



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

### **ЛАБОРАТОРНА РОБОТА 3**

З дисципліни “Теорія систем та системний аналіз”

Тема: Побудова Діаграми Взаємодії

**Виконали:**

студент групи ІА-11:

Воробей А. О

студенти групи ІА-13:

Середа А. А.

Павлюк О. І

студент групи ІА-14:

Фіалківський І. О.

**Перевірив:**

Барбарук В. М.

**Тема:** Побудова Діаграми Взаємодії

**Мета:** Ознайомлення з методологією та інструментальними засобами моделювання взаємодії компонентів системи на основі мови UML.

### **Хід роботи**

Діаграма взаємодії за своєю суттю дуже сильно схожа на діаграму послідовностей але в трохи іншому візуальному представленні. Насправді це так і є. Діаграма взаємодії дозволяє нам віодобразити потік подій у контреному варіанті використання. Різні об'єкти покроково взаємодіють між собою, щоб забезпечити кінцеву функціональність. Різниця між діаграмою взаємодії та діаграмою послідовності заключається в тому, що на діаграмі взаємодії можуть бути показані додаткові структурні відносини, що беруть участь у взаємодіях між об'єктами. Наприклад це можуть бути відносини між атрибутами, класами, а також багато інших додаткових речей

Усі діаграми взаємодії нижче розділені на дві частини: Frontend частина та Backend. З Frontend частиною взаємодіє користувач. Саме на ній відбувається читання творів, оцінення творів, перегляд статистики і тому подібні речі. На Backend частині виконується вся логіка застосунку електронною бібліотеки

