntimeException("user not logged in");
}
}
```

					КПІ.ІП-8225.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ВІДГУКІВ ТА  
ОЦІНОК ПОСЛУГ**

**Програма та методика тестування**

КПІ.ПІ-8225.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олена ХАЛУС

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Ростислав ПРИЙМАК

Київ – 2022

## ЗМІСТ

1 Об'єкт випробувань .....	3
2 Мета тестування .....	4
3 Методи тестування.....	5
4 Засоби та порядок тестування.....	6

					КПІ.ІП-8225.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

## 1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є програмне забезпечення, що було розроблено для покращення вибору певного місця чи послуги. Google Chrome розширення було розроблено за допомогою фреймворків Vue.js та Spring. Програмне забезпечення може бути запущене у браузері Google Chrome на Windows, Linux та MacOS системах.

					КПІ.ІП-8225.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

## 2 МЕТА ТЕСТУВАННЯ

Дане програмне забезпечення протестовано з метою перевірки коректності роботи основного функціоналу:

- надавати можливість створення нового об'єкту місця у випадку відсутності;
- надавати можливість збереження налаштувань у вигляді створення фільтру.
- надавати можливість перегляду фінальної оцінки за посиланням
- надавати можливість для реєстрації та авторизації
- надавати можливість перегляду існуючих відгуків
- надавати можливість включати та виключати фільтри
- надавати можливість редагування та видалення фільтрів
- надавати можливість додавання відгуку за посиланням

					КПІ.ІП-8225.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

### 3 МЕТОДИ ТЕСТУВАННЯ

Тестування проводилось у ролі користувача. Тобто розробник, що виступає у ролі користувача, перевіряє усі можливі юз кейси поведінки користувача за коректної та некоректної роботи програми. Також під час тестування було використано логування подій програмного забезпечення для пришвидшення пошуку помилок. У результаті було проведено тестування наступних рис програмного забезпечення:

- тестування UI;
- функціональне тестування;
- нефункціональне тестування;

					КПІ.ІП-8225.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

#### 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується вручну, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності в користуванні. Для того, щоб перевірити працездатність та відмовостійкість додатку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на персональних комп'ютерах з різною версією операційної системи та браузеру;
- тестування на виведення повідомлень про помилку, коли це необхідно;
- тестування працездатності програми у випадку відсутності з'єднання до мережі;
- тестування інтерфейсу користувача;
- тестування зручності використання.

					КПІ.ІП-8225.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ВІДГУКІВ ТА  
ОЦІНОК ПОСЛУГ**

**Керівництво користувача**

КПІ.ПІ-8225.045440.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Олена ХАЛУС

Нормоконтроль:

\_\_\_\_\_ Катерина ЛШЦУК

Виконавець:

\_\_\_\_\_ Ростислав ПРИЙМАК

Київ – 2022

## ЗМІСТ

1 Загальні відомості .....	3
2 Підготовка до роботи.....	4
2.1 Системні вимоги для коректної роботи .....	4
2.2 Завантаження застосунку.....	4
2.3 Перевірка коректної роботи.....	4
3 Робота із застосунком.....	5

					КПІ.ІП-8225.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

## 1 ЗАГАЛЬНІ ВІДОМОСТІ

«Програмне забезпечення для аналізу відгуків та оцінок послуг» - це застосунок у вигляді Google Chrome розширення для звичайних користувачів що хочуть вибрати місце чи послугу що буде максимально задовільняти їхні бажання.

					КПІ.ІП-8225.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

## 2 ПІДГОТОВКА ДО РОБОТИ

### 2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- наявність персонального комп'ютера із операційною системою Windows, Linux чи MacOS;
- наявність доступу до Інтернету;
- для встановлення додатку на персональному комп'ютері повинно бути не менше 70 МБ вільної пам'яті та встановлено браузер Google Chrome.

### 2.2 Завантаження застосунку

На даний момент застосунок можна встановити через магазин Google Store.

### 2.3 Перевірка коректної роботи

По завершенню встановлення розширення повинна з'явитися іконка у налаштуваннях розширення, яку потім можна перенести та закріпити на панель у браузері.

					КПІ.ІП-8225.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

### 3 РОБОТА ІЗ ЗАСТОСУНКОМ

При запуску програмного застосунку користувачу буде відображено сторінку що містить відгуки та фінальну оцінку, в даному випадку відгуків ще не було, тому бачимо лише відповідне повідомлення(рис. 3.1).