## Воробей Антон

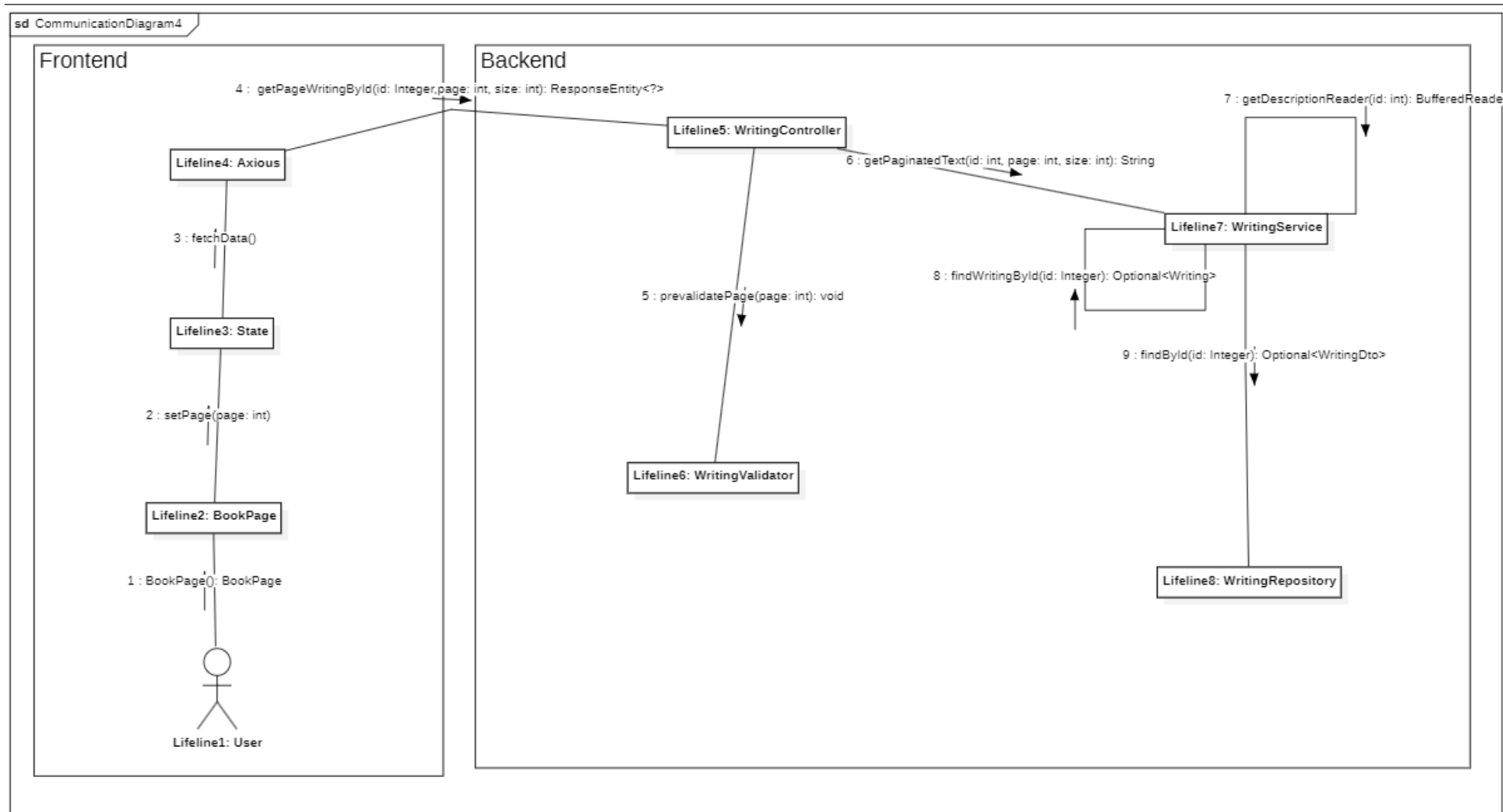


Рис. 1. Діаграма взаємодії процесу читання сторінки книги

На Рисунку 1 зображено користувача, що виконує читання певної сторінки книги. Спочатку користувач відкриває компонент сторінки книги та вказує бажану сторінку. З введеної інформації утворюється запит, що відправляється на серверну частину нашого застосунку. На серверній частині відбувається обробка цього запиту. Спочатку відбувається початкова валідація бажаного номера сторінки. Якщо валідація пройшла успішно, то відбувається виклик метода в сервісі, що здатен повернути пагіновану сторінку. Щоб отримати текст книжки, ми виконуємо запит до бази даних, отримуємо розташування цієї книжки, та зчитуємо її. За допомогою спеціального алгоритму відбувається обчислення сторінки та перевірка чи ця сторінка можлива. Якщо все пройшло успішно, текст повертається на клієнтську частину та відображається користувачу. Якщо на якомусь етапі виникла помилка, користувач буде повідомлений про це

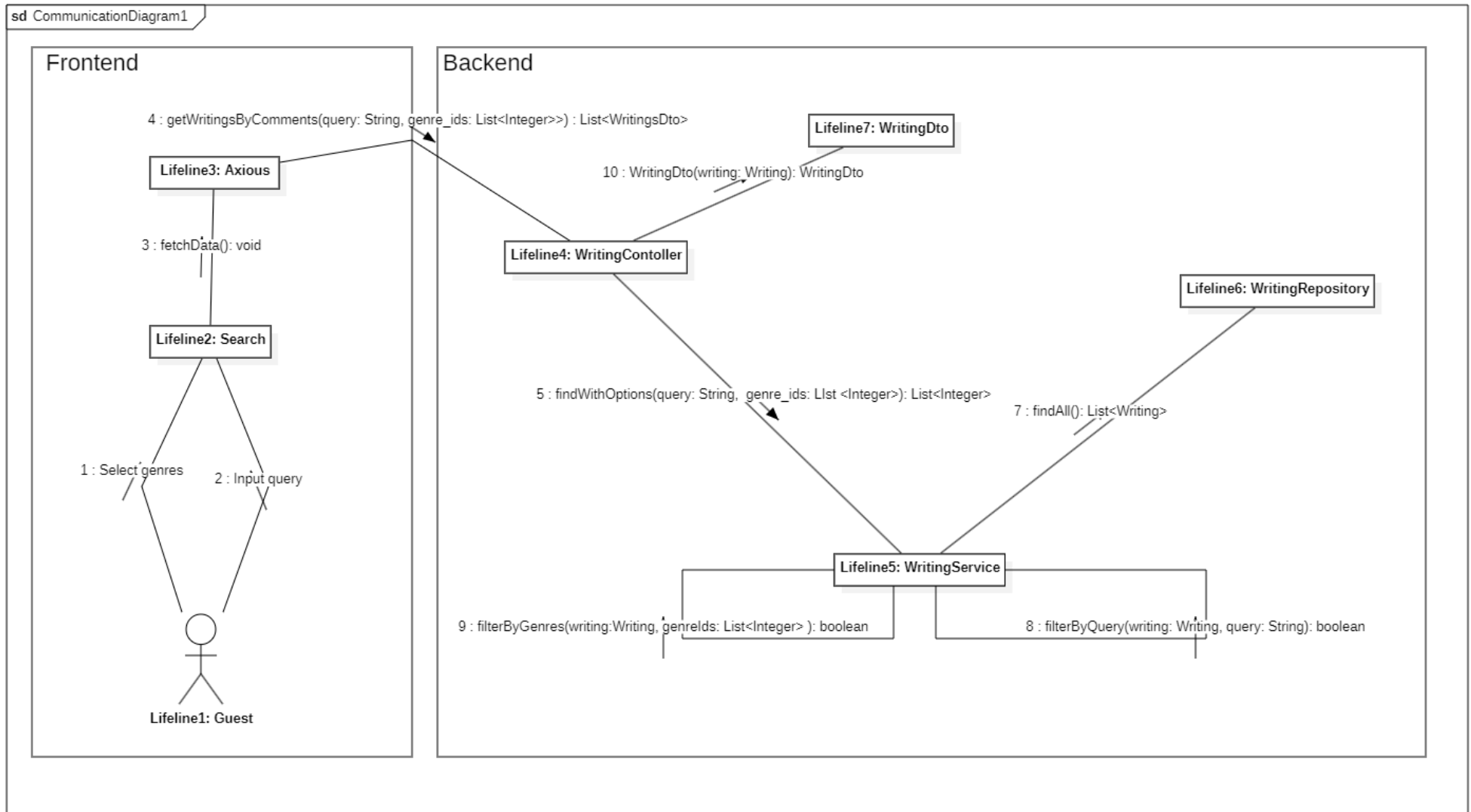


Рис. 2. Діаграма взаємодії процесу пошуку твору за жанрами та назвою

На Рисунку 2 зображено діаграму взаємодії процесу пошуку твору за жанрами. Користувач відкриває сторінку(компонент) пошуку на якій можна вказати назву а також жанри. Після того, як інформація була вказана, вона надсилається з клієнтської частини на серверну частину. На серверній частині виконується обробка отриманого запиту. Контролер приймає запит, виконує базову його валідацію та звертається до сервіса, де і лежить вся бізнес логіка. Сервіс звертається до репозиторію, який в свою чергу надсилає запит в базу даних щоб отримати інформацію про усі твори. Коли інформація була успішно отримана, в сервісі відбувається фільтрація цієї інформації за параметрами що надав користувач. Відфільтровані твори перетворюються в об'єкти для передавання та надсилаються на клієнську частину. Клієнту відображаються знайдені результати