Рисунок 3.1 – Початкова сторінка додатку

Користувач має змогу зареєструватися та авторизуватися. При реєстрації користувач повинен ввести ім'я користувача, електронну пошту, мобільний телефон та пароль для входу(рис. 3.2). Після цього можна натиснути на кнопку із підтвердженням, та користувач перейде на сторінку для авторизації.

The image shows a web registration form with a header containing three links: 'Reviews', 'Login', and 'Sign up'. The main heading is 'Sign up'. Below it are four input fields labeled 'username', 'email', 'phone', and 'password'. At the bottom is a yellow 'Submit' button.

Рисунок 3.2 – Сторінка розширення із реєстрацією

Для авторизації користувачу потрібно буде ввести ім'я користувача та пароль. Після натискання кнопки користувач буде перенаправлений на сторінку із додаванням відгуків.

The image shows a web interface for user authentication. At the top, there are three tabs: "Reviews", "Login", and "Sign up". The "Sign in" title is centered. Below it are two input fields labeled "username" and "password". A yellow "Submit" button is positioned below the password field.

Рисунок 3.3 – Сторінка розширення із авторизацією

Для додавання відгуку користувачу потрібно ввести значення рейтингу та написати коментар у відповідних полях. Після чого натиснути на кнопку із підтвердженням.

The image shows a web interface for adding a review. At the top, there is a navigation bar with the following items: 'Add review', 'Reviews', 'Filters', 'Add filter', and 'Logout'. Below this bar, the main form area contains a 'Rate' section with a slider control currently set to 0. Underneath the slider is a 'Comment' section, which is a large, empty text input box. At the bottom center of the form is a prominent yellow button labeled 'add'.

Рисунок 3.4 – Сторінка із додаванням відгуку

Після додавання відгуку користувач перенаправляється на сторінку із відгуками та фінальною оцінкою. Оскільки ми додали відгук, тепер ми можемо бачити його у списку і також фінальну оцінку.

Змін.	Арк.	№ докум.	Підп.	Дата.

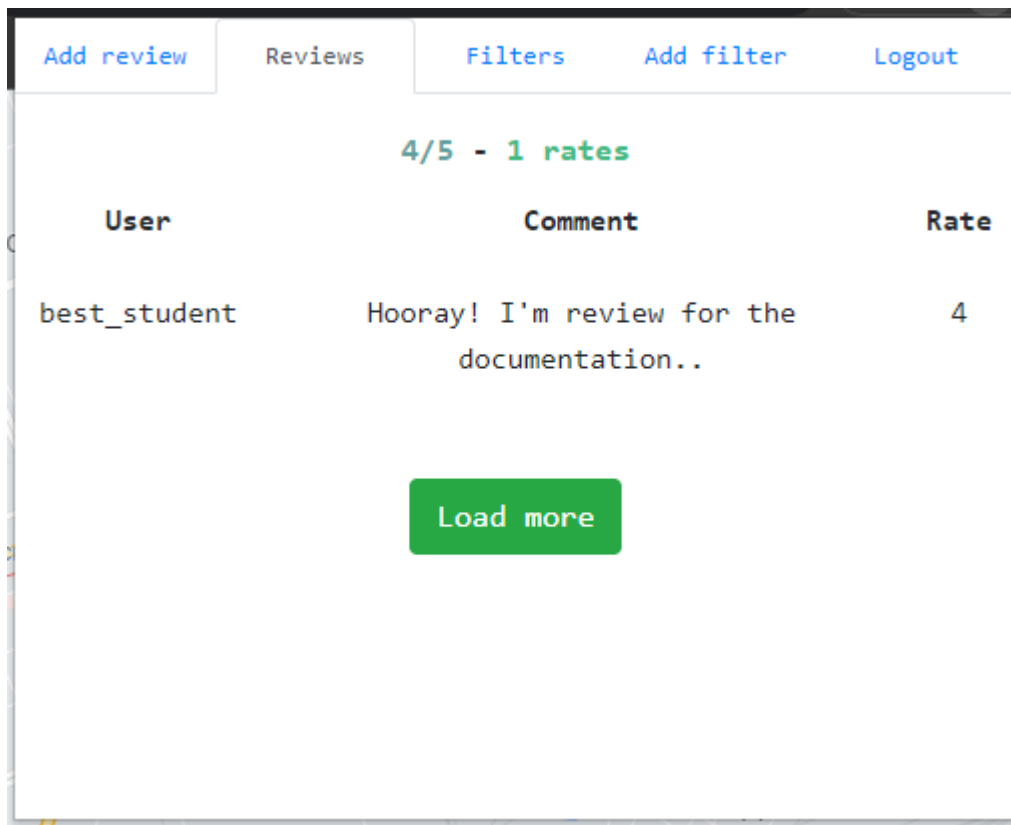


Рисунок 3.5 – Сторінка із переглядом відгуків та оцінки після додавання відгуку

Після реєстрації користувачу також доступні сторінки для збереження налаштувань – з фільтрами. Для створення фільтру потрібно додати назву фільтру і також інші поля за бажанням(рис. 3.6).