## Фіалківський Ілля

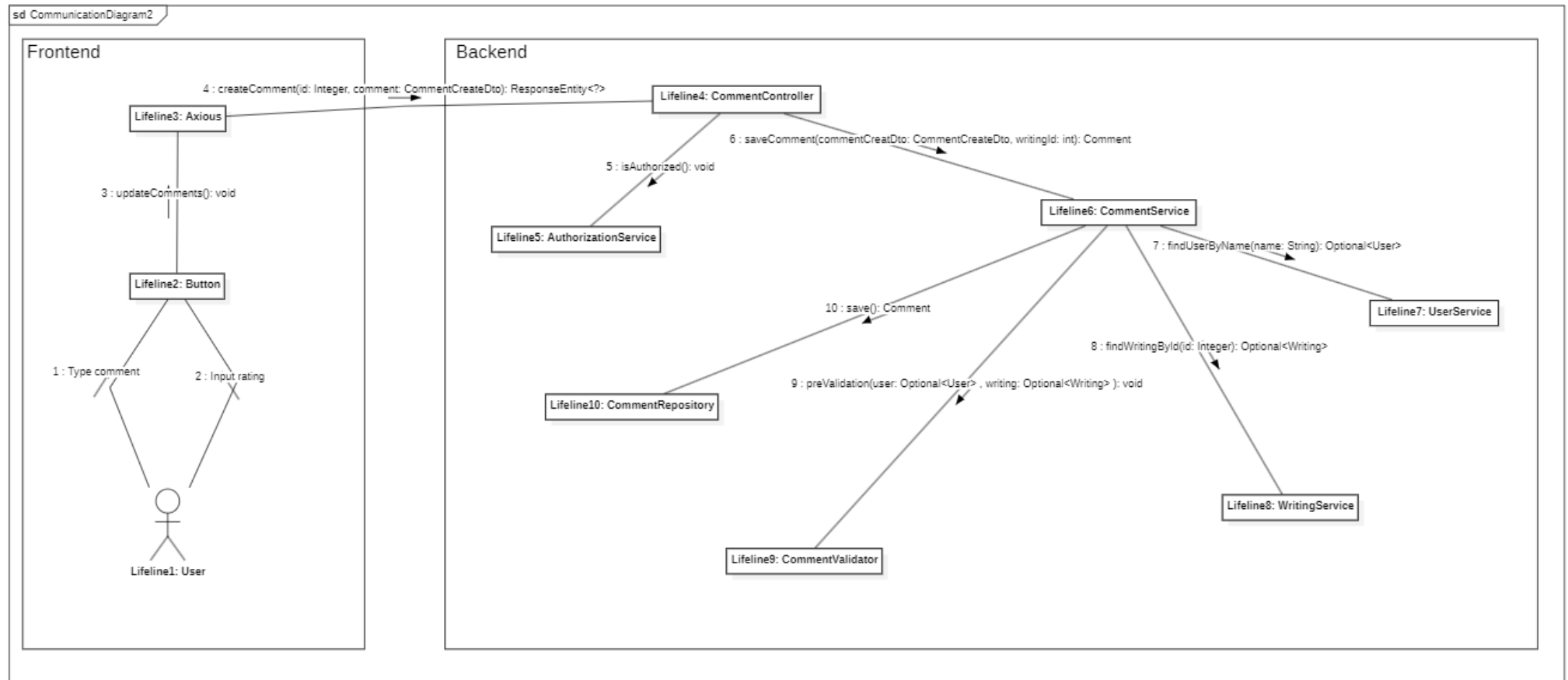


Рис. 3. Діаграма взаємодії процесу створення коментаря та оцінення твору

На Рисунку 3 вище зображена діаграма взаємодії для процесу створення коментаря та оцінення твору

1. Введення інформації. Користувач вводить коментарій та рейтинг. Та натискає кнопку пошуку
2. Відправка інформації на сервер. На клієнтській частині застосунку відбувається відправка запиту з вказаними параметрами на сервер
3. Обробка запиту на сервері. Сервер приймає запит та обробляє його. Спочатку відбувається перевірка чи користувач авторизований. Якщо користувач пройшов цю перевірку, то потрібна інформація передається в сервіс
4. Обробка даних в сервісі та запити до бази даних. Коли текст коментарю та рейтинг попадає до сервісу, відбуваються усі потрібні запити до бази даних через репозиторії. Після того як уся потрібна інформація отримана, відбувається виконання усієї валідації. Якщо валідація пройшла успішно то дані зберігаються до бази даних
5. Відображення коментарю та статистики. Якщо з серверу приходить підтвердження, що операція була виконана успішно, то користувач може переглянути новостворений коментарій та оцінки. Якщо прийшла помилка, то користувач побачить повідомлення цієї помилки