[Add review](#)   [Reviews](#)   [Filters](#)   [Add filter](#)   [Logout](#)

filter name

since:

mm/dd/yyyy

until:

mm/dd/yyyy

min reviews for user

enable filter

enable discount

discount

Рисунок 3.6 – Сторінка із створенням фільтру

Після створення фільтру користувач перенаправляється на сторінку із переглядом фільтрів, де включений фільтр позначено зеленим кольором(рис. 3.5).

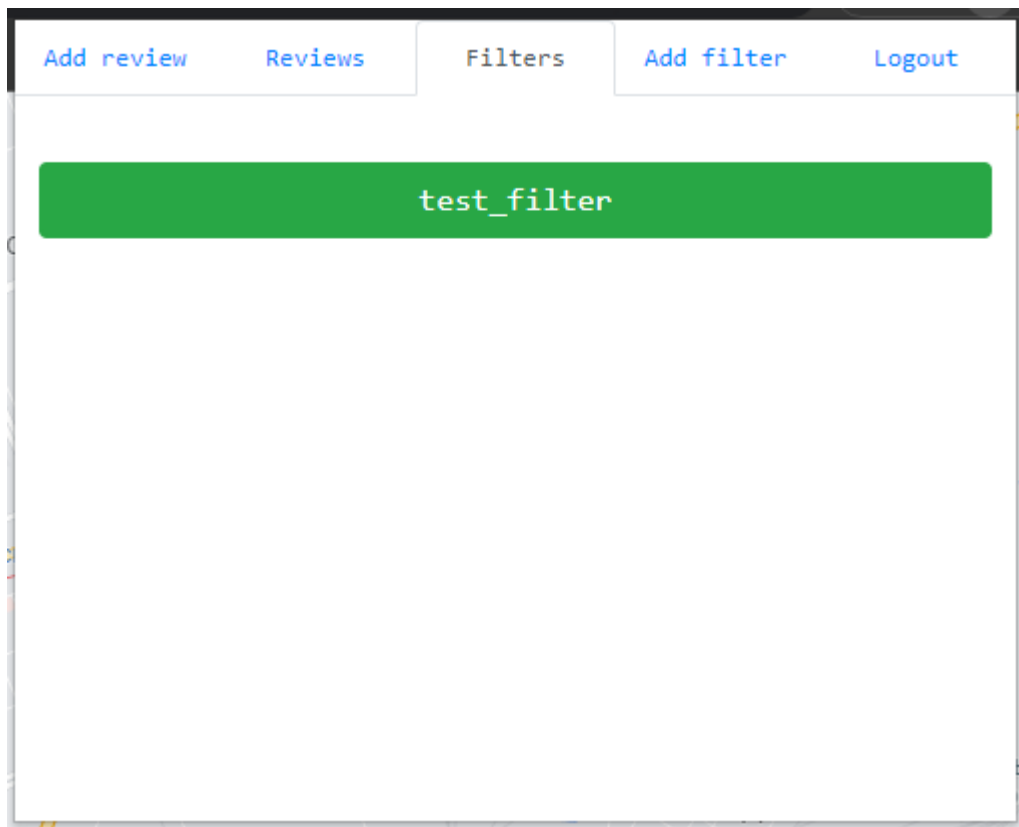


Рисунок 3.7 – Сторінка із переглядом фільтрів

Натиснувши на назву фільтру відкривається вкладка із редагування фільтру(рис. 3.8). На ній можна змінити деякі параметри із фільтру, включаючи його включення та виключення, і також його видалення.

Filter name

test\_filter

Since:

06/14/2022

Until:

06/15/2022

Min reviews for user:

1

Enable filter

Enable discount

0

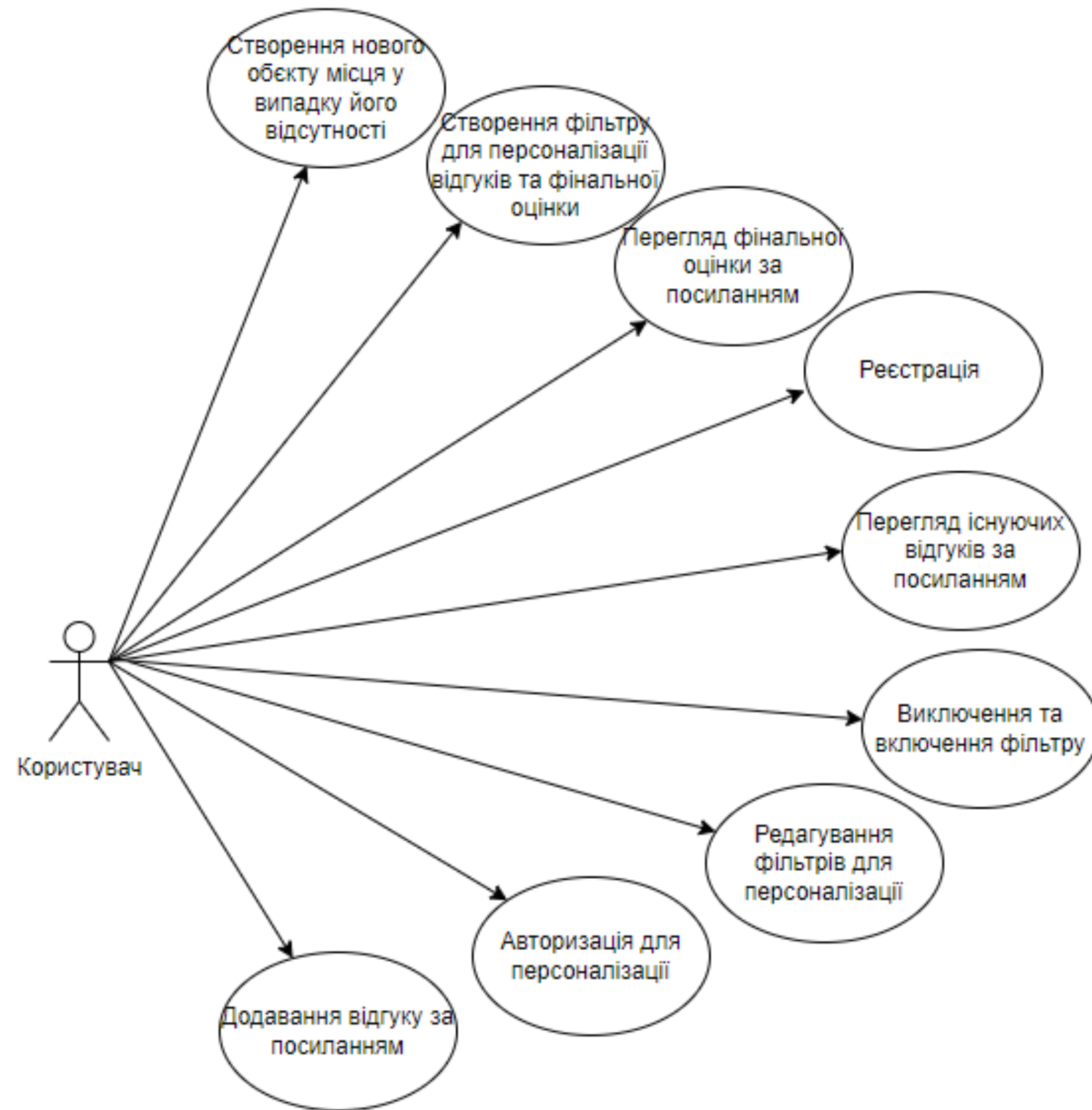
Discount

Delete Save

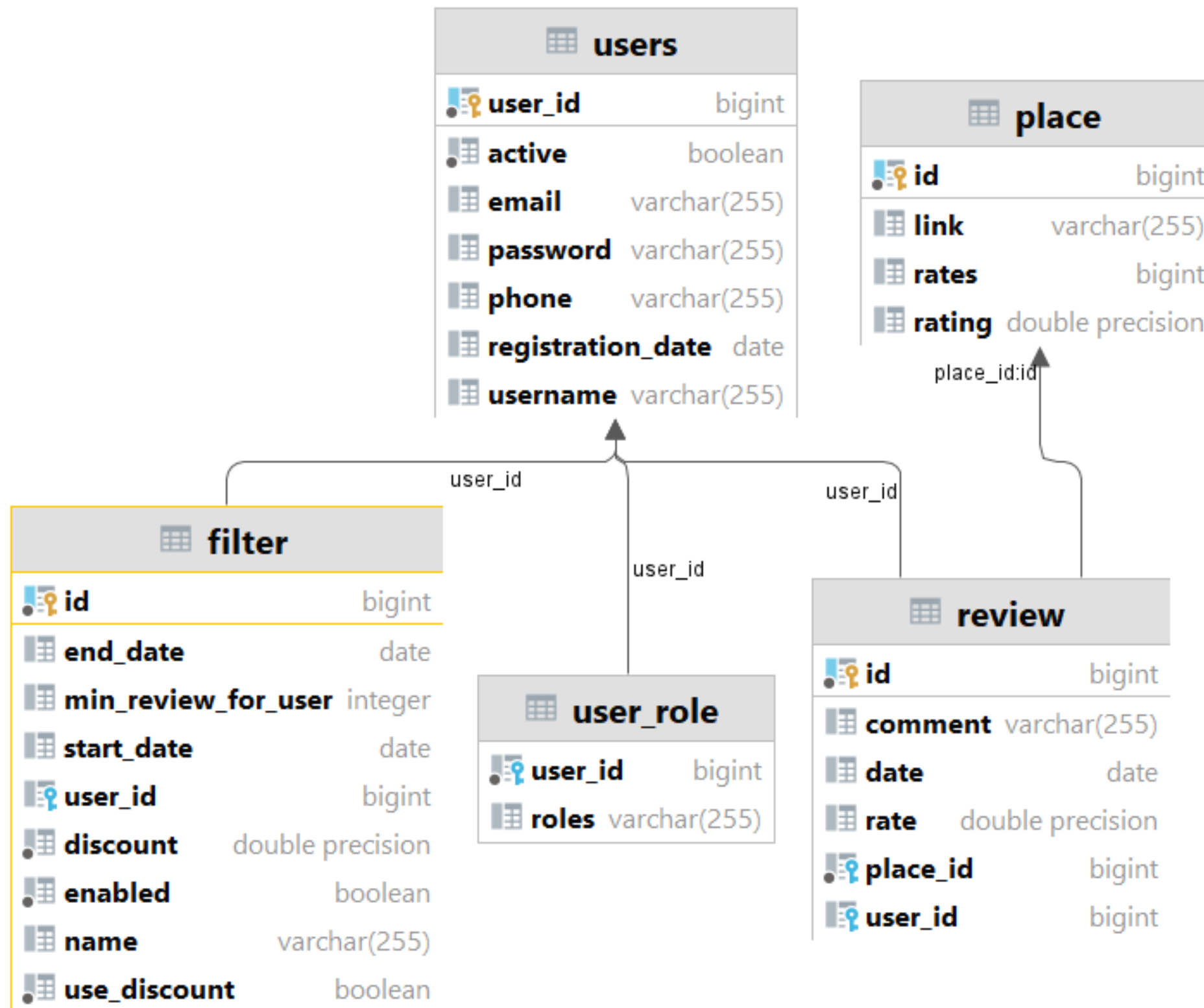
Рисунок 3.9 – Сторінка із редагуванням та видаленням фільтрів

Для авторизованого користувача доступна кнопка для виключення авторизації. Для не авторизованих – кнопки авторизації та реєстрації, зо переносять користувача на відповідні сторінки.

Змін.	Арк.	№ докум.	Підп.	Дата.



					КПІ.ІП-8225.045440.06.99.СС			
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна варіантів використань	Літера	Маса	Масштаб
Розробив		Приймак Р.Я.						
Перевішив		Халус О.А.						
Т. кон.						Аркуш	Аркушів	
Н. кон.		Ліщук К.І.			Програмне забезпечення для аналізу відгуків та оцінок послуг	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-82		
Затвердив		Халус О.А.						



					КПІ.ІП-8225.045440.07.99.СБД		
					Схема Бази даних		
Зм.	Арк.	№ документа	Підпис	Дата			
Розробив		Приймак Р.Я.					
Перевірів		Халус О.А.					
Т. кон.					Аркуш		Аркушів
Н. кон.		Ліщук К.І.			КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-82		
Затвердив		Халус О.А.					
					Програмне забезпечення для аналізу відгуків та оцінок послуг		