## Павлюк Оскар

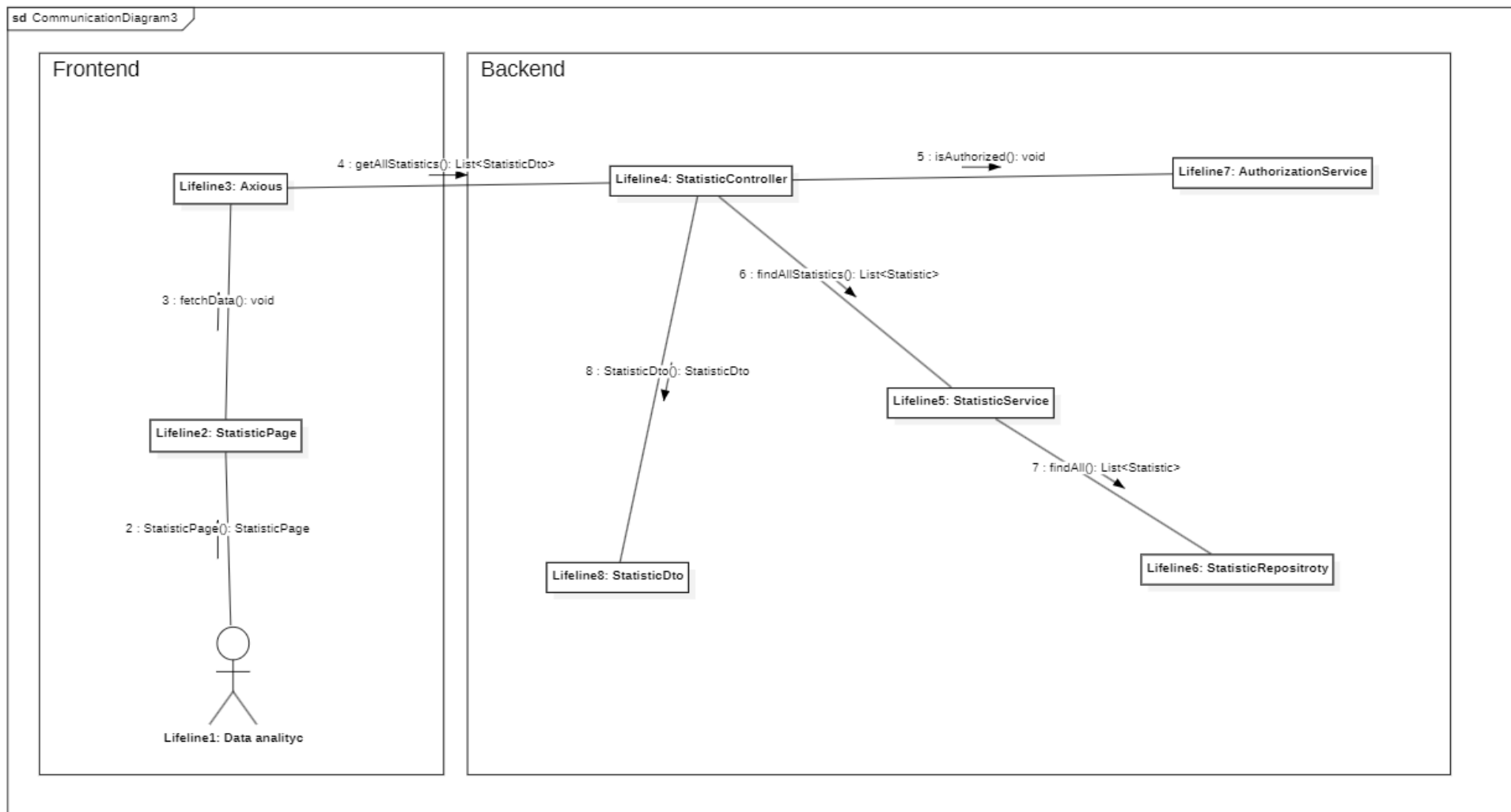


Рис. 4. Спрощена діаграма взаємодії процесу перегляду статистики твору

На Рисунку 4 я зобразив діаграму взаємодії, що показує процес перегляду статистики творів. Щоб переглядати таку статистику користувач має бути залогіненим як аналітик даних. Користувач відкриває сторінку статистики. Під час відкриття сторінки відбувається відправлення запиту на сервер. На сервері контролер обробляє запит від користувача. Спочатку відбувається перевірка того, чи користувач є дійсно аналітиком даних. Якщо перевірка пройшла успішно, то відбувається отримання сутностей з бази даних. Сутності перетворюються на передавальні об'єкти та відправляються користувачу назад. На стороні користувача відбувається відображення отриманої статистики

### **Посилання**

Код діаграм та звіти розміщені в репозиторії

<https://github.com/korovkaK22/LibraryDiargarm>

Код backend частини розміщений в репозиторії

<https://github.com/korovkaK22/biblioteka>

Код frontend частини розміщений в репозиторії

<https://github.com/vergovters/library>

**Висновки:** на цій лабораторній роботі ми навчилися працювати з діаграмами взаємодії. В процесі виконання роботи ми вивчили, що таке діаграма взаємодії, де вона використовується та яка різниця між діаграмою взаємодії та діаграмою послідовності. За для того, щоб закріпити знання на практиці кожен з учасників команди виконав створення власної діаграми взаємодії та описав її